Adaptation, Learning, and Optimization 19

D. P. Acharjya
Satchidananda Dehuri
Sugata Sanyal *Editors*

Computational Intelligence for Big Data Analysis

Frontier Advances and Applications

Springer

Adaptation, Learning, and Optimization

Volume 19

Series editors

Meng-Hiot Lim, Nanyang Technological University, Singapore
e-mail: emhlim@ntu.edu.sg

Yew-Soon Ong, Nanyang Technological University, Singapore
e-mail: asysong@ntu.edu.sg

About this Series

The role of adaptation, learning and optimization are becoming increasingly essential and intertwined. The capability of a system to adapt either through modification of its physiological structure or via some revalidation process of internal mechanisms that directly dictate the response or behavior is crucial in many real world applications. Optimization lies at the heart of most machine learning approaches while learning and optimization are two primary means to effect adaptation in various forms. They usually involve computational processes incorporated within the system that trigger parametric updating and knowledge or model enhancement, giving rise to progressive improvement. This book series serves as a channel to consolidate work related to topics linked to adaptation, learning and optimization in systems and structures. Topics covered under this series include:

- complex adaptive systems including evolutionary computation, memetic computing, swarm intelligence, neural networks, fuzzy systems, tabu search, simulated annealing, etc.
- machine learning, data mining & mathematical programming
- hybridization of techniques that span across artificial intelligence and computational intelligence for synergistic alliance of strategies for problem-solving.
- aspects of adaptation in robotics
- agent-based computing
- autonomic/pervasive computing
- dynamic optimization/learning in noisy and uncertain environment
- systemic alliance of stochastic and conventional search techniques
- all aspects of adaptations in man-machine systems.

This book series bridges the dichotomy of modern and conventional mathematical and heuristic/meta-heuristics approaches to bring about effective adaptation, learning and optimization. It propels the maxim that the old and the new can come together and be combined synergistically to scale new heights in problem-solving. To reach such a level, numerous research issues will emerge and researchers will find the book series a convenient medium to track the progresses made. More information

about this series at http://www.springer.com/series/8335

D.P. Acharjya · Satchidananda Dehuri
Sugata Sanyal
Editors

Computational Intelligence for Big Data Analysis

Frontier Advances and Applications

Springer

Editors
D.P. Acharjya
School of Computing Sciences
and Engineering
VIT University
Vellore
India

Satchidananda Dehuri
Department of Information
and Communication Technology
Fakir Mohan University
Balasore
India

Sugata Sanyal
Corporate Technology Office
Tata Consultancy Services
Mumbai
India

ISSN 1867-4534 ISSN 1867-4542 (electronic)
Adaptation, Learning, and Optimization
ISBN 978-3-319-16597-4 ISBN 978-3-319-16598-1 (eBook)
DOI 10.1007/978-3-319-16598-1

Library of Congress Control Number: 2015934937

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

The amount of data collected from various sources is expected to double every two years. It has no utility unless these are analyzed to get useful information and it necessitates the development of techniques which can be used to facilitate big data analysis. The transformation of big data into knowledge is by no means an easy task. Expressing data access requirements of applications and designing programming language abstractions to exploit parallelism are an immediate need. In addition, these data may involve uncertainties. Big-data analysis, intelligent and cloud computing are active areas of current research for their potential application to many real life problems. Therefore, it is challenging for human beings to analyze and extract expert knowledge from a universe due to lack of computing resources available. More importantly, these new challenges may comprise, sometimes even deteriorate, the performance, efficiency, and scalability of the dedicated data intensive computing systems. In addition, fast processing while achieving high performance and high throughput, and storing it efficiently for future use is another issue.

The objective of this edited book is to provide the researchers and practitioners the recent advances in the fields of big-data analysis and to achieve these objectives, both theoretical advances, and its applications to real life problems, case studies are stressed upon. This will stimulate further research interest in big data analytics. Moreover, it will help those researchers who have interest in this field of big data analysis and cloud computing and their importance for applications in real life. The book is comprised of three sections. The first section is an attempt to provide an insight on theoretical foundation on big data analysis that includes scalable architecture for big data processing, time series forecasting for big data, hybrid intelligent techniques, and applications to decision making by using neutrosophic sets. The second section discusses architecture for big data analysis and its applications whereas final section discusses the issues pertaining to cloud computing.

Data structures provide the ability to computers to fetch and store data efficiently at any place in its memory. But, it is difficult to think of a data structure for big data. For processing of giant big data in 4Vs (Volume, Varity, Velocity and Veracity) neither any existing data structure nor any existing type of distributed system is sufficient. Also, existing network topologies seem to be weak topologies for big

data processing. Therefore, it is essential to develop a new data structure, new distributed systems for big data having a very fast and tremendous extent of mutual compatibility and mutual understanding with the new data structures. In addition, it is also essential to have a new type of network topologies for big data to support the new distributed system and of course new mathematical models for big data science. Another important issue is to integrate all these new to make ultimately a single and simple scalable system to the laymen users with a new simple big data language. With these views in mind, a special type of distributed system called Atrain distributed system (ADS) is discussed in Chapter 1 which is very suitable for processing big data using the heterogeneous data structure r-atrain for big data. Consequently, new network topologies such as multi-horse cart topology and cycle topology are introduced to support the new ADS.

In recent years data accumulated through various sources contains uncertainties and vagueness. Crisp models fail to process such uncertainties present in the datasets. On the other hand, with the rise of big data concept, demand for a new time series prediction models emerged. For this purpose, a novel big data time series forecasting model is thoroughly discussed in Chapter 2. It hybridizes two intelligent computing techniques such as fuzzy set and artificial neural network. The new hybridized model establishes the linear association among the various fuzzified observations, and takes the decision from these fuzzy relations. The major advantage is that the proposed model is applicable to the problems where massive fuzzy sets are involved.

Chapter 3 focuses on a few key applications of hybrid intelligence techniques in the field of feature selection and classification. Hybrid intelligent techniques have been used to develop an effective and generalized learning model for solving these applications. Some of the applications are addressed in this chapter. The first application integrates correlation based binary particle swarm optimization (BPSO) with rough set algorithm whereas another application is addressed on using correlation based partial least square (PLS). It also discusses a hybridized correlation based reduct algorithm with rough set theory (CFS-RST). To support these algorithms, in chapter 3, extensive study is carried out on two public multi-class gene expression datasets.

Smarandache introduced the concept of neutrosophic set in 2005 which is a mathematical tool for handling problems involving imprecise, indeterminacy and inconsistent data. A neutrosophic set has the potentiality of being a general framework for uncertainty analysis in data sets also including big datasets. Useful techniques like distance and similarity between two neutrosophic sets have been discussed in Chapter 4. These notions are very important in the determination of interacting segments in a dataset. The chapter also introduces the notion of entropy to measure the amount of uncertainty expressed by a neutrosophic set. It is further generalized to interval valued neutrosophic sets. Some important properties of these sets under various algebraic operations and its application in decision making have been discussed in this chapter.

Second section of this book gives emphasis on architecture development that helps in analyzing big data. Clustering is the unsupervised classification of patterns

into groups. In addition, determination of the number of clusters is a major issue. To begin with in Chapter 5, an efficient grouping genetic algorithm for data clustering and big data analysis is discussed under the circumstances of an anonymous number of clusters. The chapter also discusses concurrent clustering with different number of clusters on the same data. The proposed algorithm identifies new solutions with different clusters. The accuracy and the efficiency of the algorithm are also tested on various artificial and real data sets in a comparable manner. The theory developed will lead to the successful applications in the big data analysis.

Self organizing migrating algorithm (SOMA) is a population based stochastic search algorithm which is based on the social behavior of group of individuals. The main characteristics of SOMA are that it works with small population size and no new solutions are generated during the search, only the positions of the solutions are changed. Though it has good exploration and exploitation qualities but as the dimension of the problem increases it trap to local optimal solution and may suffer from premature convergence due to lack of diversity mechanism. To overcome the limitations, a hybrid variant of self organizing migrating algorithm for large scale function optimization, which combines the features of Nelder Mead crossover operator and log-logistic mutation operator, is thoroughly discussed in Chapter 6. In addition, the hybridized algorithm is tested on a set of large scale unconstrained test problems with considerable increase in problem size.

With the expansion of information and communication technology, vast amount of data is getting collected from various resources. In addition, storing, processing, and data analysis became a challenging task. Also, mining hidden information from big data provides various application capabilities. To provide an overview on this, Chapter 7 gives emphasis on mining in medical application domain. In addition, a framework which can handle big data by using several preprocessing and data mining technique to discover hidden knowledge from large scale databases is designed and implemented.

Electroencephalogram (EEG) is gaining attention and becoming attractive to researchers in various fields of engineering. Chapter 8 discusses various data analysis techniques devoted to the development of brain signals controlled interface devices for the purpose of rehabilitation in multidisciplinary engineering. The knowledge of EEG is essential for the neophytes in the development of algorithms. Most literatures, demonstrates the application of EEG signals and no much definite study describes the various components that are censorious for development of interface devices using prevalent algorithms in real time data analysis. Keeping this in mind, chapter 8 covers the EEG generation, various components of EEG used in development of interface devices and algorithms used for identification of information from EEG.

The vast amount of data (big-data) collected from various sources has no utility unless these are analyzed to get useful information. More often than not, the big-data are reduced to include only the important characteristics necessary for a particular study point of view or depending upon the application area. Analysis of big-data and inferring knowledge from it is no means an easy task. Cloud computing seems to be a solution to this end. Therefore, various cloud computing techniques

that has been developed is emphasized in third section. Chapter 9 gives a brief overview on big data, how to process data intensive applications, current techniques, and technologies. One way of looking at big data is that it represents the large and rapidly growing volume of information that is mostly untapped by existing analytical applications and data warehousing systems. The actual technologies used will depend on the volume, variety of data, and complexity of the analytical processing required by the business. It also depends on the capabilities provided by vendors for managing, administering, and governing the enhanced environment. These capabilities are important for product evaluation and are stressed in this chapter.

Cloud computing has gained popularity because of low cost involved setup, ease of resource configuration and maintenance. The increase in the number of cloud providers in the market has led to availability of a wide range of cloud solutions offered to the customers. These solutions are based on various cloud architectures. But, they are incompatible with each other. Also, not a single provider offers all services to the end users. As a result, cloud providers force cloud customers to decide the design and deployment at the early stages of software development. One of the major issues of this paradigm is the applications and services hosted with a specific cloud provider are locked to their specific implementation technique and operational methods. Therefore, migration from one service provider to another is a tedious task. Hence a way to provide portability of applications across multiple clouds is a major concern. According to the literature very few efforts have been made in order to propose a unified standard for cloud computing. Chapter 10 aims to sketch the architecture of DSkyL that provides a way for reducing the cloud migration efforts.

Growth of business mainly depends on competitiveness that includes many factors. In addition, escalating malware and malicious content have created lot of pressure on business expansion. Also, ever increasing data volumes at off-site, and greater than ever use of content rich applications are mandating organizations to optimize their network resources. Trends such as virtualization and cloud computing further emphasize this requirement in the current era of big data. To assist this process, companies are increasingly relying on a new generation of wide area network optimization techniques, appliances, controllers, and platforms. Chapter 11 provides an overview of WAN optimization, tools, techniques, controllers, appliances, and the solutions that are available for cloud based big data analytics.

E-governance is the delivery of online government services that provides the opportunity to increase citizen access to government, reduce government bureaucracy, increase citizen participation in democracy and enhance agency responsiveness to citizens needs. This improves efficiency by reducing the time spent upon manual tasks, providing rapid online responses, and improvements in organizational competitiveness within public sector organizations. Chapter 12 presents a case study on cloud based e-governance solutions.

Many of the researchers in different organizations across the globe have started research in big-data analysis, computational intelligence, and cloud computing. To keep abreast with this development in a cohesive manner, we strove to keep the book

reader friendly. The main objective is to bring most of the major developments in the above mentioned area in a precise manner, so that it can serve as a handbook for many researchers. Also, many of the universities have introduced this topic as a course at the postgraduate level. We trust and hope that this edited book will help the researchers, who have interest in big-data analysis, cloud computing, and its applications to keep insight into recent advances and their importance in real life applications.

D.P. Acharjya
VIT University, India

Dr. Satchidananda Dehuri
F M University, India

Dr. Sugata Sanyal
Research Advisor, TCS, Mumbai
India, Professor (Retired):
April, 2015 School of Technology & Computer Science, TIFR (1973-2012)

Acknowledgment

It is with great sense of satisfaction that we present our edited book and wish to express our views to all those who helped us both direct and indirect way to complete this project. First and foremost, we praise and heart fully thanks the almighty God, which has been unfailing source of strength, comfort and inspiration in the completion of this project.

While writing, contributors have referred to several books and journals, and we take this opportunity to thank all those authors and publishers. We are extremely thankful to the reviewers for their constant support during the process of evaluation. Special mention should be made of the timely help given by different persons during the project work, those whose names are not mentioned here. Last but not the least, we thank the series editors Meng-Hiot Lim, Yew-Soon Ong and the production team of **Springer-Verlag, USA** for encouraging us and extending their cooperation and help for a timely completion of this edited book. We trust and hope that it will be appreciated by many readers.

Contents

Part I
Theoretical Foundation of Big Data Analysis

"Atrain Distributed System" (ADS): An Infinitely Scalable Architecture for Processing Big Data of Any 4Vs

Ranjit Biswas

Abstract. The present day world dealing with big data (expanding very fast in 4Vs: Volume, Varity, Velocity and Veracity) needs **N**ew advanced kind of logical and physical storage structures, **N**ew advanced kind of heterogeneous data structures, **N**ew mathematical theories and **N**ew models: all these four together we call by 4Ns. As on today, the 4N-set lagging behind in the race with 4V-set. If 4V-set continues its dominance over the 4N-set with respect to time, then it will be difficult to the world to think of "BIG DATA: A Revolution That Will Transform How We Live, Work, and Think". The main objective of this chapter is to report the latest development of an easy and efficient method for processing big data with 4Ns. For processing of giant big data in 4Vs, neither any existing data structure alone nor any existing type of distributed system alone is sufficient. Even the existing network topologies like tree topology or bus/ring/star/mesh/hybrid topologies seem to be weak topologies for big data processing. **For a success, there is no other way but to develop 'new data structures' for big data, 'new type of distributed systems' for big data having a very fast and tremendous extent of mutual compatibility and mutual understanding with the new data structures, 'new type of network topologies' for big data to support the new distributed system, and of course 'new mathematical/logical theory' models for big data science. The next important issue is how to integrate all these 'new' to make ultimately a single and simple scalable system to the laymen users with a new simple 'big data language'**. With these views in mind, a special type of distributed system called by 'Atrain Distributed System' (ADS) is designed which is very suitable for processing big data using the heterogeneous data structure r-atrain for big data (and/or the homogeneous data structure r-train for big data). Consequently, new network topologies called by 'multi-horse cart' topology and 'cycle' topology are introduced to support the new ADS. An ADS could be a 'uni-tier ADS' or a 'Multi-tier ADS'. Scalability upto any extent of our desire both in breadth

Ranjit Biswas
Department of Computer Science and Engineering,
Jamia Hamdard University, Hamdard Nagar, New Delhi-110062, India
e-mail: ranjitbiswas@yahoo.com

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_1

(horizontally) and depth (vertically) can be achieved in ADS by a simple process and this is the unique and extremely rich merit of ADS to challenge the 4Vs of big data. Where r-atrain and r-train are the fundamental data structures for processing big data, the data structures 'heterogeneous data structure MA' and 'homogeneous data structure MT' are higher order data structures for the processing of big data including temporal big data. Both MA and MT can be well implemented in multi-tier ADS. In fact 'Data Structures for Big Data' is to be regarded as a new subject, not just a new topic in big data science. The classical Matrix Theory has been studied by the world with unlimited volume of applications in all branches of Science, Engineering, Statistics, Optimization, Numerical Analysis, Computer Science, Medical Science, Economics, etc. The classical matrices are mainly two dimensional having the elements from the set of real numbers in most of the cases. An infinitely scalable notion of 'solid matrix' (SM) i.e. n-dimensional hyper-matrix, is introduced as a generalization of the classical matrix, and also introduced the notion of 'solid latrix' (SL). In the 'Theory of Solid Matrices', the corresponding matrix-algebra is developed explaining various operations, properties and propositions on them for the case while the elements are objects of the real region RR. As a generalization of the n-SM, another new notion called by 'solid hematrix' (SH) (solid heterogeneous matrix i.e. n-dimensional hyper-matrix with heterogeneous data) is introduced. A method is proposed on how to implement Solid Matrices, n-dimensional arrays, n-dimensional larrays etc. in a computer memory using the data structures MT and MA for big data. The combination of r-atrain (r-train) with ADS and the combination of MA (MT) with multitier ADS can play a major role in a new direction to the present data-dependent giant galaxies of organizations, institutions and individuals to process any big data irrespective of the influence of 4Vs.

1 Introduction

The present world of big data [2-4, 6-8, 10, 14, 15, 19] are expanding very fast in 4Vs: Volume, Varity, Velocity and Veracity, and also in many more directions. How to deal with big data, how to process big data in an efficient way within limited resources, etc. are of major concern to the computer scientists now-a-days. In particular, the 'Velocity' at which the big data have been expanding (or, the 4Vs in which big data have been expanding very fast in the present day world) does not have a one-to-one matching with the 'Velocity' at which the new hardware or new software or new mathematical theories or new models are being developed by the scientists. Let us designate the following two sets by 4V-set and 4N-set:

(i) 4V-set = {Volume, Varity, Velocity and Veracity}, and
(ii) 4N-set = {**N**ew Theories, **N**ew Hardware, **N**ew Software, **N**ew Models}.

It is obvious that big data can be efficiently processed by a faster development of the 4N-set only. If 4V-set continues its dominance over 4N-set with respect to

time, then it will be difficult to the world to think of "BIG DATA: A Revolution That Will Transform How We Live, Work, and Think"[13]. As on today, the 4N-set lagging behind in the race with 4V-set. For a proper encounter and success, there is no other way but to develop 'new data structures' exclusively for big data, 'new type of distributed systems' exclusively for big data having a very fast and tremendous extent of mutual compatibility and mutual understanding with the new data structures, 'new type of network topologies' for big data to support the new distributed system, and ofcourse 'new mathematical/logical theory models' etc. to enrich the big data science which can ultimately provide the new requirements for the development of new hardware/software. In fact 'Data Structures for Big Data' [8] is to be regarded as a new subject, not just a new topic in big data science. This chapter presents a complete package of new 4Ns to challenge the 4Vs. The chapter is organized into ten sections: Section 2 describes the homogeneous data structure r-Train for homogeneous big data and the heterogeneous data structure r-Atrain for heterogeneous big data [3, 4], the first appropriate data structures exclusively designed for big data; Section 3 and Section 4 present the Theory of Solid Matrices/Latrices with their algebraic properties; Section 5 describes the homogeneous data structure MT; Section 6 introduces two logical storage models Hematrix/Helatrix for heterogeneous big data; Section 7 describes a new type of infinitely scalable distributed system called by 'ADS' exclusively designed for big data, introducing two new type of network topologies called by 'multi-horse cart' topology and 'cycle' topology; Section 8 explains in details how the data structures atrain/train can process big data in an ADS; Section 9 describes the heterogeneous data structure MA in ADS; and Section 10 presents an overall conclusion.

Upto Section 5 in this chapter, the implementation of big data using the data structures train or atrain or MT are shown in autonomous computer systems only for the sake of easy understanding, and then the progress is made to the newly proposed distributed system ADS with new type of network topologies. ADS is easily scalable upto any desired extent both horizontally (in breadth) and vertically (in depth) to challenge any amount of 4Vs of big data of any momentum. The most important issue like "how to integrate all these 4Ns to make ultimately a single and simple system to the laymen users at their ground levels" is an inbuilt solution provided by the rich architecture of ADS for big data. The proposed 4Ns as an integrated team has an excellent team spirit due to 100% compatibility and understanding among them.

2 "r-Train" (train) and "r-Atrain" (atrain): The Data Structures for Big Data

The existing classical data structures available in the literatures are not sufficiently rich and potential enough to deal with big data. Consequently, the homogeneous data structure 'r-Train' (train) for homogeneous big data and the heterogeneous data structure 'r-Atrain' (atrain) for heterogeneous big data are introduced by Biswas[3, 4]. These two data structures are then further extended to define the data structures

MT and MA for big data if the storage structures are helatrix or hematrix. The architecture of a new type of distributed system 'ADS' is exclusively designed for processing big data. Both the data structures train and atrain are 100% compatible with the new distributed system 'ADS' being scalable upto any desired amount both horizontally (in breadth) and vertically (in depth) for processing big data, in particular to coup easily with 4Vs of any momentum.

2.1 Larray

Larray is not a valid word in dictionary. Larray stands for "**L**ike **ARRAY**". A larray is like an array of elements of identical datatype, but with an additional property that zero or more number of elements may be null element (i.e. no element, empty). Denote the null element by ε. Assuming that ε is of the same datatype, the memory space reserved for it will be same as that required by any other element of the larray. The number of elements (including ε) in a larray is called the length of the larray.

Example of larrays:

(i) a = $< 5, 2, \varepsilon, 13, 25, \varepsilon, 2, >$
(ii) b = $< 6, 9, 8 >$
(iii) c = $< \varepsilon >$
(iv) d = $< \varepsilon, \varepsilon, \varepsilon, >$
(v) e = $< 2.8, \varepsilon, \varepsilon, \varepsilon, \varepsilon >$
(vi) f = $< >$, which is called the **empty larray** denoted by ϕ.

If all the elements of a larray are ε, then it is called a **null larray**. Any null larray is denoted by θ. Clearly θ could be of any length (any non-negative integer) and hence null larray is not unique. As a special instance, the empty larray ϕ defined above is also a null larray of length 0 (zero), i.e. with 0 (zero) number of ε elements. Thus empty larray and null larray are not the same concept in our work here. Empty larray is a unique object unlike null larray. In our theory, an object like k = $< , , , >$ is an undefined object and hence should not be confused with empty larray or null larray. In the above examples, the larrays c and d are null larrays, but e is not. The larray f is empty larray but c, d are not. It is obvious that two distinct larrays may contain ε elements of different datatypes, because ε elements of a larray are of the same datatype analogous to that of the non-ε elements of that larray (by virtue of its construction). For example, each ε element in the larray a above requires 2 bytes in memory, whereas each ε element in the larray e above requires 4 bytes in memory. But for the case of larray d, we can not extract any idea about the datatype of the ε elements of d just by looking at it. In such case we must have information from outside. Each ε element in the larray d above requires as many bytes as defined at the time of creation of d by the concerned programmer. By using the name z of a larray z, we do also mean the address of the larray z in the memory, analogous to the case in arrays.

2.2 *Homogeneous Data Structure "r-Train" (train) for Homogeneous Big Data*

The data structure 'r-train' (train, in short) is a new and a robust kind of dynamic homogeneous data structure which encapsulates the merits of the arrays and of the linked lists, and at the same time inherits a reduced amount of their characteristic demerits. By the nature 'dynamic' we mean that it changes over time, it is not like static. Apparently it may seem that the homogeneous data structure r-train is of hybrid nature, hybrid of linked list and array; but construction-wise (by virtue of its architecture) it is much more. Using the data structure 'r-train', any big number of homogeneous data elements can be stored and basic operations like insertion/deletion/search etc. can be executed very easily. In particular parallel programming can be done very easily in many situations with improved time complexities and performance efficiency as shown by Alam[1]. The notion of r-train is not a direct generalization of that of the linked list. It is not just a linked list of arrays as it looks so apparently, but something more. However, every linked list is an example of 1-train (i.e. r-train with r = 1). The main aim of introducing the data structure 'r-train' is how to store a large, very large or big array in memory in an alternative way successfully, and how to perform the fundamental operations of data structures efficiently for big data. For small size arrays it is obvious that the data structure train should not be used as an alternative to array. The data structure train dominates the data structures array and linked list in merits. It is neither a dynamic array [9] nor a HAT [16].

2.2.1 Coach of a r-Train

By a **coach** C we mean a pair (A, **e**) where A is a non-empty larray (but could be a null larray) and **e** is an address in the memory. Here **e** is basically a kind of link address. Its significance is that it says the address of the immediate next coach, and thus it links two consecutive coaches. If the coach C is the last coach then the address field **e** will be put equal to an invalid address, otherwise **e** will be the address of the next coach. There is no confusion in differentiating the two objects **e** and ε. A coach can be easily stored in memory inside which the data **e** resides next to the last element of the larray A in the memory. Logical diagram of a coach of r-train is available in Fig. 1, 2, 3, 4, 5 and 6.

Suppose that the larray A has m number of elements in it. If each element of A is of size x bytes and if the data **e** require two bytes to be stored in memory, then to store the coach C in memory, exactly (m.x+2) number of consecutive bytes are required, and accordingly the coach (GETNODE) be created by the concerned programmer.

2.2.2 Status of a Coach and Tagged Coach (TC) of a r-Train

The **status** s of a coach in a r-train is a non-negative integer real time variable which is equal to the number of ε elements present in it (i.e. in its larray) at this real point of time. Therefore, $0 \leq s \leq r$. The significance of the variable s is that it informs us about the exact number of free spaces available in the coach at this point of time. If there is no ε element in the larray of the coach C, then the value of s is 0 at this real point of time. If C = (A, **e**) is a coach, then the corresponding tagged coach (TC) is denoted by the notation (C, s) where s is the status of the coach. This means that C is a coach tagged with an information on the total amount of available free space (here it is termed as elements) inside it at this time.

For example, consider the larrays of section 2.1. Clearly a TC with the larray a will be denoted by (C, 2), a TC with the larray b will be denoted by (C, 0), a TC with the larray d will be denoted by (C, 4), and so on. In our discussion here, without any confusion, the two statements "there is no ε element in the larray" and "there is no ε element in the coach" will have identical meaning where the larray is that of the coach.

2.2.3 r-Train (Train)

The data structure **'r-train'** is a new robust, flexible and homogeneous data structure which is very useful if there are big data of homogeneous datatype [3, 4]. A r-train is basically a linked list of tagged coaches. This linked list is called the'pilot' of the r-train. The pilot of a train is thus, in general, a linked list. But, if we are sure that there will be no requirement of any extra coach in future, then it is better to implement (to code) the pilot as an array. The number of coaches in a r-train is called the 'length' of the r-train which may increase or decrease time to time. A 'r-train' may also be called in short by the name **'train'** if there is no confusion. A r-train T of length l (> 0) will be denoted by the notation: $T = < (C_1, s_{C_1}), (C_2, s_{C_2}), (C_3, s_{C_3}), \ldots, (C_l, s_{C_l}) >$, where the coach C_i is $(A_i, \mathbf{e_i})$ with A_i being a larray of length r, e_i being the address of the next coach C_{i+1} (or an invalid address in case C_i is the last coach) and s_{C_1} being the status of the coach C_i, for i = 1, 2, 3, ..., l.

For a r-train, **START** is the address of the pilot (viewing its implementation in memory as a linked list). Thus **START** points at the data C_1 in memory. The length l of the pilot could be any natural number, but the larrays of the TCs are each of fixed length r which store data elements (including ε elements) of common datatype where r is a natural number. Thus, by name one could see that 1-train, 60-train, 75-train, 100-train, etc. are few instances of r-train, where the term 0-train is undefined.

The notion of the data structure r-train is a new but very simple data structure, a type of 2-tier data structure, having very convenient methods of executing various fundamental operations like insertion, deletion, searching etc. for big data, in particular for parallel computing as shown by Alam in [1]. The most important characteristic of the data structure r-train is that it can store x number of elements

of identical datatype even if an array of size x of same elements can not be stored in memory at some moment of time. And also, the data can be well accessed by using the indices.

In a r-train, the coach names C_1, C_2, C_3, …, C_l do also mean the addresses (pointers) serially numbered as in case of arrays, for example: the array z = (5, 9, 3, 2) can be easily accessed by calling its name z only.

Status of a coach in the data structure r-train reflects the availability of a seat (i.e. whether a new data element can be stored in this coach now or not). Status may vary with time. Each coach of a r-train can accommodate exactly r number of data elements serially numbered, each data element being called a **passenger**. Thus each coach of a r-train points at a larray of r number of passengers. *By definition, the data e_i is not a passenger for any i because it is an address and not an element of the larray*.

Consider a coach $C_i = (A_i, \mathbf{e_i})$. In the larray $A_i = < e_{i1}, e_{i2}, e_{i3}, \ldots, e_{i(r-1)}, e_{ir} >$, the data element e_{ij} is called the 'j^{th} passenger' or "j^{th} data element' for j = 1, 2, 3, 4, …, r. Thus we can view a r-train as a linked list of larrays. Starting from any coach, one can visit the inside of all the next coaches (However in ADS with doubly linked twin address or in case of circular trains, one could visit the previous coaches too which are explained later in subsection 8.1 and 8.2 here). In this sense the r-train is a forward linear object. The r-train is neither a dynamic array nor a HAT. It has an added advantage over HAT that starting from one data element, all the next data elements can be read well without referring to any hash table or the pilot.

Example 1: Any classical linked list is a 1-train where all the coaches are having a common status equal to zero (as each coach accommodates one and only one passenger). However the converse is not necessarily true, i.e. if each coach of a train is having status 0, it does not necessarily mean that the train is a 1-train (linked list).

Example 2: Consider a 3-train T of length 3 given by T = < (C_1, 1), (C_2, 0), (C_3, 1) > , where C_1 = < 4, ε, 7, $\mathbf{e_1}$ >, and C_2 = < 2, 9, 6, $\mathbf{e_2}$ >, and C_3 = <6, 2, ε, an invalid-address>.

Here $\mathbf{e_1}$ is the address of coach C_2 (i.e. address of larray A_2), and $\mathbf{e_2}$ is the address of coach C_3 (i.e. address of larray A_3). Since it is a 3-train, each coach C_i can accommodate exactly three passenger (including ε element, if any). In coach C_1, the larray is A_1 = <4, ε , 7> which means that the first passenger is the integer 4, second passenger is ε (i.e. this seat is vacant now), and the last/third passenger is the integer 7; the data $\mathbf{e_1}$ being the address of the next coach C_2. Thus, T is a larray of three TCs which are (C_1, 1), (C_2, 0), and (C_3, 1).

The logical diagram of the above mentioned 3-train T is shown in Fig. 1 where data in coaches are to be read clockwise, i.e. starting from e_{11} for the coach C_1, from e_{21} for the coach C_2 and from e_{31} for the coach C_3. Fig. 2 shows a r-train with 30 coaches:

Fig. 3 shows one coach (ith coach) of a 11-train, where data are to be read clockwise starting from e_{i1}:

Full Coach in a r-Train: A coach is said to be a **full coach** if it does not have any passenger ε, i.e. if its status s is 0 (i.e. no seat is vacant now). The physical significance of the variable 'status" is "availability of space at this point of time" for

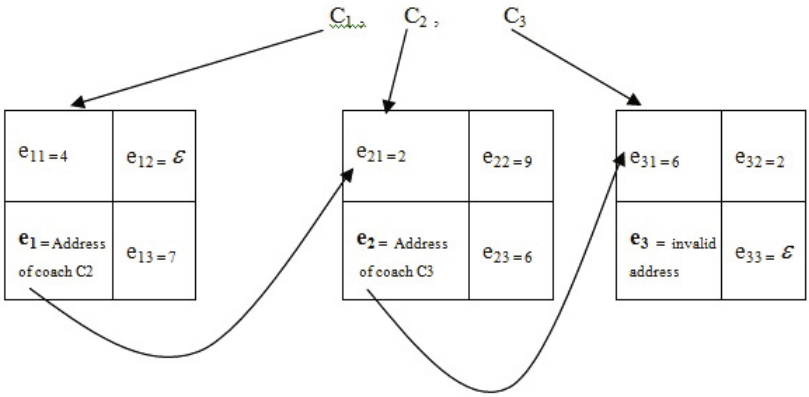


Fig. 1 A 3-train with 3 coaches

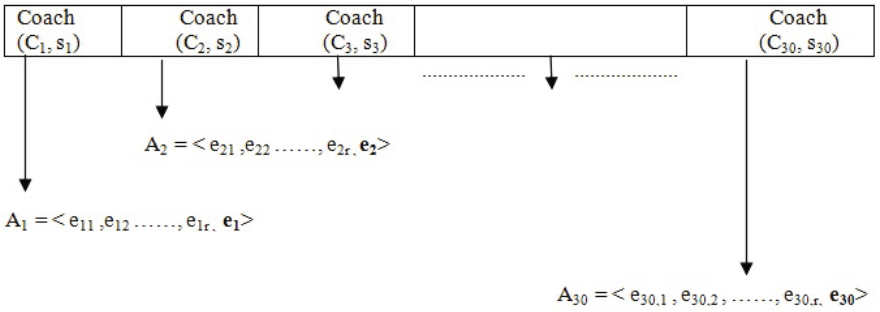


Fig. 2 A r-train with 30 coaches

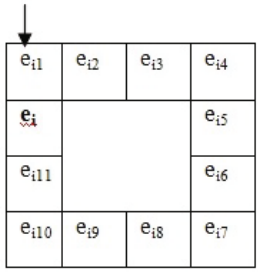


Fig. 3 A Coach C_i of a 11-train

storing a data element. In a full coach, we can not insert any more data (passenger) at this point of time (however, may be possible at later time). Clearly, the coaches of any linked list (i.e. 1-train) are all full, and always full. See the example in Fig. 4.

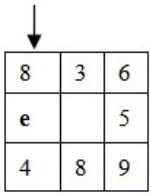


Fig. 4 A full coach of a 7-train

Empty Coach in a r-Train: A coach in a r-train is said to be an **empty coach** if every passenger of it is ε, i.e. if the corresponding larray is a null larray. Thus for an empty coach of a r-train, the status is equal to r (all the seats are vacant). Fig. 5 shows an empty coach of a 13-train.

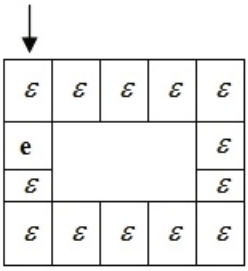


Fig. 5 An empty coach of a 13-train

A coach may be sometimes neither empty nor full (see Fig. 6 below).

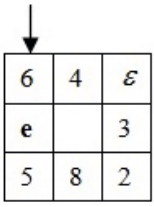


Fig. 6 A coach in a 7-train

2.2.4 A r-Train in Memory

Consider the data: 5.4, 6.3, 0.8, 1.8, 6.2, 9.9, 6.4, 2.1, 1.7, 0.2, 2.4 which are stored in the Data Segment of the 8086 memory using the data structure 3-train, as shown in Table.1, starting from START = 00B2h.

Table 1 A r-Train in 8086 memory

Address	Memory Content	Size
	...	
	X (an invalid address)	2 bytes
	ε	4 bytes
	2.4	4 bytes
EA08h	0.2	4 bytes
	008Ch	2 bytes
	9.9	4 bytes
	6.2	4 bytes
C0ABh	1.8	4 bytes
	...	
	C0ABh	2 bytes
	0.8	4 bytes
	6.3	4 bytes
10B8h	5.4	4 bytes
	...	
	1	2 bytes
00BDh	C_4 = EA08h	2 bytes
	0	2 bytes
00BAh	C_3 = 008Ch	2 bytes
	0	2 bytes
00B6h	C_2 = C0ABh	2 bytes
	0	2 bytes
START = 00B2h	C_1 = 10B8h	2 bytes
	...	
	EA08h	2 bytes
	1.7	4 bytes
	2.1	4 bytes
008Ch	6.4	4 bytes

This 3-train is T = $<$ (10B8h, 0), (C0ABh, 0), (008Ch, 0), (EA08h, 1) $>$ which is of length 4 where the coach C_1 begins from the address 10B8, the coach C_2 begins from the address C0ABh, the coach C_3 begins from the address 008C, the coach C_4 begins from the address EA08Ch. Here START = 00B2h, i.e. the pilot of this 3-train is stored at the address 00B2h. Also in this example, the pilot is implemented as an array, not using linked list.

2.2.5 Fundamental Operations on the Data Structure 'r-Train'

The three fundamental operations on the data structure r-train are 'insertion', 'deletion' and 'search' which are defined below:

2.2.5.1 Insertion

There are two types of insertion operation in the data structure r-train: -

i) insertion (addition) of a new coach in a r-train.
ii) insertion of a data element (passenger) in a coach of a r-train.

(i) Insertion of a New Coach in a r-Train

Insertion of a new coach can be done at the end of the pilot, nowhere else. Consider the r-train

$$T = < (C_1, s_{C_1}), (C_2, s_{C_2}), (C_3, s_{C_3}), \ldots, (C_k, s_{C_k}) >$$

with k number of coaches, where the coach $C_i = (A_i, \mathbf{e_i})$ for i = 1, 2, 3, ..., k. After insertion of a new coach, the updated r-train immediately becomes the following r-train:

$$T = < (C_1, s_{C_1}), (C_2, s_{C_2}), (C_3, s_{C_3}), \ldots, (C_k, s_{C_k}), (C_{k+1}, r) >.$$

Initially at the time of insertion, we create C_{k+1} as an empty coach (i.e. with status = r) which is likely to get filled-up with non-ε passengers (data) later on with time. For insertion of a new coach C_{k+1} in a r-train, we need to do the following:

(i) GETNODE for a new coach C_{k+1}
(ii) update the pilot (linked list)
(iii) $\mathbf{e_i}$ in C_k is to be updated and to be made equal to the address C_{k+1}
(iv) set $e_{k+1,j} = \varepsilon$ for j = 1, 2, ..., r
(v) set $\mathbf{e_{k+1}}$ = an invalid address.
(vi) set s_{C_k} = r.

(ii) Insertion of an Element x inside the Coach $C_i = (A_i, e_i)$ of a r-Train

Insertion of an element (a new passenger) x inside the coach C_i is feasible if x is of same datatype (like other passengers of the coach) and if there is an empty space available inside the coach C_i.

If status of C_i is greater than 0 then data can be stored successfully in the coach, otherwise insertion operation fails here. After each successful insertion, the status s of the coach is to be updated by doing s = s-1.

For insertion of x, we can replace the lowest indexed passenger ε of C_i with x.

2.2.5.2 Deletion

There are two types of deletion operation in the data structure r-train: -

(i) Deletion of a data element ($\neq \varepsilon$) from any coach of the r-train.
(ii) Deletion of the last coach C_i, if it is an empty coach, from a r-train.

(i) Deletion of a Data e_{ij} ($\neq \varepsilon$) from the Coach C_i of a r-Train

Deletion of e_i from the coach C_i is not allowed as it is the link, not a passenger. But we can delete a data element e_{ij} from the coach C_i. Deletion of a data (passenger)

from a coach means replacement of the data by an ε element (of same datatype). Consequently, if $e_{ij} = \varepsilon$, then the question of deletion does not arise. Here it is pre-assumed that e_{ij} is a non-ε member element of the coach C_i. For j = 1, 2, …, r, deletion of e_{ij} is done by replacing it by the null element ε, and updating the status s by doing s = s+1. Deletion of a data element (passenger) does not effect the size r of the coach. For example, consider the tagged coach (C_i, m) where $C_i = <e_{i1}, e_{i2}, e_{i3}, e_{i4}, \ldots, e_{ir}>$. If we delete e_{i3} from the coach C_i, then the updated tagged coach will be (C_i, m+1) where $C_i = <e_{i1}, e_{i2}, \varepsilon, e_{i4}, \ldots, e_{ir}>$.

(ii) Deletion of the Last Coach C_i from a r-Train

Deletion of coaches from a r-train is allowed from the last coach only and in backward direction, one after another. We advertently avoid the deletion of any interim coach from a r-train while dealing with big data (although deletion of interim coach in a r-train can not be well coded by the programmers). The last coach C_i can be deleted if it is an empty coach (as shown in Fig. 7):

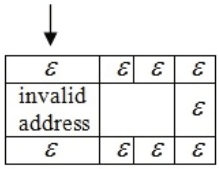


Fig. 7 A last coach of a 9-train which can be deleted

If the last coach is not empty, it can not be deleted unless its all the passengers are deleted to make it empty. To delete the empty last coach C_i, of a r-train, we have to do the following actions: -

(i) update e_{i-1} of the coach C_{i-1} by storing an invalid address in it.
(ii) delete (C_i, r) from the r-train

$$T = <(C_1, s_{C_1}), (C_2, s_{C_2}), (C_3, s_{C_3}), \ldots, (C_{i-1}, s_{C_{i-1}}), (C_i, r)>$$

and get the updated r-train

$$T = <(C_1, s_{C_1}), (C_2, s_{C_2}), (C_3, s_{C_3}), \ldots, (C_{i-1}, s_{C_{i-1}})>.$$

(iii) update the pilot.

2.2.5.3 Searching for a Passenger (data) in a r-Train of Length k

Searching a data x in a r-train T is very easy.

(i) If we know in advance the coach number C_i of the passenger x, then visiting the pilot we can enter into the coach C_i of the r-train directly and then can read the data-elements $e_{i1}, e_{i2}, \ldots, e_{ir}$ of the larray A_i for a match with x.

(ii) If we do not know the coach, then we start searching from the coach C_1 onwards till the last coach. We need not go back to the pilot for any help.

Here lies an important dominance of the data structure r-train over the data structure HAT introduced by Sitarski [16].

Both BFS and DFS can be implemented in a very easy way as the data structure itself matches the properties of searching technique, in particular the BFS. In case of multi processor system the searching can be done in parallel very fast, which is obvious from the physiology of the data structure r-train. In the next section, the notion of the heterogeneous data structure 'r-Atrain' (Atrain) [3, 4, 8] for heterogeneous big data is discussed.

2.3 r-Atrain (Atrain): A Powerful Heterogeneous Data Structure for Big Data

In a homogeneous data structure the data elements considered are all of the same datatype, like in array, linked list, r-train, etc. In a heterogeneous data structure data elements are of various datatypes as in a 'structure'. In this section the heterogeneous data structure '**r-Atrain**' [3, 4, 8] is discussed. The term '**Atrain**' stands for the phrase '**A**dvanced **train**', because in the construction of r-Atrain we incorporate few advanced features (not available in r-train) which make it suitable to deal with heterogeneous data, in particular for heterogeneous big data. The data structure r-Atrain is not a substitute or a competitor of the data structure r-train to the programmers or developers, although any r-train is a trivial instance of a r-atrain. For working with homogeneous data, r-train is suitable while r-atrain is not. But for working with heterogeneous data, r-atrain is suitable while r-train can not be applicable. The term "train" has been coined from the usual railways transportation systems because of a lot of similarities in nature of the object r-train with the actual train of railways systems; and analogously the terms coach, passenger, availability of seats, etc. were used in the way of constructing the data structure r-train.

Now, let us think of more reality here. In railways transport system we see that in most of the trains one or more coaches are created for pantry, one or more coaches are for accommodating postal mails, one or more are for accommodating booked luggages, one or more are for accommodating goods, one or more exclusively for ladies, etc. and most of the coaches are for accommodating passengers. Different coaches are there for accommodating different types of contents, i.e. for accommodating heterogeneous types of contents. With this real example, it is encouraged to develop a new data structure 'r-Atrain' where coaches may accommodate data of various datatypes, but no coach is allowed to accommodate data of different datatypes. The datatype in a r-atrain may vary from coach to coach (unlike in r-train), but in a coach all data must be homogeneous internally i.e. of identical datatype. Thus each coach is homogeneous, but the atrain is heterogeneous.

2.3.1 Code of a Datatype and CD-Table

A table is to be created by the user (i.e. by the concerned organization) to fix unique integer code for each datatype which are under use in the organization. This is not an absolute set of codes to be followed universally by every organization, but it is a local document for the concerned organization. For different organizations, this table could be different. But once it is fixed by an organization it should not be altered by this organization, except that addition of new datatypes and corresponding codes which may be decided and be incorporated in the table at any stages later retaining the existing records. This table is called **Code of Datatype Table** or **CD-Table** (in short). A sample CD-Table of a hypothetical organization is shown below for the sake of understanding:

Table 2 A hypothetical example of a **CD-Table** of an organization

Sr. No.	Datatype	Space required in bytes (n)	Code of Datatype (c)
1	Character	1	0
2	Integer	2	1
3	Real	4	2
4	String-1	10	3
5	String-2	20	4
6	String-3	50	5
7	File-1	100 KB	6
8	File-2	1 MB	7
9	File-3	10 MB	8
10	File-4	25 MB	9
11	...	...	...
12	...	...	...
13	...	...	...

It may be noted that for any organization, for the datatypes character, integer, boolean, etc. the individual space requirement respectively are absolutely fixed. But for a particular organization, for the datatypes String-1, String-2 and String-3 (String type appearing thrice in this case) the space requirement in the above table has been fixed at 10 bytes for one kind, 20 bytes for another and 50 bytes for another kind, and so on (which could be different for another organization. These codes will vary from organization to organization). Similarly there are four types of file: File-1, File-2, File-3 and File-4, in the CD-Table and the space requirement for them have been fixed at 100 KB, 1 MB, 10 MB and 25 MB respectively, fixed by choice of the concerned organization (developers). The datatype of any file of size 100 KB or less will be regarded as File-1 with code 6, the datatype of any file of size more than 100 KB but less than or equal to 1MB will be regarded as File-2 with code 7, and so on.

2.3.2 Coach of a r-Atrain

By a **coach** C of a r-atrain we mean a pair (A, **e**) where A is a non-empty larray (could be a null larray) and **e** is an address in the memory. Here **e** is basically a kind of **link address**. Its significance is that it says the address of the immediate next coach, and thus it links two consecutive coaches in the r-atrain. If the coach C is a single coach or the last coach then the address field **e** will be put equal to an invalid address, otherwise **e** will be the address of the next coach.

A coach can be easily stored in memory where the data **e** resides next to the last element of the larray A in the memory. Logical diagram of a coach in a r-atrain is shown in Fig. 8, 9, 10, 11, 12 and 13, and ofcourse the coaches in Fig. 1, 2, 3, 4, 5, 6 are also the instances of coaches in r-atrain. A coach in a r-atrain can store homogeneous data only, not heterogeneous data. But different coaches of a r-atrain store data elements of different datatypes. Thus the data structure r-atrain is called a heterogeneous data structure, exclusively designed for processing heterogeneous big data. For constructing a coach for a r-atrain in an organization, we must know in advance the datatype of the data to be stored in it. For this we have to look at the CD-Table of the organization, and reserve space accordingly for r number of data for the coach. The CD-Table may be expanded by the concerned organization.

Suppose that the larray A has r number of elements in it of a given datatype. If each element of A is of size x bytes (refer to CD-Table) and if the data **e** requires two bytes to be stored in memory, then to store the coach C in memory exactly (r.x+2) number of consecutive bytes are required, and accordingly the coach be created by the programmer (concerned organization). In our discussion henceforth, by the phrase "datatype of a coach" we shall always mean the datatype of the data elements of the coach. A coach stores and can store only homogeneous data (i.e. data of identical datatype), but datatype may be different for different coaches in a r-atrain. And in this way a r-atrain can accommodate heterogeneous big data.

2.3.3 Status of a Coach and Tagged Coach (TC) in a r-Atrain

The notion of 'status of a coach' and 'tagged coach' for r-train data structure are discussed earlier in subsection 2.2. earlier. The **status** s of a coach in a r-atrain is a pair of information (c, n), where c is a non-negative integer variable which is the code of datatype (with reference to the concerned CD-Table) of the data to be stored in this coach and n is a non-negative integer variable which is equal to the number of ε elements present in it (i.e. in its larray) at this point of time. Therefore, $0 \leq$ n $\leq$ r. In the status s = (c, n) of a coach, the information c is called the **"code-status"** of the coach and the information n is called the **"availability-status"** of the coach at this time. The significance of the variable n is that it informs us about the exact number of free spaces available in the coach at this point of time. If there is no ε element at this time in the larray of the coach C, then the value of n is 0. Thus, without referring to the CD-Table, the status of a coach can not be and should not be fixed.

If C = (A, **e**) is a coach in a r-atrain, then the corresponding tagged coach (TC) is denoted by the notation [C, s], where s = (c, n) is the status of the coach. This means that C is a coach tagged with the following two information in its status:

(i) one signifying the datatype of the data of the coach as per CD-Table, and
(ii) the other reflects the total amount of available free spaces (here it is termed as ε elements) inside the coach at this time.

For example, consider the larrays of section 2.1 and the CD-Table (Table.2) of section 2.3.1. Clearly a TC with the larray a will be denoted by [C, (1, 2)], a TC with the larray b will be denoted by [C, (1, 0)], a TC with the larray d will be denoted by [C, (3, 4)] assuming that this coach will accommodate strings only, and so on.

In our discussion here, without any confusion, the two statements "there is no ε element in the larray" and "there is no element in the coach" will have identical meaning where the larray is that of the coach (as followed analogously in section 2.2 earlier to introduce the data structure train).

2.3.4 Heterogeneous Data Structure r-Atrain

A r-atrain is basically a linked list of tagged coaches of various datatypes. This linked list is called the 'pilot' of the r-atrain. The implementation of this pilot in memory can also be done using the data structure array in some cases. The pilot of an r-atrain is thus, in general, a linked list. But, if we are sure that there will be no requirement of any extra coach in future, then it is better to implement pilot as an array. The number of coaches in a r-atrain is called the 'length' of the r-atrain which may increase or decrease time to time. A 'r-atrain' may also be called by the name 'atrain', if there is no confusion.

A r-atrain T of length l (> 0) will be denoted by the following notation

$$T = < (C_1, s_{C_1}), (C_2, s_{C_2}), (C_3, s_{C_3}), \ldots, (C_l, s_{C_l}) >$$

where the coach C_i is (A_i, e_i) with A_i being a larray of length r, e_i being the address of the next coach C_{i+1} (or an invalid address in case C_i is the last coach) and s_{C_i} being the status (c_i, n_i) of the coach C_i, for i = 1, 2, 3, ..., l.

For a r-atrain, **START** is the address of the pilot (viewing its implementation in memory as a linked list). Thus **START** points at the coach C_1 in memory. The length l of the pilot could be any natural number, but the larrays of the TCs are each of fixed length r which store data elements (including ε elements) of heterogeneous datatype, although each coach itself can accommodate homogeneous data only, not heterogeneous data, where r is a natural number. Thus, by name, 1-atrain, 64-atrain, 150-atrain, etc. are few instances of r-atrain, where the term 0-atrain is undefined.

The notion of the heterogeneous data structure r-atrain is a new but very simple data structure, a type of 2-tier data structure, having very convenient methods of executing various fundamental operations like insertion, deletion, searching etc. for heterogeneous big data, in particular for parallel computing.

In a r-atrain, the data can be well accessed by using the indices. The coach names $C_1, C_2, C_3, \ldots, C_l$ do also mean the addresses (pointers) serially numbered as in

case of arrays, for example:- the array z = (image-1, image-2, image-3, image-4) can be easily accessed calling by its name z only. The second information in the status of a coach is a dynamic information which reflects the availability of a seat (i.e. whether a valid data element can be stored now in this coach or not) while the first information is always static but can be different for different coaches. The first information 'code-status' of a coach does never alter by virtue of the construction principle of a coach, but the status of this coach may vary with time as the second information 'availability-status' may vary with time dynamically. Every coach of a r-atrain can accommodate exactly r number of homogeneous data elements serially numbered, each data element being called a **passenger**. Thus each coach of a r-atrain points at a larray of r number of passengers. *By definition, the data e_i is not a passenger for any i as it is an address, not an element of the larray.*

Consider a coach $C_i = (A_i, e_i)$ of a r-atrain. In the larray $A_i = <e_{i1}, e_{i2}, e_{i3}, \ldots, e_{i(r-1)}, e_{ir}>$, the data element e_{ij} is called the "j^{th} passenger" or "j^{th} data element" in this coach for j = 1, 2, 3, 4, …, r. Thus we can view a r-atrain as a linked list of heterogeneous larrays. Starting from any coach, one can visit the inside of all the next coaches of the atrain (However in ADS with doubly linked twin address or in case of circular atrains, one could visit the previous coaches too which are explained later in subsection 8.1 and 8.2 here). The r-atrain is neither a dynamic array nor a HAT. It has an added advantage over HAT that starting from one data element, all the next data elements can be read well without referring to any hash table or the pilot.

Example 3: Any linked list is a 1-atrain where the coaches are having a common status (c, 0), c being the code of the datatype. It may be mentioned here that, by default, any heterogeneous data structure can be used as a homogeneous data structure, although not preferred in general.

Example 4: Refer to the CD-Table of section 2.3.1. Consider a 3-atrain T of length 3 given by T = < [C_1, (1, 0)], [C_2, (3, 1)], [C_3, (0, 1)] > , where $C_1 = < 5, 9, 2, e_1 >$, and $C_2 =$ < CALCUTTA, ε, DELHI, e_2 >, and $C_3 =$ <+, #, ε, an invalid-address>. Here e_1 is the address of the coach C_2 (i.e. address of larray A_2) in this 3-atrain, and e_2 is the address of the coach C_3 (i.e. address of larray A_3). Since it is a 3-atrain, each coach C_i can accommodate exactly three passenger (including ε elements, if any). In coach C_1, the status is (1, 0) which means that this coach can accommodate data of integer datatype (with reference to the CD-Table) but there is no free space in this coach at this point of time. The larray is $A_1 =$ <5, 9, 2> which means that the first passenger is the integer 5, second passenger is the integer 9, and the last/third passenger is the integer 2; the data e_1 being the address of the next coach C_2. Thus, T is a larray of three TCs which are [C_1, (1, 0)], [C_2, (3, 1)] and [C_3, (0, 1)].

The logical diagram of this 3-atrain T is shown in Fig. 8 where data in coaches are to be read clockwise starting from e_{11} for the coach C_1, from e_{21} for the coach C_2 and from e_{31} for the coach C_3. Fig. 9 shows a r-atrain with 50 number of coaches:

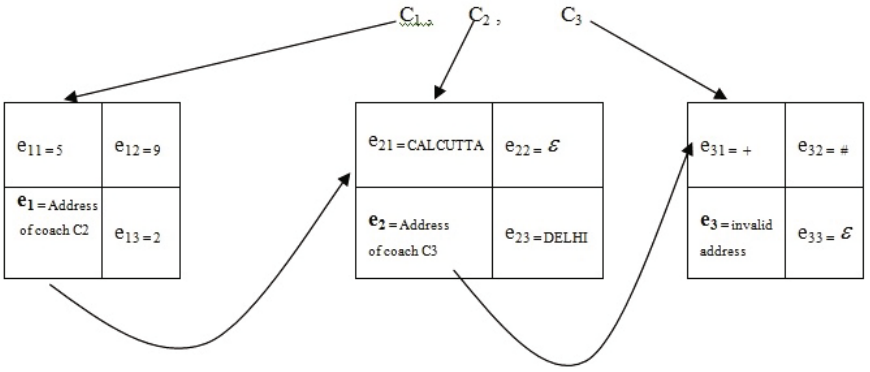


Fig. 8 Logical diagram of 3-atrain

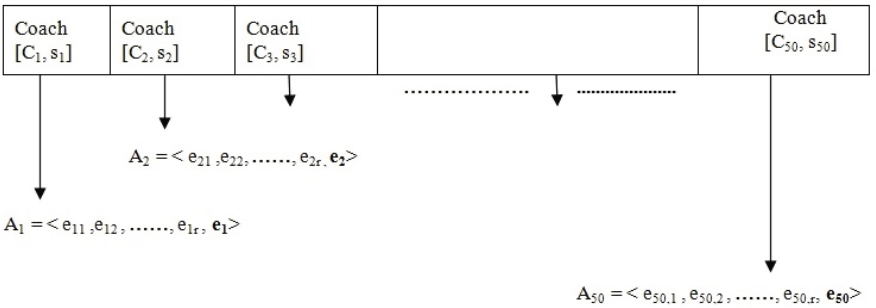


Fig. 9 A r-atrain with 50 coaches

2.3.5 Full Coach in a r-Atrain

A coach in a r-atrain is said to be a **full coach** if it does not have any passenger ε, i.e. if its status s is (c, 0) where c is the code of its datatype.

In a full coach, we can not insert (insertion operation is explained in later section here) any more data (passenger) at this point of time (however, may be possible at later point of time). Fig. 10 shows a full coach of a 7-atrain with status (8, 0) as per CD-Table of Table.2. Clearly, the coaches of any classical linked list (which is kind of 1-atrain) are all full.

2.3.6 Empty Coach in a r-Atrain

A coach in a r-atrain is said to be an **empty coach** if every passenger of it is ε, i.e. if the corresponding larray is a null larray. Thus for an empty coach of a r-atrain, the status is equal to (c, r). Fig. 11 shows an empty coach of a 13-atrain.

A coach may be sometimes neither empty nor full (see Fig. 12).

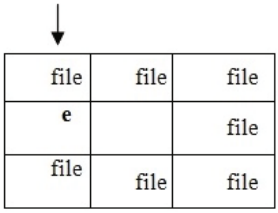


Fig. 10 A full coach in a 7-atrain

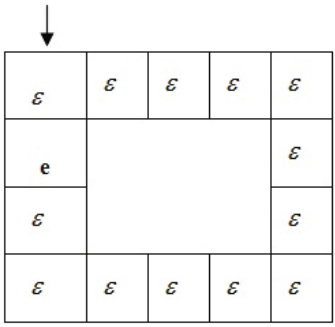


Fig. 11 An empty coach of a 13-atrain

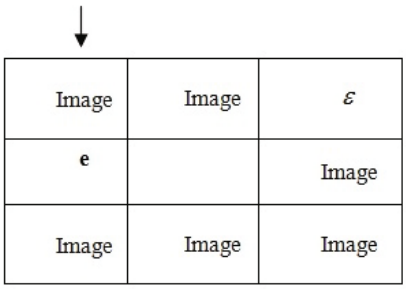


Fig. 12 A coach of a 7-atrain which is neither empty nor full

The Fig. 13 shows two consecutive coaches C_i and C_{i+1} in a 7-atrain, where C_i is a full coach and C_{i+1} is a coach which is neither empty nor full. Here the datatypes of two coaches are different (unlike in train coaches). Fundamental operations in a r-atrain are explained in section 8.3 later here.

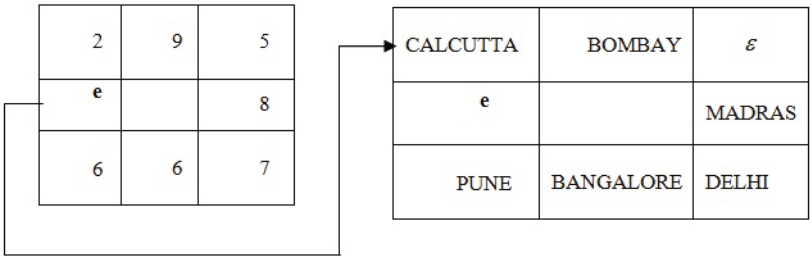


Fig. 13 Two consecutive coaches of different datatypes

2.3.7 A r-Atrain in Memory

Refer to the CD-Table of section 2.3.1. Consider the data: 5.3, 6.2, 9.1, DELHI, CALCUTTA, BOMBAY, #, %, {, 9, 2 which are stored in the Data Segment of the 8086 memory using the data structure 3-atrain, as shown in Table.3 starting from START = 0A02h.

Table 3 A r-Atrain in 8086 memory

Address	Memory Content	Size
	X (an invalid address)	2 bytes
	ε	2 bytes
	2	2 bytes
E49Bh	9	2 bytes
	...	
	00ADh	2 bytes
	BOMBAY	20 bytes
	CALCUTTA	20 bytes
BA98h	DELHI	20 bytes
	...	
	BA98h	2 bytes
	9.1	4 bytes
	6.2	4 bytes
10A5h	5.3	4 bytes
	...	
	1	2 bytes
	1	2 bytes
	C_4 = E49Bh	2 bytes
	0	2 bytes
	0	2 bytes
	C_3 = 00ADh	2 bytes
	0	2 bytes
	3	2 bytes

Table 3 (*continued*)

	C_2 = BA98h	2 bytes
	0	2 bytes
	2	2 bytes
START = 0A02h	C_1 = 10A5h	2 bytes
	...	
	E49Bh	2 bytes
	{	1 byte
	%	1 byte
00ADh	#	1 byte
	...	

This is a 3-atrain T = < [10A5h, (2, 0)], [BA98h, (3, 0)], [00ADh, (0, 0)], [E49Bh, (1, 1)] > which is of length 4 where the coach C_1 begins from the address 10A5h, the coach C_2 begins from the address BA98h, the coach C_3 begins from the address 00ADh, the coach C_4 begins from the address E49Bh. Here START = 0A02h, i.e. the pilot of this 3-atrain is stored at the address 0A02h. Also in this example, the pilot is implemented as an array, not using linked list. However for a large pilot linked list representation may be preferred.

3 Solid Matrix and Solid Latrix (for Big Data and Temporal Big Data)

We know that a matrix is a rectangular array of numbers or other mathematical objects, for which various operations such as addition and multiplication are defined [11]. Most commonly, a matrix over a field F is a rectangular array of scalars from F. But we may also consider a generalized kind of matrix whose elements are objects over the real region RR [5], or over any appropriate region R. For details about the 'Region Algebra' and the 'Theory of Objects' in a region, one could see [5]. In this section a new theory [6] of solid matrices (solid latrices) is discussed.

3.1 Solid Matrix and Solid Latrix

A **solid matrix** is an n-dimensional hyper-matrix where n > 2 and the elements are objects from the real region RR or from any appropriate region R, none being ε elements. We say that it has n number of hyper layers. A solid matrix is a mathematical object and should not be confused with the data structure 'n-dimensional array' in computer science. However, for details about the n-dimensional array (multi-dimensional array) and its MATLAB implementation, one could see any good MATLAB book.

A **latrix** is a rectangular array of numbers (or, objects from the region RR or from any appropriate region R) and ε elements. We define a solid latrix as an

n-dimensional hyper-latrix where n > 2 and the elements are objects from the region RR or from an appropriate region R, and may be ε elements.

We use the notation n-SM to denote an n-dimensional solid matrix and n-SL to denote an n-dimensional solid latrix. An abstract bottom-up approach to view the structure of a n-SM is:

Imagine a classical two dimensional matrix S_2 of size $m_1 \times m_2$. Suppose that the matrix S_2 expands in a new dimension upto a height m_3 to form a 3-SM S_3. Now suppose that the matrix S_3 expands in another new dimension upto a hyper height m_4 to form a 4-SM S_4, and so on. Finally, suppose that the matrix S_{n-1} expands in another new dimension upto a hyper height m_n to form a n-SM S_n. Thus we have an n-SM S_n of size $m_1 \times m_2 \times m_3 \times \ldots \times m_n$ where n ≥ 3.

Height of a Solid Matrix (Solid Latrix) Consider an n-SM (n-SL) S of size $m_1 \times m_2 \times m_3 \times \ldots \times m_n$ where n ≥ 3. The last suffix-index m_n is called the height of the solid matrix (solid latrix) S. We write it as height (S) = m_n. As a trivial case, we assume that the height of a classical matrix/latrix (i.e. viewing a two dimensional matrix/latrix as a 3-SM or 3-SL of size $m_1 \times m_2 \times 1$) is 1.

Base Size, Base Matrix and Base Latrix Consider an n-SM (n-SL) S of size $m_1 \times m_2 \times m_3 \times \ldots \times m_n$ where n ≥ 3. Then the base size of S is $m_1 \times m_2$ which is constituted by the first two indices m_1 and m_2 out of the size of the SM (SL) S. The sub-matrix (sub-latrix) B of the SM (SL) S of size $m_1 \times m_2 \times m_3 \times \ldots \times m_n$ where $m_3 = m_4 = \ldots = m_n = 1$ is called the base matrix (base latrix) of S. As a trivial case, we assume that the base size of a classical matrix/latrix M (i.e. of a two dimensional matrix/latrix) is the size of the matrix/latrix itself, and the base matrix (base latrix) of M is M itself.

There are many real life examples of 3-D solid matrices/latrices in different branches of science and engineering. In the next We explain the structure of 3-D solid matrices/latrices, study various operations on them and various properties of them. The same can be well extended for n-dimensional solid matrices (hyper matrices) or for n-dimensional solid latrices (hyper latrices) in an analogous way.

3.2 3-D Solid Matrix (3-SM) and Some Characterizations

Consider h number of m$\times$n matrices $M_1, M_2, M_3, \ldots, M_h$ where m, n, h are positive integers. Consider an abstract multilayer structure where layer-1 is the bottom most layer which is the matrix M_1, layer-2 is the matrix M_2, layer-3 is the matrix M_3, $\ldots$, layer-h is the top most layer matrix M_h. One could view this structure as a rectangular parallelepiped from distance. This rectangular parallelepiped is called a 3-D Solid Matrix (in short, 3-SM) of size m$\times$n$\times$h. A solid matrix S is thus a 3-dimensional matrix having mnh number of cells. We say that the length (row-size) of the SM S is m, the breadth (column-size) of the SM S is n and the height of the SM S is h. We denote a 3-SM S of size m$\times$n$\times$h by:

$$S = < M_1, M_2, M_3, \ldots, M_h >.$$

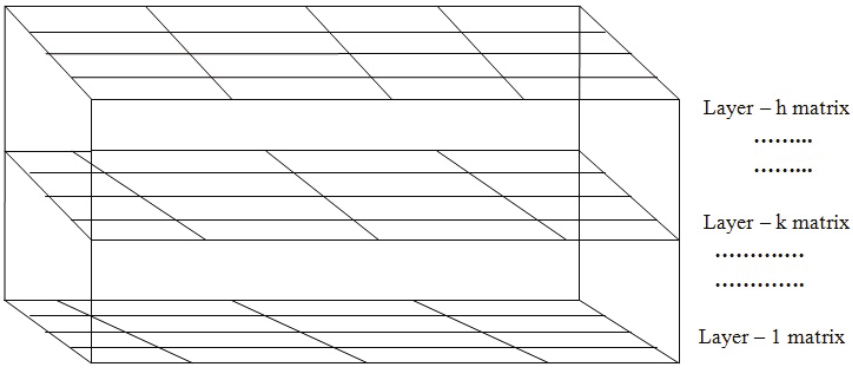


Fig. 14 A semicube of height h

If m = n = h (= a, say), then the SM is called a **Cube** of side a, which is having a^3 number of cells. If m = n, then the SM is called a **Semi-cube**. Each layer of a cube or semi-cube is a square matrix. For an n-SM S of size $m_1 \times m_2 \times m_3 \times \ldots \times m_n$ where n ≥ 3, if $m_1 = m_2 = m_3 = \ldots = m_n$ (= a, say) then the SM is called a n-dimensional **Hyper Cube** of size a.

Example of a Solid Matrix and a Solid Latrix. Consider a class of 3^{rd} semester MS(CS) Programme in a university having 60 students in the class, the names of the students being $X_1, X_2, X_3, \ldots, X_{60}$. There are five number of 'Theory' courses in this semester which are C_1, C_2, C_3, C_4, and C_5 being taught by five lecturers. Every weak one Sessional Examination is to be conducted over 50 marks by every lecturer on the subject he is teaching, as per university bye-laws for 'Continuous Evaluation Systems'. In total 12 number of Sessional Examinations are to be conducted on each course in this semester. Clearly, the result of students in 1^{st} Sessional Examination form the matrix M_1 of size 5×60 as shown below in Fig. 12. In a similar way there will be eleven matrices more, each of size 5×60. These twelve matrices $M_1, M_2, M_3, \ldots, M_{12}$, if logically placed one above another, will form one solid matrix (3-SM) S of size $5 \times 60 \times 12$. If there is a query: "What is the marks obtained by the student X_{29} in the 8^{th} Sessional Examination in course C_3?", the answer to this query will be the data element $s_{3,29,8}$ of S.

A 3-SM is a particular case of a 3-SL. A layer of a 3-SL is shown in Table.4.

Table 4 Layer-1 (bottom layer) latrix L_1 of a 3-SL

42	18	⋯	49
ε	25	⋯	08
49	ε	⋯	ε
02	00	⋯	05
ε	ε	⋯	50

	X_1	X_2	- - - -- - - - - --	X_{60}
C_1	42	18	- - - - - - - --	49
C_2	37	25	- - - - - - - --	08
C_3	49	50	- - - - - - - --	42
C_4	02	00	- - - - - - - --	05
C_5	39	28	- - - - - - - --	50

Fig. 15 Layer-1 (bottom layer) matrix M_1 of a 3-SM

Null Solid Matrix
The 3-SM S = $< O_{m\times n}, O_{m\times n}, O_{m\times n}, \ldots, O_{m\times n} >$ of size m×n×h, where $O_{m\times n}$ is the classical null matrix of size m×n, is called the **Null 3-SM** of size m×n×h.

In a recursive way the above definition can be extended to define a **Null n-SM**: The n-SM $S_o = < O_{m_1\times m_2\times m_3\times \ldots\times m_{n-1}}, O_{m_1\times m_2\times m_3\times \ldots\times m_{n-1}}, \ldots, O_{m_1\times m_2\times m_3\times \ldots\times m_{n-1}} >$ of size $m_1\times m_2\times m_3\times \ldots\times m_n$ where $O_{m_1\times m_2\times m_3\times \ldots\times m_{n-1}}$ is the null (n-1)-SM of size $m_1\times m_2\times m_3\times \ldots\times m_{n-1}$ is called the Null n-SM of size $m_1\times m_2\times m_3\times \ldots\times m_n$.

Unit Semi-cube and Unit Cube
The semi-cube I = $< I_{m\times m}, I_{m\times m}, I_{m\times m}, \ldots, I_{m\times m} >$ of size m×m×h, where $I_{m\times m}$ is the classical unit matrix of order m×m, is called the **Unit Semi-cube** of size m×m×h. The cube I = $< I_{m\times m}, I_{m\times m}, I_{m\times m}, \ldots, I_{m\times m} >$ of size m×m×m is called the **Unit Cube** of size m×m×m. In a recursive way the above definition can be extended to define a **Unit Hyper Semi-Cube** and **Unit Hyper Cube**.

4 Algebra of Solid Matrices (Whose Elements Are Numbers)

In this section an algebraic study of SMs is done. Few basic operations on SMs and their properties where the elements are from the real region RR [5] of real numbers (not any objects [5] from any region R) are explained.

Addition/Subtraction of Two SMs
Two SMs can be added (subtracted) if they are of same size. The resultant SM is also of the same size. Consider two SMs S_1 and S_2, each of size m×n×h, given by $S_1 = < M_1^1, M_2^1, M_3^1, \ldots, M_h^1 >$ and $S_2 = < M_1^2, M_2^2, M_3^2, \ldots, M_h^2 >$.
Then $S_1 + S_2 = < M_1^1 + M_1^2, M_2^1 + M_2^2, \ldots, M_h^1 + M_h^2 >$ and
$S_1 - S_2 = < M_1^1 - M_1^2, M_2^1 - M_2^2, \ldots, M_h^1 - M_h^2 >$.

Transpose of a SM
The transpose of a 3-SM A = $[a_{pqr}]$ of size m×n×h is a 3-SM B = $[b_{uvw}]$ of size n×m×h where $a_{ijk} = b_{jik}$ for every i, j, k. We write B = A^T. In A^T each layer of A is transposed. Obviously, if I be a unit semi-cube or a unit cube then I^T = I.

For an n-SM A of size $m_1 \times m_2 \times m_3 \times \ldots \times m_n$, the transpose of A is defined in a similar way but with respect to a given pair of indices ith and jth, where i≠j and i, j ≤ n.

Consider an n-SM A of size $m_1 \times m_2 \times m_3 \times \ldots \times m_i \times \ldots \times m_j \times \ldots m_n$ and an n-SM B of size $m_1 \times m_2 \times m_3 \times \ldots \times m_j \times \ldots \times m_i \times \ldots m_n$. Then the transpose of the n-SM A is the n-SM B with respect to the pair of indices ith and jth, where i≠j and i, j ≤ n, if $a_{pqr\ldots ij\ldots uv} = b_{pqr\ldots ji\ldots uv}$ for every p, q, r, …, i, j, …, u, v. We write B = A^T.

Proposition 4.1
If A be a n-SM, then $(A^T)^T$ = A.

Scalar-Height of Height z
A solid matrix of size 1×1×z of scalar elements is called a **scalar-height** of height z, where z is a positive integer. For example, the 3-SM H shown in Fig. 16 is a scalar-height of height z denoted by H = $< k_1, k_2, k_3, \ldots, k_z >$ where $k_1, k_2, k_3, \ldots, k_z$ are scalar quantities. The scalar-height of height z having all the elements 0 in it is called a **Null Scalar-height** or **Zero Scalar-height** of height z denoted by O_z = $< 0, 0, 0, \ldots, 0 >$ as shown in Fig. 16.

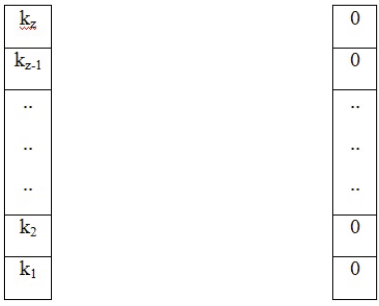


Fig. 16 Scalar-height H and Scalar-height O_z (each of height z)

Scalar-Height Multiplication of a Solid Matrix
Consider a 3-SM S of size m×n×h given by S = $< M_1, M_2, M_3, \ldots, M_h >$ and a scalar-height H = $< k_1, k_2, k_3, \ldots, k_h >$ of height h. The scalar-height multiplication of S by H yields a SM M given by M = $< k_1M_1, k_2M_2, k_3M_3, \ldots, k_hM_h >$ and we write M = HS.

Scalar Multiplication of a Solid Matrix
Consider a scalar quantity b. The scalar multiplication of the 3-SM S of size m×n×h by the scalar b is the scalar-height multiplication of S by the scalar-height H (of height h, the height of the 3-SM S) where H = < b, b, b, …, b > which yields a 3-SM M given by M = $< bM_1, bM_2, bM_3, \ldots, bM_h >$ and we write M = bS.

Determinant-Height of a Semi-cube

Consider a semi-cube SM S of size m×m×h given by S = $<M_1, M_2, M_3, \ldots, M_h>$. The **determinant-height** (det.ht) of the semi-cube SM S is the scalar-height D_S given by $D_S = < |M_1|, |M_2|, \ldots, |M_{h-1}|, |M_h| >$ where $|M_r|$ is the value of the determinant of the square matrix M_r.

If none of the elements of D_S is 0, then the semi-cube S is called to be **Non-Singular** semi-cube. If at least one element in D_S is 0 then it is called to be **Singular**. For a unit semi-cube or a unit-cube S, all the elements in D_S are 1. Also, it is obvious that for a semi-cube or a cube S, det.ht (S) = det.ht (S^T).

Multiplication of Two Solid Matrices

Two 3-SMs A of size m×n×h and B of size r×s×t are compatible for multiplication AB if (i) n = r and (ii) h = t. If A = $< A_1, A_2, A_3, \ldots, A_h >$ and B = $< B_1, B_2, B_3, \ldots, B_h >$ are two 3-SMs of size m×n×h and n×s×h respectively, then AB is a 3-SM of size m×s×h defined by AB = $< A_1B_1, A_2B_2, A_3B_3, \ldots, A_hB_h >$.

Clearly, compatibility of the multiplication AB does not yield the compatibility of the multiplication BA. Even if AB and BA both be compatible, they are in general two distinct 3-SMs of two distinct sizes, i.e. the product of two 3-SMs is not commutative. For a semi-cube S of size m×m×h, the products $S^2, S^3, S^4, \ldots$ etc. are semi-cubes of size m×m×h. Thus, for a cube S of size m×m×m, the products $S^2, S^3, S^4, \ldots$ etc. are cubes of size m×m×m. The SM equation AB = O does not necessarily mean that either A = O or B = O or both.

Inverse of a Non-singular Semi-cube

The **Inverse** of a 3-SM exists if it is a non-singular semi-cube. Consider a non-singular semi-cube S of size m×m×h given by S = $< M_1, M_2, M_3, \ldots, M_h >$. Then inverse of S denoted by S^{-1} is a non-singular semi-cube defined by

$$S^{-1} = < M_1^{-1}, M_2^{-1}, M_3^{-1}, \ldots, M_h^{-1} >.$$

The following propositions are straightforward and can be easily proved.

Proposition 4.2

If S is a non-singular semi-cube of size m×m×h, then

$$SS^{-1} = S^{-1}S = I \text{ (unit semi-cube of size m×m×h).}$$

Proposition 4.3

If A and B are two solid matrices such that AB exists, then $(AB)^T = B^T A^T$.

Proposition 4.4

If A and B are two non-singular semi-cubes such that AB exists, then $(AB)^{-1} = B^{-1}A^{-1}$.

Proposition 4.5

(i) Associativity holds good i.e. If A, B, C are three 3-SMs such that AB and BC exist, then A(BC) = (AB)C.

(ii) Distributive Property hold good i.e.

(a) A (B + C) = AB + AC and
(b) (B + C) A = BA + CA,

where the multiplications are assumed compatible.

5 Homogeneous Data Structure 'MT' for Solid Matrix/Latrix

The solid latrix/matrix [6] is useful if the big data is temporal big data. Otherwise, there is no need to choose for solid latrix/matrix. Because, a latrix/matrix can be scalable upto any extent by increasing the number of rows and/or the number of columns to store big data (Needless to mention that not all big data can be stored in the latrix/matrix model). If there is no consideration of time stamp upon the data, the logical storage structure '2-D latrix' (2-D matrix as a particular case) is sufficient to store homogeneous big data as the number of rows/columns in a 2-D latrix (matrix) can be scalable upto any big extent. Our objective is to propose an appropriate data structure and a compatible distributed system to deal with big data if stored in a 2-D latrix (2-D matrix) of big order. In this section we propose a dynamic homogeneous data structure MT to deal with big data of homogeneous datatype which can be logically stored in a SL/SM. MT is the abbreviation for '**M**ulti **T**rains', as it is an extension of the homogeneous data structure 'Train'. In the homogeneous data structure train, there are logically two layers: the pilot is the upper layer and the coaches are in the lower/inner layer. We extend the notion of Train by incorporating nil or one or more number of intermediate layers between the pilot (upper layer) and linked-coaches (lower layer) to develop a new homogeneous data structure 'MT'. Here MT is the abbreviation for 'Multi Trains'. The intermediate layers are usually Trains, but could be pilots, linked-coaches, or larrays too. Type of the various layers, according to the construction-needs for the problems under study, are decided by the developers on behalf of the organization concerned. Thus train may be regarded as a special case of MT, where there is(are) no intermediate layer(s) between the upper layer and the lower layer. The total number of layers is called the height, and then height(Train) = 2, and height(MT)$\geq$2.

5.1 Implementation of a 3-SM (3-SL)

For implementing a 3-SM (3-SL) of homogeneous data, we use MT of height 3 only. However, in general, for implementation of a higher dimensional n-SM (n-SL) of homogeneous data, we use MT of height n. Consider a 3-SM (3-SL) S of size m$\times$n$\times$h given by S = $< M_1, M_2, M_3, \ldots, M_h >$ of homogeneous data. To implement this 3-SM (3-SL) S in computer memory we need actually one chief Pilot of h number of independent trains (one for each layer of 3-SM or 3-SL), where each train is having its own pilot and contains m number of linked coaches. All the h number of trains are independent, but for every train all its coaches are linked/shunted. Surely, we need to consider a MT M with height = 3, i.e. three layers in total. The meaning

of the term 'layer' used in SM and also here in MT are to be carefully differentiated. The implementation method follows bottom-to-upward approach as shown below:-

Lower Layer L_1 of the MT M
It is the chief pilot P = $< M_1, M_2, M_3, \ldots, M_h >$ of the MT M. The 'START' of the MT M points at the chief pilot P. This chief pilot P is nothing but a larray of h number of elements M_i. The element M_i is the address of the ith layer of the 3-SM, which is the 'START' S_i of the ith Train T_i in the MT M.

Middle Layer L_2 of the MT M
It is the larray of h number of pilots corresponding to h number of independent Trains (i.e. h number of independent r-Trains where r = n for the present case), given by $< T_1, T_2, T_3, \ldots, T_h >$ where each T_i corresponds to m number of linked/shunted coaches given by:

$$T_i = < C_1^i, C_2^i, C_3^i, \ldots, C_m^i >, \quad i = 1, 2, 3, \ldots, h.$$

At the time of implementation, the concerned programmer has to take care of the 'status' of each coach.

Upper Layer L_3 of the MT M
In this layer, corresponding to each i there m number of coaches and consequently there are in total mh number of coaches C_j^i (i = 1, 2, 3, …, h and j = 1, 2, 3, …, m). For a given n-train T_i, each coach C_j^i (for j = 1, 2, 3, …, m) has n number of passengers together with one more (the last one) which is one of the following:

(i) an address to the next coach if $j < m$, or
(ii) address to the first coach of immediate higher layer if $j = m$ and $i < h$ or
(iii) an invalid address X if $j = m$ and $i = h$.

However, if the data are not all homogeneous, the concerned programmer has to go for using the heterogeneous data structure MA instead of MT (mentioned in details in Section 9). Scalability is an open option to the programmer, by increasing the number of middle layers.

Example
For the sake of presentation here we ignore big size 3-SM, but consider a small size 3-SM S = $<M_1, M_2>$ of size $3 \times 7 \times 2$ of homogeneous data, given by

$M_1 =$

2	9	5	8	7	6	6
4	6	3	5	4	1	8
9	4	0	4	6	2	0

and $M_2 =$

5	8	2	4	0	8	1
3	8	1	7	9	3	0
8	6	2	7	1	8	5

Fig. 17 Two layers of a SM S of height 2

For implementing this 3-SM S, we need two 7-Trains: $T_1 = < C_1^1, C_2^1, C_3^1 >$ and $T_2 = < C_1^2, C_2^2, C_3^2 >$, each of three coaches.

It is clear that status of each of these six coaches is 0, as there is no ε element in this 3-SM (For a 3-SL, status of few coaches could be other than 0). Let the 'START' of the 3-SM S is the address 1A12h in 8086 memory. Also suppose that the 7-train T_1 is stored at the address E74Bh and the 7-train T_2 is stored at the address D310h. Then the following will be incorporated in the MT (see Fig. 18):

Lower Layer L_1 of the MT M

The START M will point to the chief Pilot P = $<M_1, M_2>$ = < E74Bh, D310h>. Now, suppose that the address of the coach C_1^1 is 5008h, the address of the coach C_2^1 is A210h, and the address of the coach C_3^1 is 00AFh. Also suppose that the address of the coach C_1^2 is CA76h, the address of the coach C_2^2 is CC80h, and the address of the coach C_3^2 is BEBAh. Then the following layers will be incorporated in the MT:

Middle Layer L_2 of the MT M

It is the larray $< T_1, T_2 >$ of two pilots T_1 and T_2 which are the two 7-Trains given by $T_1 = <$ 5008h, A210h, 00AFh $>$ and $T_2 = <$ CA76h, CC80h, BEBAh $>$.

From the chief pilot, one can visit directly to any of these two 7-Trains. M_1 points at the 7-Train T_1 and M_2 points at the 7-Train T_2.

Upper Layer L_3 of the MT M

In this layer, corresponding to each 7-Train T_1 and T_2, there 3 number of coaches, and consequently there are in total 3.2 = 6 number of coaches (C_1^1, C_2^1, C_3^1) and (C_1^2, C_2^2, C_3^2). Each coach C_j^i has 7 number of passengers, together with one more (the last one) which is an address to the next coach if j < 3, address to the first coach of immediate higher layer if "i = 1 and j = 3", but to an invalid address for "i = 2 and j = 3". The status of each coach in this example is 0. From the 7-Train T_1, one can visit directly any of its coaches C_1^1, C_2^1 and C_3^1; and similarly from the 7-Train T_2 one can visit directly any of its coaches C_1^2, C_2^2 and C_3^2.

The Fig. 18 shows the implementation of the data structure MT and Table.5 shows how the 3-SM S is stored in 8086 Memory in an autonomous computer system (implementation in a distributed system is explained in Section 8 for big data). In this example we consider a SM of data for which the r-trains with r = 7 have been used. During the implementation with the data structure MT, one can use as many trains as required according to the size of big data. But for any MT the top most layer shall always consist of coaches only, not of any train. In this case, for storing every coach the 'GETNODE' will always provide sixteen number of free consecutive bytes from the memory. In each of such nodes, the first fourteen bytes contain the information and the last two bytes contain an address as explained earlier. However, T_1 and T_2 being larrays will require six bytes each. The Table.5 shows how this 3-SM (3-SL) S is stored in 8086 Memory (autonomous system) starting from START = 1A12h.

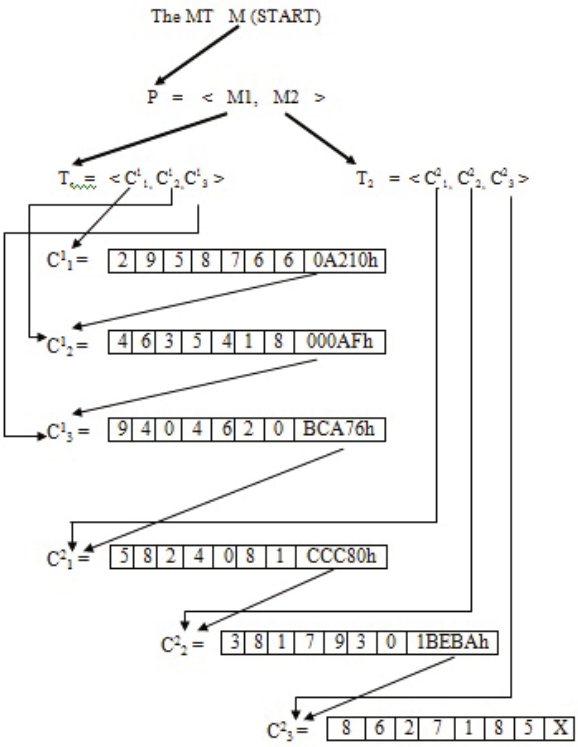


Fig. 18 Implementation of the data structure MT for the 3-SM of height 2

Table 5 A 3-SM (3-SL) in 8086 memory

Address	Memory Content	Size
	...	
	0	2 bytes
	C_3^1 = 000AFh	2 bytes
	0	2 bytes
	C_2^1 = 0A210h	2 bytes
	0	2 bytes
FE74Bh	C_1^1 = 05008h	2 bytes

Table 5 (*continued*)

	...	
	0	2 bytes
	C_3^2 = 1BEBAh	2 bytes
	0	2 bytes
	C_2^2 = CCC80h	2 bytes
	0	2 bytes
CD310h	C_1^2 = BCA76h	2 bytes
	...	
	1BEBAh	2 bytes
	0	2 bytes
	3	2 bytes
	9	2 bytes
	7	2 bytes
	1	2 bytes
	8	2 bytes
CCC80h	3	2 bytes
	...	
	CCC80h	2 bytes
	1	2 bytes
	8	2 bytes
	0	2 bytes
	4	2 bytes
	2	2 bytes
	8	2 bytes
BCA76h	5	2 bytes
	...	
	X (an invalid address)	2 bytes
	5	2 bytes
	8	2 bytes
	1	2 bytes
	7	2 bytes
	2	2 bytes
	6	2 bytes
1BEBAh	8	2 bytes

Table 5 (*continued*)

	...	
	000AFh	2 bytes
	8	2 bytes
	1	2 bytes
	4	2 bytes
	5	2 bytes
	3	2 bytes
	6	2 bytes
0A210h	4	2 bytes
	...	
	T_2 = CD310h	2 bytes
START = 01A12h	T_1 = FE74Bh	2 bytes
	...	
	0A210h	2 bytes
	6	2 bytes
	6	2 bytes
	7	2 bytes
	8	2 bytes
	5	2 bytes
	9	2 bytes
05008h	2	2 bytes
	...	
	BCA76h	2 bytes
	0	2 bytes
	2	2 bytes
	6	2 bytes
	4	2 bytes
	0	2 bytes
	4	2 bytes
000AFh	9	2 bytes
	...	

In many real cases in particular in many engineering problems, statistical and science problems, big data can be viewed as an n-SM (n-SL). Thus, for implementation of an n-SM (n-SL) S of size $m_1 \times m_2 \times m_3 \times \ldots \times m_n$ of homogeneous data, we need to use a MT of height n as below:

Upper Layer L_n of the MT M:
Middle Layer L_{n-1} of the MT M:
Middle Layer L_{n-2} of the MT M:
. . . . . .
. . . . . .
. . . . . .
Middle Layer L_3 of the MT M:
Middle Layer L_2 of the MT M:
Lower Layer L_1 of the MT M:

6 Hematrix and Helatrix: Storage Model for Heterogeneous Big Data

A **Hematrix (HM)** is a rectangular logical array of objects of heterogeneous data types where the data are heterogeneous in different rows, but identical inside every row. Every row itself contains homogeneous data, but the property of heterogeneity incorporated in the different rows. Thus a hematrix H may have objects like: image, DOC file, PDF file, integer number, string of characters, etc. in different rows, but same type of objects inside a row. The term 'Hematrix' stands for 'Heterogeneous Matrix'. In the logical structure HM, for any given row all the cells will require equal amount of space in memory for storing their respective contents, but cells of different rows will require different amount of space in memory.

A **Helatrix (HL)** is similar to a hematrix, but it may contain ε elements in its cells. The term 'Helatrix' stands for 'Heterogeneous Latrix'. As a trivial case every hematrix is a helatrix but the converse is not true. An Example of a Helatrix H_1 of order 5×7 is shown in Table 6 as a bottom layer of a 3-SHL:

Table 6 Layer-1 (bottom layer) helatrix H_1 of a 3-SHL

h_{11}	h_{12}	...	h_{17}
h_{21}	h_{22}	...	h_{27}
h_{31}	h_{32}	...	h_{37}
h_{41}	h_{42}	...	h_{47}
h_{51}	h_{52}	...	h_{57}

In the helatrix H_1 above, h_{1i} are files or ε elements each of size close to (but less than) 1 MB, h_{2i} are integers or ε elements, h_{3i} are files or ε elements each of size close to (but less than) 25 MB, h_{4i} are strings or ε elements each of maximum 50 characters, h_{5i} are files or ε elements each of size close to (but less than) 10 MB, where i = 1, 2, 3..., 7. It may be recollected that the ε elements introduced in

subsection 2.1 in such a way that they do not have any absolutely fixed datatype but they occupy space according to the datatype of the concerned larrays.

A **solid hematrix (SHM)** is an n-dimensional hyper-hematrix where $n > 2$ and the elements are objects of heterogeneous data types, none being ε elements. We say that it has n number of hyper layers. A **solid helatrix (SHL)** is an n-dimensional logical hyper-helatrix where $n > 2$.

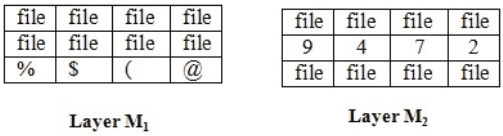


Fig. 19 Two layers of a SHM

The mathematical models HM, HL, SHM, SHL can easily support big data of heterogeneous datatype because of the fact that they are scalable upto any extent by extending the number of rows or columns or height, according to requirements. The Solid Hematrix/Helatrix is useful if the big data is a temporal big data. The logical storage structure '2-D Helatrix' (2-D Hematrix as a particular case) is sufficient to store heterogeneous big data as the number of rows/columns in a 2-D helatrix can be scalable upto any big extent. Our objective is to propose an appropriate data structure to deal with big data if stored in a 2-D helatrix (2-D hematrix) of big order. Consequently, instead of MT or MA which are useful for temporal big data, the heterogeneous data structure r-atrain (r-train for homogeneous big data) will be the appropriate data structure in a distributed system of appropriate architecture.

7 Atrain Distributed System (ADS) for Big Data

In this section we introduce a new distributed system for big data universe called by 'Atrain Distributed System' (ADS) with new type of network topologies called by 'Multi-horse Cart Topology' and 'Cycle Topology'.

7.1 Atrain Distributed System (ADS)

A distributed system consists of a collection of autonomous computers (may be at different locations) connected through a network and distribution middleware which enables all the computers to work by sharing the resources of the system, so that users perceive the system as a single, integrated computing facility.

For the purpose of storing big data, a new type of distributed system is explained here in which there is a **Pilot Computer** (**PC**, in short) connected to m number of computers called by **Distributed Computers** (**DC**, in short) which are DC-1, DC-2, DC-3, . . . , DC-m. Additional connection of computers is allowed only with the distributed computers either in breadth (i.e. horizontally) at the end so as to

be identified by DC-(m+1), DC-(m+2) …etc. or in depth (i.e. vertically) from one or more DCs. Additional connection of computers is not allowed horizontally with the Pilot Computer PC which is one and only one with unique identity. The programmers work on big data with the Pilot Computer (PC) only, not with any DC. From PC to every DC of the next layer, there is a connectivity. But from DC-i to DC-j where j = i+1 and j ≠ m, all such connections are either unidirectional (forward) or bidirectional (forward and backward both). Besides that, the developer (of the concerned organization) may choose for a connection from DC-m to DC-1 to make it circular with unidirectional, or both DC-m to DC-1 with vice-versa to make it circular with bidirectional. For non-circular system, the last DC may be connected with an invalid address if it is unidirectional or in addition to that the first DC may have an invalid backward address for bidirectional system. Such type of distributed system is called an **'Atrain Distributed System' (ADS)**. The name of this type of distributed system is called so because of the fact that it can support the powerful heterogeneous data structure r-atrain [11, 12] to process big data with any challenge from 4Vs.

7.2 'Multi-horse Cart Topology' and 'Cycle Topology' for ADS

An atrain distributed system (ADS) may look apparently to be in the network topology tree where the Pilot Computer is the root node (parent node), and distributed computers are the children nodes sequentially (horizontally) connected. But, as per definition of various types of network topologies [1], it is neither a tree topology nor a bus/ring/star/mesh/hybrid topology. If the last DC be connected with an invalid address (not to the DC-1), then the topology of the network is called by **'Multi-horse Cart'** Topology, as shown in Fig. 20 and Fig. 21. But if the last DC be connected to the DC-1 making it circular, then the topology of the network is called by **'Cycle topology'**, because it looks like a ring (wheel) of a riding pedal cycle connected to the centre PC by spokes, as shown in Fig. 22 and Fig. 23 (however, if the circular connection from DC-m to DC-1 is not incorporated, it will not be a cycle topology). But whatever be the topology, a DC can communicate with any other DC either directly or via other DCs. This type of Atrain Distributed System (ADS) is in fact an **uni-tier ADS** as shown in Fig. 24. However, Fig. 25 shows a distributed system which is not an ADS, because the corresponding topology is neither a multi-horse cart topology nor a cycle topology.

A **multi-tier ADS** can be defined recursively, where every DC can also act as a PC having its own DCs by branching. Thus a multi-tier is a tree having at least one subtree which is too a multi-tier (or at least an uni-tier). The Fig. 22 and Fig. 23 below show multi-tier ADS of 2-tier and 3-tier respectively in multi-horse cart topology of network. The distributed system ADS is designed for the data structures r-atrain and r-train for processing big data, whereas the 'multi-tier ADS' is designed for the data structures MA and MT (not for r-atrain or r-train) for processing too big data. However, all the data structures for big data like r-atrain, r-train, MA and MT can

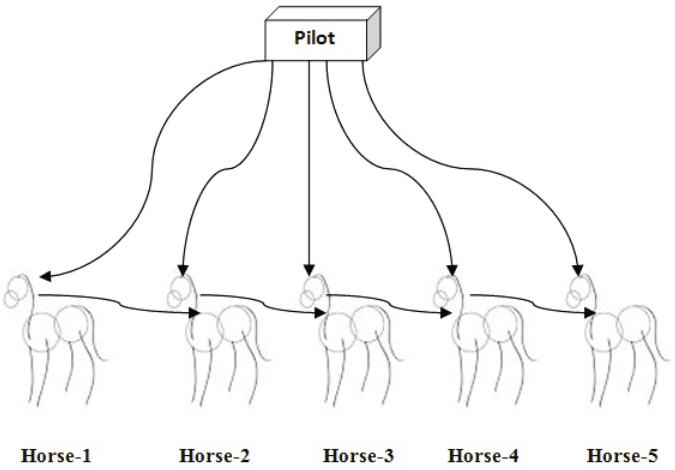


Fig. 20 A Multi-horse Cart Topology of Networks for big data (with twin address **e**)

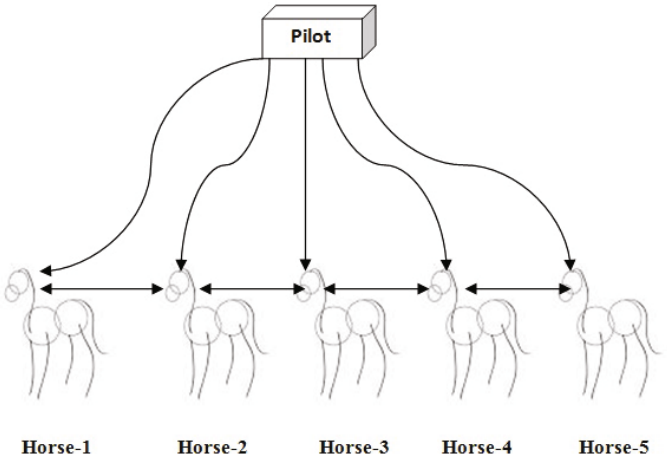


Fig. 21 A Multi-horse Cart Topology of Networks for big data (with doubly twin address **e**)

be well useful in an autonomous computer too, even for small/large data depending upon the code of the datatype.

While implementing ADS, if the last DC is connected to an invalid address then the ADS is in multi-horse cart topology, otherwise if it is connected to the first DC of its siblings then the ADS is in cycle topology. The link address **e** could be twin address or doubly twin address for both the topologies. An ADS is scalable upto any desired extent both in breadth and depth. This is one of the rich merits of ADS to deal with any amount of 4Vs of big data.

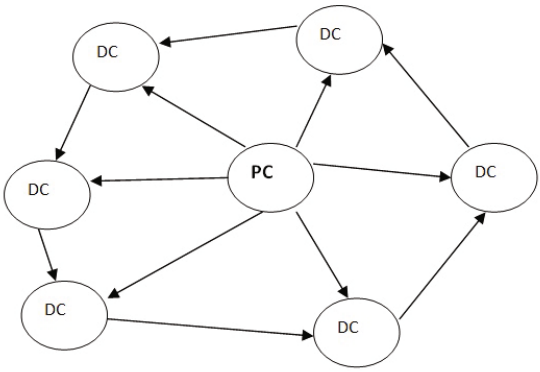


Fig. 22 A Cycle Topology of Networks for big data (with twin address **e**)

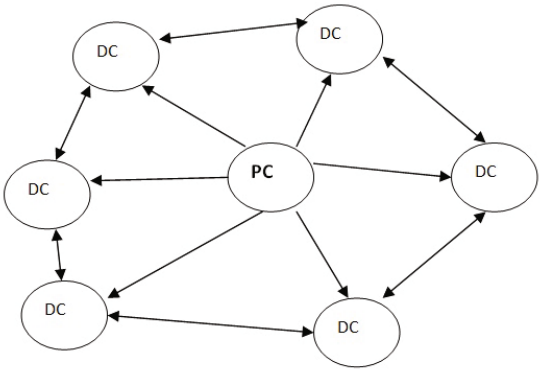


Fig. 23 A Cycle Topology of Networks for big data (with twin address **e**)

8 The Heterogeneous Data Structure 'r-Atrain' in an Atrain Distributed System (ADS)

For implementation of the big data structures train/atrain in an autonomous processor system (not in a distributed system), the notion of CD-Table, Coach, Status of a coach etc. are discussed earlier in section 2 and subsection 2.3. In this section a method is presented on implementation of helatrix (hematrix) in an Atrain Distributed System (ADS) using the heterogeneous data structure 'r-Atrain'. For an atrain distributed system, the notion of CD-Table, Coach, Status of a coach, etc. are almost same as those for an autonomous processor system but with slight adjustment which are presented below.

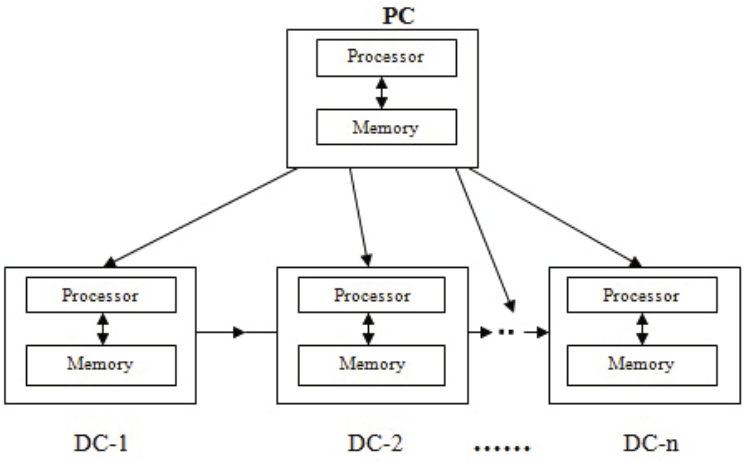


Fig. 24 A unitier ADS in Multi-horse Cart Topology

8.1 Coach of a r-Atrain in an ADS

By a coach C of a r-atrain we mean a pair (A, **e**) where A is a non-empty larray (could be a null larray) and **e** is the 'Twin Address' or 'Doubly Twin Address'.

8.1.1 'Twin Address' e and 'Doubly Twin Address' e

In ADS the atrain coaches will have the link address **twin address e** which is either a 'Twin Address' or a 'Doubly Twin Address'. The **twin address e** is stored in an address node having two fields: Computer Address Field and Memory Address Field, as shown in Table 7 and Table 8.

The Computer Address Field contains the address of a Computer (immediate next one) and the Memory Address Field contains the address of a memory element inside that computer in the atrain distributed system.

Table 7 A Node for Twin Address **e** in an ADS

Computer Address Field	Memory Address Field

Table 8 Twin Address **e** in an ADS

S_i	s

Here the twin address **e** is basically a kind of link address. Its significance is: it says location and the address of the immediate next coach, and thus it links two coaches in the r-atrain.

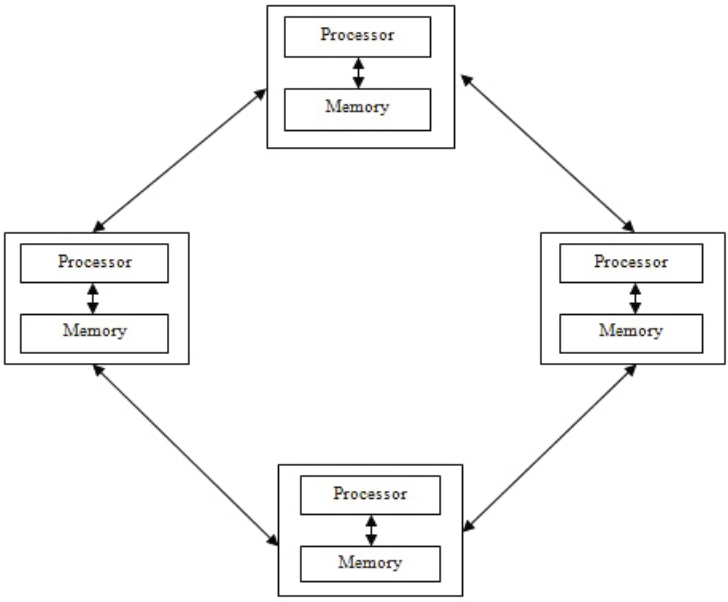


Fig. 25 A distributed system which is not an ADS

However, a **doubly twin address e** sometimes plays a better role to the developers (although not always). The doubly twin address **e** is stored in an address node having four fields: Successor Computer Address Field, Successor Memory Address Field, Predecessor Computer Address Field, Predecessor Memory Address Field, as shown in Table 9 and Table 10.

Table 9 A Node for Twin Address **e** in an ADS

Successor Computer Address Field	Successor Memory Address Field	Predecessor Computer Address Field	Predecessor Memory Address Field

In the 'Doubly Twin Address' **e** in the DC-i, the Successor Computer Address Field contains the address of the immediate next Computer DC-(i+1) and the Successor Memory Address Field contains the address of a memory element inside that computer, whereas the Predecessor Computer Address Field contains the address of the previous Computer DC-(i-1) and the Predecessor Memory Address Field contains the address of a memory element inside that computer, in an atrain distributed system.

Here the twin address **e** is basically a kind of link address. Its significance is: it says location and the address of the immediate next coach (first coach of the immediate next computer), and also it says location and the address of the previous coach (first coach of the previous computer). However, it is the choice

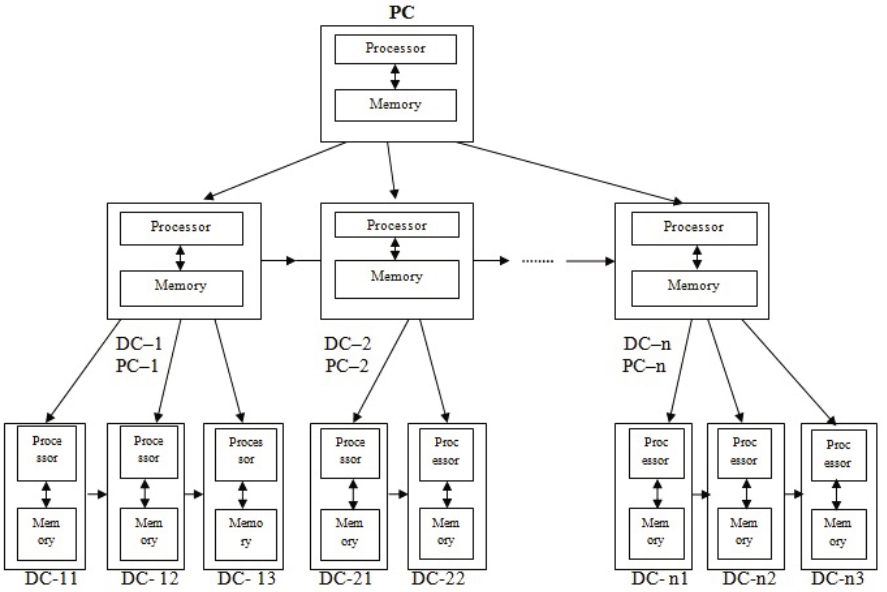


Fig. 26 A multi-tier ADS (2-tier) in a Multi-horse Cart Topology

Table 10 A Node for Twin Address **e** in an ADS

S_{i+1}	s_{i+1}	S_{i-1}	s_{i-1}

of the developer whether to use twin address system or doubly twin address system for the link address **e**. Since each coach can store big data, the excess amount of space required by doubly twin address system (compared to twin address system) is negligible. The doubly twin address system in an ADS allows horizontal movement fluently both forward and backward. However it is advisable that if twin address system in ADS suffices the purpose then doubly twin address system is to be avoided by the developers.

For a cycle topology, the fields corresponding to the Predecessor Computer Address and the Predecessor Memory Address of the link address **e** of DC-1 are filled up with the addresses from the last DC of the siblings (although in a cycle topology there is no significance of first or second DC, etc. as all the DCs are in a cycle, nevertheless we assume that the DCs are identified as 1^{st}, 2^{nd}, 3^{rd}, etc. in the siblings). However, for a multi-horse topology, the fields corresponding to the Predecessor Computer Address and the Predecessor Memory Address of the DC-1 are filled up with invalid addresses, like the Successor Computer Address field and the Successor Memory Address field of the link address ε of the last DC in their siblings.

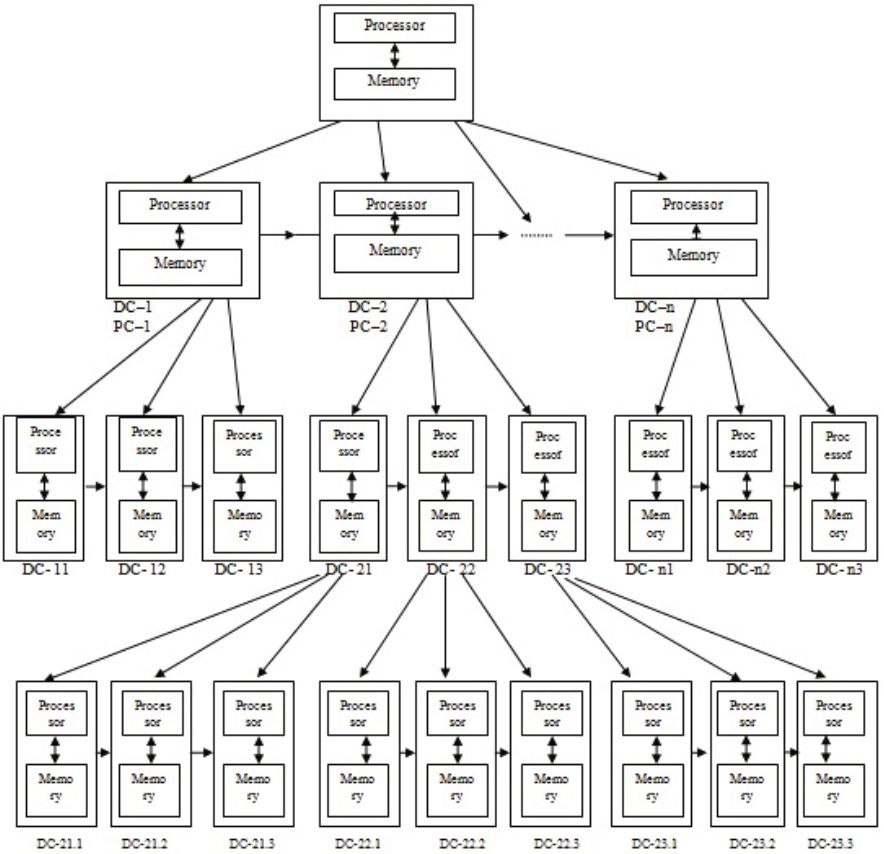


Fig. 27 A multi-tier ADS (3-tier) in a Multi-horse Cart Topology

In a atrain distributed system dealing with big data, different coaches are stored in different computers. It is desired so because the data inside a coach is homogeneous but coach to coach is heterogeneous, and hence the GETNODE will be different for different distributed computers but in accordance with the CD-Table only. If desired by the developer of the concerned organization, one distributed computer may store files of approximately 10 MB size, another distributed computer may store files of approximately 100 MB size, another distributed computer may store integers, and so on. But it is not a strict policy, sometimes two or more coaches may be stored in one computer too if desired by the developer (although not recommended), and in that case GETNODE module needs to be designed accordingly for that distributed computer. These type of decisions is taken by the developer of the concerned organization. If the coach C is the only coach of the r-atrain then the fields of the twin address ε will be put equal to invalid addresses and if the coach C is the last coach of the r-atrain then the fields of the twin address ε will be put either equal to invalid addresses or linked to the first coach (in case circular link is desired).

Otherwise ε will be the twin address of the next coach. The address S_i is the address of a computer and the address s is the address in memory of the computer S_i. Thus the next coach is stored in the computer S_i at the memory address s.

8.1.2 Status of a Coach and Tagged Coach (TC) in a r-Atrain in an ADS

The terms 'status of a coach' and 'tagged coach' (TC) in a r-atrain data structure are discussed in subsection 2.3.3 for a uniprocessor system. The status s of a coach in a r-atrain is a pair of information (c, n), where c is a non-negative integer variable which is the code of datatype (with reference to the concerned CD-Table) of the data to be stored in this coach and n is a non-negative integer variable which is equal to the number of ε elements present in it (i.e. in its larray) at this point of time. If C = (A, **e**) is a coach where **e** is its link address (twin address or doubly twin address, depending upon the choice of the programmer in his ADS), the corresponding tagged coach (TC) will be [(A, **e**), (c, n)].

The most important characteristic of the data structure r-atrain in an atrain distributed system is that it can handle the 4V issue of big data without any problem by making it scalable both horizontally (in breadth) and vertically (in depth) i.e. by adding more number of distributed computers (DCs) to the ADS as per estimated requirements.

Example of a r-Atrain in ADS

Earlier in subsection 2.3.4 few examples of r-atrains are given for a uniprocessor system. In this subsection we present examples in ADS.

Refer to the CD-Table of Table.2. Consider a 3-atrain T of length 3 given by

$$T = < [C_1, (8, 0)], [C_2, (3, 1)], [C_3, (9, 1)] >$$

where $C_1 = <$ F11, F12, F13, $\mathbf{e_1} >$, $C_2 = <$ CALCUTTA, ε, DELHI, $\mathbf{e_2} >$, and $C_3 = <$ F31, F32, ε, an invalid twin address $>$.

Consider the multi-horse cart topology of ADS. Here $\mathbf{e_1}$ is the twin address of the coach C_2 (i.e. address of larray A_2) in this 3-atrain, and $\mathbf{e_2}$ is the twin address of the coach C_3 (i.e. address of larray A_3). Since it is a 3-atrain, each coach C_i can accommodate exactly three passengers (including ε elements, if any). In coach C_1, the status is (8, 0) which means that this coach can accommodate file of size 10 MB or less (with reference to the CD-Table in Table.2 in Section 2.3.1) and there is no free space in this coach at this point of time. The larray is $A_1 = <$F11, F12, F13$>$, which means that according to the CD-Table the first passenger is the file F11, second passenger is the file F12, and the last/third passenger is the file F13; all these three files are of size 10 MB or less, and the data $\mathbf{e_1}$ being the twin address of the next coach C_2. Thus, T is a larray of three TCs which are $[C_1, (8, 0)]$, $[C_2, (3, 1)]$, $[C_3, (9, 1)]$. The logical diagram of this 3-atrain T is shown below where data in coaches are to be read clockwise starting from e_{11} for the coach C_1 and from e_{21} for the coach C_2, from e_{31} for the coach C_3 (as shown in Fig. 28).

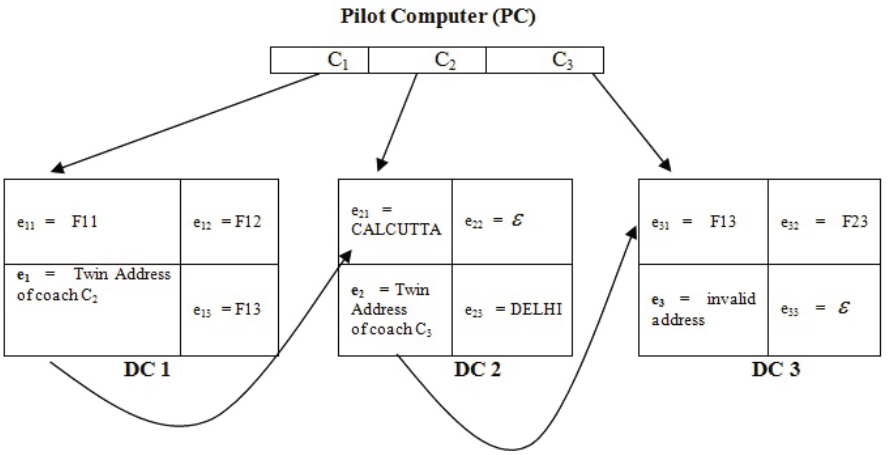


Fig. 28 A 3-atrain with 3 coaches in an ADS

Fig. 29 shows a r-atrain with 30 number of coaches in 30 DCs:-

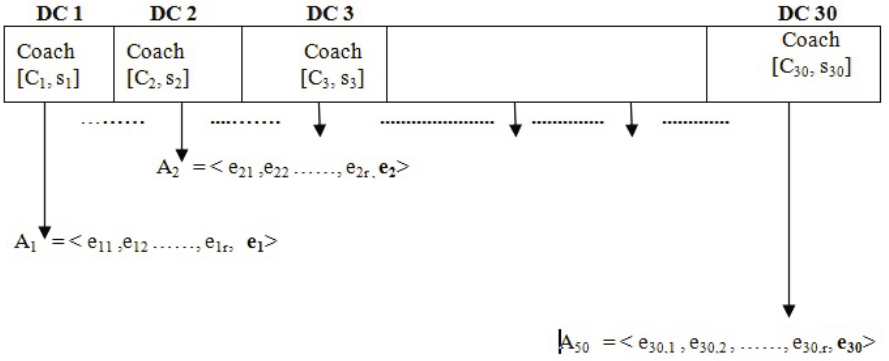


Fig. 29 A r-atrain with 30 coaches in 30 DCs in an ADS

8.2 Circular Train and Circular Atrain

In section 2 the homogeneous data structure r-train (train) and the heterogeneous data structure r-atrain (atrain) for big data are discussed in details in an autonomous processor system. Every coach in a train/atrain is linked to the immediate next coach by the link address **e**. Although all the coaches can be reached directly from the Pilot but there is a forward link from each coach to its immediate next coach. However, the last coach is connected to an invalid address in general. A train (atrain) is called to be a **circular train (circular atrain)** if the last coach is connected to the first coach.

Train/Atrain with Doubly Linked Address e
Here the node of the link address **e** consists of two fields called by "Predecessor" and "Successor". The Predecessor field contains the address of the previous coach and the Successor field contains the address of the next coach. In a cycle topology, the Predecessor field of the first coach is filled up with the address of the last coach (although in a cycle topology there is no significance of first or second coach, etc. as all the coaches are in a cycle, nevertheless we assume that the coaches are identified as 1^{st}, 2^{nd}, 3^{rd}, etc.). However, for a multi-horse cart topology, the Predecessor field of the first coach is filled up with an invalid address, like the successor field of the link address **e** of the last coach.

Fig. 30 shows doubly linked coaches in a r-atrain in ADS:-

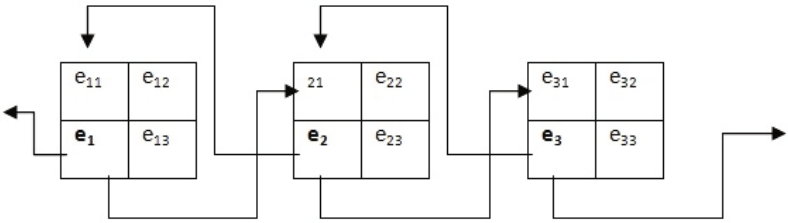


Fig. 30 Doubly linked coaches in a r-atrain

In a doubly linked train/atrain in an ADS, both forward and backward movements are possible from any coach. However, for multi-horse cart topology backward movement is not possible from the first coach and forward movement is not possible from the last coach.

8.3 Fundamental Operations on 'r-Atrain' in an ADS for Big Data

The three fundamental operations on the heterogeneous data structure r-atrain (atrain) in an atrain distributed system are 'insertion', 'deletion', and 'search', which are explained below assuming that the ADS is in multi-horse network topology and the link address e is in twin address system.

(However, if the link address e is in double twin address system then the definition of these three basic operations can be adjusted appropriately. Even, if the ADS in Cycle topology instead of in multi-horse topology, the implementation method needs slight adjustment as the last address here does not link to invalid address but to the address of the first DC of its siblings DCs).

8.3.1 Insertion

There are three types of insertion operations in the heterogeneous data structure r-atrain in an ADS:

(i) insertion (addition) of a new coach in a r-atrain.
(ii) insertion of a data element (passenger) in a given coach of a r-atrain.
(iii) insertion of a data element (passenger) in a r-atrain.

(i) Insertion of a New Coach in a r-Atrain

The first job is to decide about the datatype of the coach which is now required to be inserted.

Consider the r-atrain T = $< (C_1, s_{C_1}), (C_2, s_{C_2}), (C_3, s_{C_3}), \ldots, (C_k, s_{C_k}) >$ with k number of coaches, where the coach $C_i = (A_i, e_i)$ for i = 1, 2, 3, . . . , k. After insertion of a new additional coach, the updated r-atrain immediately becomes the following r-atrain:

$$T = < (C_1, s_{C_1}), (C_2, s_{C_2}), (C_3, s_{C_3}), \ldots, (C_k, s_{C_k}), (C_{k+1}, r) >.$$

Initially at the time of insertion, we create C_{k+1} as an empty coach with status = (c, r) where c is the code of the intended datatype of the coach and since the new coach is empty therefore the number of available space is r at this time, but the coach is likely to get filled-up with non-ε passengers (data) later on with time.

For insertion of a new coach C_{k+1} in a r-atrain, we need to do the following steps:

(i) read the CD-Table for the code c of the datatype of the new coach intended for insertion. If the code c is not available in the CD-Table, expand CD-Table accordingly.
(ii) update the pilot (linked list).
(iii) $\mathbf{e_k}$ in C_k is to be updated and to be made equal to the address C_{k+1}.
(iv) set $e_{k+1,j} = \varepsilon$ for j = 1, 2, . . . , r
(v) set $\mathbf{e_{k+1}}$ = an invalid address.
(vi) set $s_{C_{k+1}}$ = (c, r).

(ii) Insertion of an Element x inside the Coach $C_i = (A_i, e_i)$ of a r-Atrain

Insertion of an element (a new passenger) x inside the coach C_i is feasible if x is of same datatype (like other passengers of the coach C_i) and if there is an empty space available inside the coach C_i. If the availability-status n of C_i is greater than 0 then data can be stored successfully in this coach, otherwise insertion operation fails here at this moment of time. For insertion of x, we can replace the lowest indexed passenger ε of C_i with x.

After each successful insertion, the availability-status n of the coach is to be updated by doing n = n-1, and thus by updating the status s_{C_i} = (c, n) by its new value given by s_{C_i} = (c, n-1).

(iii) Insertion of an Element x in a r-Atrain

In this case too, the code c (with reference to the concerned CD-Table) corresponding to the datatype of the data x has to be given the first attention in the process of insertion. An initial search is done for the coaches (starting from C_1 onwards) which are having the same code c in their status. Suppose that, the array of the code-matched coaches so extracted from the pilot is Z = $(Z_1, Z_2, Z_3, \ldots, Z_t)$.

If the array Z is a null array or if the availability-status is zero for each and every member of Z, then the insertion operation is to be done by inserting a new coach

C_μ first of all, as per steps mentioned above and then by performing the insertion operation. Otherwise we find out the coach Z_k in Z with lowest index k for which the availability-status n is greater than 0, and then perform the insertion operation.

8.3.2 Deletion

There are two types of deletion operation in the data structure r-atrain:

(i) Deletion of a data element ($\neq \varepsilon$) from any coach of the r-atrain.
(ii) Deletion of the last coach Ci, if it is an empty coach, from a r-atrain.

(i) Deletion of a Data e_{ij} ($\neq \varepsilon$) from the Coach C_i of a r-Atrain
Deletion of e_i from the coach C_i is not allowed as it is the link. But we can delete a data element e_{ij} from the coach C_i. Deletion of a data (passenger) from a coach means replacement of the data by an ε element (of same datatype). Consequently, if $e_{ij} = \varepsilon$, then the question of deletion does not arise. Here it is pre-assumed that e_{ij} is a non-ε member element of the coach C_i.

For j = 1, 2, …, r, deletion of e_{ij} is done by replacing it by the null element ε, and updating the availability-status n by doing n = n+1. Deletion of a data element (passenger) does not effect the size r of the coach.

For example, consider the tagged coach $[C_i, (c_i, \text{m})]$ where

$$C_i = <e_{i1}, e_{i2}, e_{i3}, \ldots, e_{ir}>.$$

If we delete e_{i3} from the coach C_i, then the updated tagged coach will be $[C_i, (c_i, \text{m+1})]$ where $C_i = <e_{i1}, e_{i2}, \varepsilon, e_{i4}, \ldots, e_{ir}>$.

(ii) Deletion of the Last Coach C_i from a r-Atrain
Deletion of coaches from a r-atrain is allowed from the last coach only and in backward direction, one after another. An interim coach can not be deleted. The last coach C_i can be deleted if it is an empty coach (as shown in Fig. 11): -

If the last coach is not empty, it can not be deleted unless its all the passengers are deleted to make it empty. To delete the empty last coach C_i, of a r-atrain, we have to do the following actions:

(i) update e_{i-1} of the coach C_{i-1} by storing an invalid address in it.
(ii) delete $[C_i, (c_i, \text{r})]$ from the r-atrain
$$\text{T} = < [C_1, s_{C_1}], [C_2, s_{C_2}], [C_3, s_{C_3}], \ldots, [C_{i-1}, s_{C_{i-1}}], [C_i, (c_i, \text{r})] >$$
and get the updated r-atrain
$$\text{T} = < [C_1, s_{C_1}], [C_2, s_{C_2}], [C_3, s_{C_3}] , \ldots, [C_{i-1}, s_{C_{i-1}}] > .$$
(iii) update the pilot.

Note: *Although insertion of any coach anywhere in-between and deletion of any interim coach from anywhere can be well coded and implemented by the developers in the data structures train and atrain (like the operations insertion/deletion of a node in/from a linked-list), nevertheless it is not advisable as these two data structures are designed exclusively for big data.*

8.3.3 Searching for a Data x in a r-Atrain T of Length k

Searching for a data x in a r-atrain T is very easy. If we know in advance the coach number C_i of the passenger x, then by visiting the pilot we can enter into the coach Ci of the r-atrain directly and then can read the data-elements $e_{i1}, e_{i2}, e_{i3}, \ldots, e_{i(r-1)}$, e_{ir} of the larray A_i for a match with x. Otherwise, the code c (from the CD-Table) of the datatype of the data x plays an important role in the process of searching. The initial search is done for the coaches (starting from C_1 onwards) which are having the same code c in their status. Suppose that the array of the code-matched coaches so extracted is $Z = (Z_1, Z_2, Z_3, \ldots, Z_t)$. If the array Z is a null array, the search fails. Otherwise we start searching inside, beginning from the coach Z_1 onwards till the last coach Z_t of Z. The search may lead to either success or failure.

We need not go back to the pilot for any help during the tenure of our searching process. Here lies an important dominance of the data structure r-atrain over the data structure HAT introduced by Sitarski [16]. In case of multi-processor system the searching can be done in parallel very fast, which is obvious from the architecture of the data structure r-atrain.

9 Heterogeneous Data Structures 'MA' for Solid Helatrix of Big Data

Today's supercomputers or multiprocessor systems which can provide huge parallelism has become the dominant computing platforms (through the proliferation of multi-core processors), and the time has come to stand for highly flexible advanced level of data structures that can be accessed by multiple threads which may actually access any big volume of heterogeneous data simultaneously, or even that can run on different processors simultaneously. In most of the giant business organizations, the system has to deal with a large volume of heterogeneous data or heterogeneous big data for which the data structures of the existing literature can not lead to the desired solution for thirst or desired optimal satisfaction. The very common and frequent operations like Insertion, deletion, searching, etc. are required to be faster for the big data. Such situations require some way or some method which work more efficiently than the simple rudimentary existing data structures. Obviously, there is a need of a dash of creativity of a new or better performed heterogeneous data structure for big data, new mathematical models for big data, new distributed systems for big data, which at the same time must be of rudimentary in nature.

In this section we propose a very powerful and dynamic real time heterogeneous data structure MA to deal with big data of heterogeneous datatype, and then we present a generalized type of application of MA. MA is the abbreviation for '**M**ulti **A**trains', as it is an extension of the heterogeneous data structure 'Atrain' (**A**dvanced **train**). In the heterogeneous data structure Atrain, there are logically two layers: the pilot is the lower layer and the coaches are in the upper/inner layer. We extend the notion of Atrain by incorporating nil or one or more number of intermediate layers between the pilot (lower layer) and linked-coaches (upper

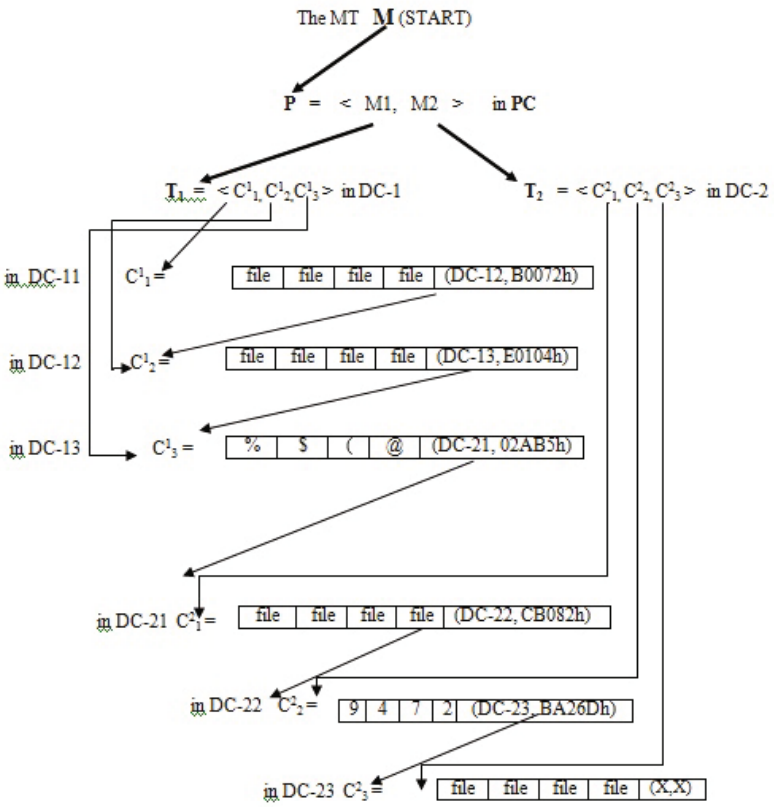


Fig. 31 Implementation of a 3-SHM S of height 2 using the data structure MA in ADS

layer) to develop a new heterogeneous data structure 'Multi Atrains (MA)'. The intermediate layers are usually Atrains, but could be pilots, linked-coaches, or larrays too. Type of the various layers, according to the construction-needs for the problems under study, are decided by the developers on behalf of the organization concerned. Thus Atrain may be regarded as a special case of MA, where there is(are) no intermediate layer(s) between the upper layer and the lower layer. If the total number of layers is called the 'height', then height(Atrain) = 2, and height(MA) $\geq$ 2. Clearly, an 'MT of height h' is a particular case of an 'MA of height h'. The Solid Hematrix/Helatrix is very useful if the big data is a temporal big data (although in Bangalore/Chennai/Hyderabad many of the the IT experts say that trivially any big data is a kind of temporal big data). Otherwise, the logical storage structure '2-D Helatrix' (2-D Hematrix) is the appropriate model to store heterogeneous big data as the number of rows/columns in a 2-D helatrix can be scalable upto any big extent having significant consistence with ADS. Consequently, the data structures MT or MA are useful for temporal big data only. Implementation method of a 3-SHL(3-SHM) using MA in an autonomous system is similar to the

implementation of a 3-SL(3-SM) using MT in an autonomous system as explained earlier in section 5.

In Fig. 31 each of M_1 and M_2 are twin addresses given by M_1 = (DC-1, T_1) and M_2 = (DC-2, T_2) respectively. The elements C_1^1, C_2^1 and C_3^1 in T_1 are the twin addresses (DC-11, A21B0h), (DC-12, B0072h) and (DC-13, E0104h) respectively. Similarly the elements C_1^2, C_2^2 and C_3^2 in T_2 are the twin addresses (DC-21, 02AB5h), (DC-22, CB082h) and (DC-23, BA26Dh) respectively.

10 Conclusion

The homogeneous data structure r-train and the heterogeneous data structure r-atrain are two appropriate data structures developed for big data. A r-train is a trivial case of a r-atrain. The atrain (train) should not be confused with a thought that it is just a linked list of arrays, but it is much more. For storing big data we must have big amount of big spaces in our availability for which we need big amount of memory spaces to work in a fully integrated environment of software, hardware, and required peripherals. Consequently, there is a need of an appropriate and efficient new model for distributed system. The new type of distributed system introduced in this chapter for big data is called by 'Atrain Distributed System' (ADS) which could be unitier or multitier. An ADS will have a single unique Pilot Computer (PC) and several Distributed Computers (DCs) connected by new type of network topologies called by 'Multi-horse Cart Topology' and 'Cycle Topology'. These topologies are designed to support the ADS for big data compatible with the data structures train and atrain exclusively for big data. A 'Multi-horse Cart Topology' or a 'Cycle Topology' is neither a tree topology nor a bus/ring/star/mesh/hub/hybrid topology of existing known. A cycle topology looks like a ring (wheel) of a riding pedal cycle connected to the centre PC by spokes. Both the new type of topologies are implemented in an ADS either by twin address **e** or by doubly twin address **e**, choice being of the developer of the concerned organization/institution. The doubly twin address facilitates the sibling DCs to communicate with their respective next and respective previous DCs (i.e. both forward and backward communication possible). For twin address, Cycle can revolve either clockwise only (or, anti-clockwise only). For doubly twin address, Cycle can revolve both in clockwise and anti-clockwise directions. ADS is exclusively designed for processing big data, in particular to challenge 4Vs in a controlled manner. Driver can send signal to any of the horses (DCs) from his own seat (PC). Scalability upto any desired amount in ADS is a simple process and can be implemented with minor adjustment at the PC and the identified DCs, without disturbing the other DCs It is to be noted that the programmers/users will be always in the PC which is unique). One of the rich merits of ADS is that the scalability can be implemented both in breadth (horizontally) and depth (vertically) in any desired direction. The data structures atrain/train are the appropriate tools for implementing BFS, DFS or any Search algorithms, Divide and Conquer algorithms, Branch and Bound type algorithms, etc. for big data. Recently Alam[1] has applied the r-train data structure in matrix multiplication method using

a parallel processing technique to reduce the complexity. In fact 'Data Structures for Big Data' [8] is to be regarded as a new subject, not just a new topic in the area of big data science as this subject seems to have become like 'Big Data Universe'.

A solid matrix can be regarded as a mathematical object which can facilitate operations on big data in many cases. Multidimensional structure [12, 17, 19] is quite popular for analytical databases that use online analytical processing (OLAP) applications. Analytical databases use these databases because of their ability to deliver answers to complex business queries swiftly. Data can be viewed from different angles, which gives a broader perspective of a problem unlike other models. The notion of multi-dimensional matrices studied by Solo in [17] (and also by Krattenthaler et.al. in [12]) is almost analogous to our notion of solid matrices; but the 'Theory of Solid Matrices/Latrices' presented in this chapter is precise, complete and sound, compatible and scalable with the real life application domains of any organization, easy to implement in computer memory and easy to be applied in the real problems of various fields of Science, Engineering, Statistics, OR, etc. to list a few only out of many.

The 'Theory of Solid Matrices/latrices' is an extension of the classical theory of matrices. The name 'Hematrix' stands for Heterogeneous Matrix, and the name 'Helatrix' stands for Heterogeneous Latrix. Each row in a hematrix/helatrix contains data of homogeneous datatype, but different rows contain data of heterogeneous datatype. This mathematical model is designed to make a logical storage structure for big data. The number of rows/columns in a hematrix/helatrix are scalable, i.e. can be made as large as desired. The Solid Hematrix/Helatrix is useful if the big data is a temporal big data. Otherwise, the logical storage structure '2-D Helatrix' (2-D Hematrix) is the appropriate model to store heterogeneous big data as the number of rows/columns in a 2-D helatrix can be scalable upto any big extent. Consequently, the data structure MT is useful for temporal homogeneous big data and the data structure MA is useful for temporal heterogeneous big data only, whereas the Atrain/train data structure can easily handle heterogeneous/homogeneous big data for storage in an ADS for immediate or future processing. How to manage the issue of memory fragmentation in ADS is yet to be solved. A hematrix/helatrix of big size may not be always possible to be implemented in a limited memory space of an autonomous computer. The powerful architecture of the distributed system ADS can handle the big data in any amount of 4Vs. The most important issue like "how to integrate all these 'new' (new data structures, new network topologies, new distributed systems, new logical models, etc) to make ultimately a single and simple system to the laymen users at their ground levels" is an inbuilt solution provided by the rich and highly scalable architecture of the ADS for big data. The implementation of big data using the data structures train or atrain or MT are shown here in autonomous computer systems initially, and then the progress is made to the newly proposed distributed system ADS for big data.

The next requirement to the computer scientists is to develop a new but simple big data language called by **"ADSL" (Atrain Distributed System Language)** which can easily be used by any laymen user at the PC of an ADS to download unlimited amount of relevant big data of any heterogeneous datatype (if not of homogeneous datatype) from the cyberspace, to upload unlimited amount of relevant big data of any heterogeneous datatype (if not of homogeneous datatype) in the cyberspace, to store the downloaded big data online in the coaches of PC/DCs of ADS in an organized way of atrain (train) Physiology, to answer any query be it arithmetical or statistical or relational query or imprecise query on big data universe.

References

1. Alam, B.: Matrix Multiplication using r-Train Data Structure. In: AASRI Conference on Parallel and Distributed Computing Systems, AASRI (Elsevier) Procedia 5, 189–193 (2013), doi: 10.1016/j.aasri.2013.10.077
2. Berman, J.J.: Principles of Big Data: Preparing, Sharing, and Analyzing Complex Information. Morgan Kaufmann (Elsevier) Publisher, USA (2013)
3. Biswas, R.: Heterogeneous Data Structure "r-Atrain". An International Journal (2012 International Information Institute of Japan & USA) 15(2), 879–902 (2012)
4. Biswas, R.: Heterogeneous Data Structure "r-Atrain", Chapter-12. In: Tripathy, B.K., Acharjya, D.P. (eds.) Global Trends in Knowledge Representation and Computational Intelligence, IGI Global, USA (2013), doi:10.4018/978–1–4666–4936–1
5. Biswas, R.: Region Algebra, Theory of Objects & Theory of Numbers. International Journal of Algebra 6(8), 1371–1417 (2012)
6. Biswas, R.: Theory of Solid Matrices & Solid Latrices, Introducing New Data Structures MA, MT: for Big Data. International Journal of Algebra 7(16), 767–789 (2013), http://dx.doi.org/10.12988/ija.2013.31093
7. Biswas, R.: Processing of Heterogeneous Big Data in an Atrain Distributed System (ADS) Using the Heterogeneous Data Structure r-Atrain. International Journal Computing and Optimization 1(1), 17–45 (2014)
8. Biswas, R.: Data Structures for Big Data. International Journal Computing and Optimization (in press)
9. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. MIT Press and McGraw-Hill (2001)
10. Feinleib, D.: Big Data Demystified: How Big Data Is Changing The Way We Live, Love And Learn. The Big Data Group Publisher, LLC, San Francisco (2013)
11. Franklin, J.L.: Matrix Theory, Mineola, N.Y. (2000)
12. Krattenthaler, C., Schlosser, M.: A New Multidimensional Matrix Inverse with Applications to Multiple q-series. Discrete Mathematics 204, 249–279 (1999)
13. Mayer-Schönberger, V., Cukier, K.: BIG DATA: A Revolution That Will Transform How We Live, Work, and Think. Eamon Dolan/Houghton Mifflin Harcourt Publisher (2013)
14. Needham, J.: Disruptive Possibilities: How Big Data Changes Everything. O'reilly Publisher, Cambridge (2013)

15. Simon, P.: Too Big to Ignore: The Business Case for Big Data. John Wiley & Sons, New Jersey (2013)
16. Sitarski, E.: HATs: Hashed array trees. Dr. Dobb's Journal 21(11) (1996), http://www.ddj.com/architect/184409965?pgno=5
17. Solo, A.M.G.: Multidimensional Matrix Mathematics: Part 1-6. In: Proceedings of the World Congress on Engineering, WCE 2010, London, June30-July 2, vol. III, pp. 1–6 (2010), http://www.iaeng.org/publication/WCE2010/WCE2010_pp1824-1828.pdf
18. Tanenbaum, A.S.: Computer Networks, 3rd edn. Pearson Education India, New Delhi (1993)
19. Thomsen, E.: OLAP Solutions: Building Multidimensional Information Systems, 2nd edn. John Wiley & Sons, USA (2002)

Big Data Time Series Forecasting Model: A Fuzzy-Neuro Hybridize Approach

Pritpal Singh

Abstract. Big data evolves as a new research domain in the era of 21st century. This domain concerns with the study of voluminous data sets with multiple factors, whose sizes are rapidly growing with the time. These types of data sets can be generated from various autonomous sources, such as scientific experiments, engineering applications, government records, financial activities, etc. With the rise of big data concept, demand for a new time series prediction models emerged. For this purpose, a novel big data time series forecasting model is introduced in this chapter, which is based on the hybridization of two soft computing (SC) techniques, viz., fuzzy set and artificial neural network. The proposed model is explained with the stock index price data set of State Bank of India (SBI). The performance of the model is verified with different factors, viz., two-factors, three-factors, and M-factors. Various statistical analyzes signify that the proposed model can take far better decision with the M-factors data set.

1 Introduction

Big data is the emerging trend of research, due to its capability to gather, store, manage large volume of data with high processing speed to discover hidden knowledge or information. The term *big data* can be defined in terms of any kind of data source that has three characteristics [1]: massive *volume* of data, high *velocity* of data, and wide *variety* of data. The term *big data* has been evolved due to growth in technology, scientific research, market, social media, etc. in recent years.

Time series data are highly non-stationary and uncertain in nature [4]. Therefore, to design a predictive model for the big data time series is a challenging task for the researchers in the domain of time series analysis and forecasting. For good forecasting accuracy, large number of information or factors should be considered

Pritpal Singh
School of Mathematics and Computer Applications,
Thapar University, Patiala-04, Punjab, India
e-mail: pritpal@tezu.ernet.in, pritpal.singh@thapar.edu

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_2

with enormous amount of observations. For example, the stock indices for most of the companies consist of four-factors, viz., Open, High, Low, and Close. Previously market analysts or researchers only employ *Close* factor to take the final decision. However, recent research trend shows that consideration of more than one factor at the time of final decision making improves the predictive skills of the models [7]. But more inclusion of factors increases the size as well as volume of the data set. Hence, one can face the following crucial issues at the time of designing predictive model for the big data as:

Models: Is it possible to predict the time series values in advance? If it is so, then which models are the best fitted for the big data that are characterized by different variables.

Quantity of Data: What amount of data (i.e., small or massive) needed for the prediction that fit the model well?

Improvement in Models: Is there any possibility to improve the efficiency of the existing time series models that can deal with the big data? If yes, then how it could be possible?

Factors: What are the factors that influence the time series prediction? Is there any possibility to deal with these factors together? Can integration of these factors affect the prediction capability of the models?

Analysis of Results: Are results given by the models statistically acceptable? If it is not, then which parameters are needed to be adjusted that influence the performance of the models.

Model Constraints: Can linear statistical or mathematical models successfully deal with the non-linear nature of the big data time series? If it is not possible, then what are the models and how they are advanced?

Data Preprocessing: Can data need to be transformed from one to another form? In general, what type of transformation is suitable for data that can directly be employed as input in the models?

Consequences of Prediction: What are possible consequences of time series prediction? Are there advance predictions advantageous for the society, politics and economics?

All these issues indicate the need for intelligent forecasting technique, which can discover useful information from the big data. For this purpose, soft computing (SC) technique can be adopted, which is widely used in designing intelligent or expert system, machine learning, artificial intelligence, pattern recognition, uncertainties and reasoning. The SC has been evolved as an amalgamated field of different methodologies such as fuzzy set, neural computing, evolutionary computing and probabilistic computing [3]. In this study, author investigates the application of two SC techniques by integrating them together, viz., fuzzy set and artificial neural network (ANN), for out-of-sample prediction in big data time series. The main aim of designing such a hybridized model is explained below.

1. **Lengths of Intervals:** In case of fuzzy-neuro hybridize modeling approach, initially time series data need to be fuzzified. For fuzzification of big data time series, determination of lengths of intervals is an important phase. But due to involvement of voluminous data with multiple factors, this makes the fuzzification process vary complex. Therefore, to resolve this research issue, an *Equal-interval based discretization (EIBD)* approach is introduced in this chapter, which discretize the big data time series in various equal lengths of intervals.
2. **Fuzzy Relations and Their Defuzzification:** After fuzzification of big data time series, fuzzy sets are further used to establish the fuzzy relations among them. This operation, no doubt, leads to evolve massive fuzzy relations. These fuzzy relations are the combination of previous states' (left-hand side) and current states' (right-hand side) of fuzzy sets. To obtain forecasting results out-of-sample from these fuzzy relations, we need to use previous states' fuzzified values. Hence, there is a need of an architecture that can process all these previous states' fuzzified values together, and generate the desired outputs. For this purpose, an ANN based architecture is designed, and integrates with this model, because ANN serves as a powerful tool that can establish the linear association between inputs and target outputs.

This chapter aims to propose a novel big data time series forecasting model, that can deal with the above research problems simultaneously. To demonstrate the application of the model, daily stock index price data set of State Bank of India (SBI) is used. The remainder of this chapter is organized as follows. Section 2 briefly reviews the concept of fuzzy set, big data, and M-factors fuzzy relation. Section 3 introduces the way to hybridize ANN with fuzzy set to solve big data time series forecasting problem. In Section 4, description of data set is provided. Section 5 presents data discretization approach and algorithm for the big data time series forecasting. Detail explanation of the proposed model is presented in Section 6. The performance of the model is assessed with various statistical parameters, which are discussed in Section 7. Section 8 presents all the experimental results. Finally, conclusion and discussion are presented in Section 9.

2 Foundations of Fuzzy Set

This section provides various definitions for the terminologies used throughout this chapter.

Definition 0.1. (Fuzzy set) [10]. A fuzzy set is a class with varying degrees of membership in the set. Let U be the universe of discourse, which is discrete and finite, then fuzzy set A can be defined as follows:

$$\tilde{A} = \{\mu_{\tilde{A}(x_1)}/x_1 + \mu_{\tilde{A}(x_2)}/x_2 + \ldots\} = \Sigma_i \mu_{\tilde{A}}(x_i)/x_i, \tag{1}$$

where $\mu_{\tilde{A}}$ is the membership function of $\tilde{A}$, $\mu_{\tilde{A}}$: $U \rightarrow [0,1]$, and $\mu_{\tilde{A}(x_i)}$ is the degree of membership of the element x_i in the fuzzy set $\tilde{A}$. Here, the symbol "+" indicates

the operation of union and the symbol "/" indicates the separator rather than the commonly used summation and division in algebra, respectively.

When U is continuous and infinite, then the fuzzy set A of U can be defined as:

$$\tilde{A} = \{\int \mu_{\tilde{A}(x_i)}/x_i\}, \forall x_i \in U, \tag{2}$$

where the integral sign stands for the union of the fuzzy singletons, $\mu_{\tilde{A}(x_i)}/x_i$.

Definition 0.2. (Big data) [1]. Big data is the capability to manage a huge volume of disparate data, at the right speed, and within the right time frame to allow real-time analysis and reaction. A data set can be called big data if it is formidable to perform capture, curation, analysis and visualization on it at the current technologies.

Definition 0.3. (M-factors time series). Let $O(t)$, $H(t)$, $L(t)$, $C(t)$, $F(t)$ be factors/ observations for the time series *Open*, *High*, *Low*, *Close*, and *Final Price*, respectively. If we only use $F(t)$ to solve the forecasting problems, then it is called a one-factor time series. If we use remaining secondary-factors/secondary-observations $O(t)$, $H(t)$, $L(t)$, $C(t)$ with $F(t)$ to solve the forecasting problems, then it is called M-factors time series.

Definition 0.4. (M-factors fuzzy relation). Let $O(t)$, $H(t)$, $L(t)$, $C(t)$, $F(t)$ be factors/ observations for the time series *Open*, *High*, *Low*, *Close*, and *Final Price*, respectively. Consider that all these observations can be fuzzified as: $O(t-n-1) = \tilde{O}_o$, $H(t-n-1) = \tilde{H}_h$, $L(t-n-1) = \tilde{L}_l$, $C(t-n-1) = \tilde{C}_c$, and $F(t+n) = \tilde{F}_f$. An M-factors fuzzy relation can be defined as:

$$\tilde{O}_o, \tilde{H}_h, \tilde{L}_l, \tilde{C}_c \rightarrow \tilde{F}_f, \tag{3}$$

where $\tilde{O}_o$, $\tilde{H}_h$, $\tilde{L}_l$, $\tilde{C}_c$ refer to as the RHS (right-hand side), and $\tilde{F}_f$ refer to the LHS (left-hand side) of the M-factors fuzzy relation, respectively. Here, $\tilde{O}_o$, $\tilde{H}_h$, $\tilde{L}_l$, $\tilde{C}_c$ and $\tilde{F}_f$ represent the fuzzified time series values for the historical time series data set with the indices o, h, l, c and f, respectively.

3 Fuzzy-Neuro Hybridization and Big Data Time Series

In the following, we describe the basics of ANN technique along with the way to hybridize this technique with the fuzzy set in terms of big data modeling approach.

3.1 Artificial Neural Network: An Overview

ANNs are massively parallel adaptive networks of simple nonlinear computing elements called *neurons* which are intended to abstract and model some of the functionality of the human nervous system in an attempt to partially capture some of

its computational strengths [2]. The neurons in an ANN are organized into different layers. Inputs to the network are existed in the input layer; whereas outputs are produced as signals in the output layer. These signals may pass through one or more intermediate or *hidden* layers which transform the signals depending upon the neuron signal functions.

The neural networks are classified into either single-layer (SL) or multi-layer (ML) neural networks. This layer exists in between input and output layers. A SL neural network is formed when the nodes of input layer are connected with processing nodes with various weights, resulting to form a series of output nodes. A ML neural network architecture can be developed by increasing the number of layers in SL neural network.

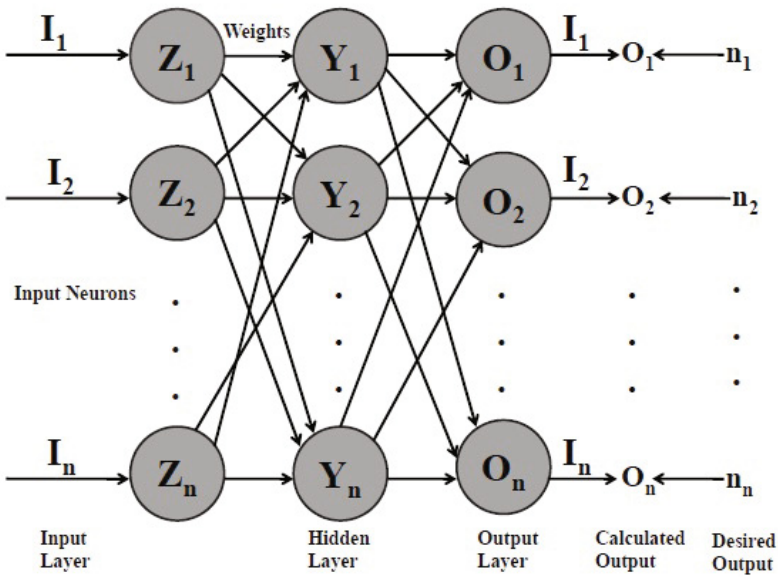


Fig. 1 A BPNN architecture with one hidden layer

In literature, several types of neural networks could be found, but usually FFNN (Feed-forward neural network) and BPNN (Back-propagation neural network) are used in time series forecasting (especially seasonal forecasting). The main objective of using the BPNN with ML neural network is to minimize the output error obtained from the difference between the calculated output $(o_1, o_2, \ldots, o_n)$ and target output $(n_1, n_2, \ldots, n_n)$ of the neural network by adjusting the weights. So in BPNN, each information is sent back again in the reverse direction until the output error is very small or zero. An architecture of the BPNN is depicted in Fig. 1. The BPNN is trained under the process of three phases: (a) using FFNN for training process

of input information. Adjustment of weights and nodes are made in this phase, (b) to calculate the error, and (c) update the weights. More detail description on applications of ANN (especially BPNN) could be found in the article of Singh and Borah [6].

3.2 *Fuzzy-Neuro Hybridized Approach: A New Paradigm for the Big Data Time Series Forecasting*

Hybridization of ANN with fuzzy set is a significant development in the domain of forecasting. It is an ensemble of merits of ANN and fuzzy set technique, by substituting the demerits of one technique by the merits of another technique. This includes various advantages of ANN, such as parallel processing, handling of large data set, fast learning capability, etc. Handling of imprecise/ uncertainty and linguistic variables are done through the utilization of fuzzy set. Besides these advantages, the fuzzy-neuro hybridization help in designing complex decision-making systems. Recent application of this approach could be found in the article of Singh and Borah [5].

The performance of the neural network architecture relies on number of layers, number of nodes in each layer, and number of interconnection links with the nodes [9]. Since, a neural network with more than three layers generate arbitrarily complex decision regions. Therefore, the proposed model is simulated with a single hidden layer with one input layer and one output layer. The neural network architecture for this purpose is shown in Fig. 2.

Before giving inputs to the neural network, the M-factors fuzzy relations are established based on Definition 0.4 among the fuzzified stock index values for the observations O, H, L, C and F as shown in Eq. 4.

$$\begin{aligned}
\tilde{O}_o(t-1), \tilde{H}_h(t-1), \tilde{L}_l(t-1), \tilde{C}_c(t-1) &\rightarrow \tilde{F}_f(t) \\
\tilde{O}_o(t-2), \tilde{H}_h(t-2), \tilde{L}_l(t-2), \tilde{C}_c(t-2) &\rightarrow \tilde{F}_f(t+1) \\
&\cdots \\
\tilde{O}_o(t-n-1), \tilde{H}_h(t-n-1), \tilde{L}_l(t-n-1), \tilde{C}_c(t-n-1) &\rightarrow \tilde{F}_f(t+n). \qquad (4)
\end{aligned}$$

Here, each $\tilde{O}_o$, $\tilde{H}_h$, $\tilde{L}_l$, $\tilde{C}_c$, and $\tilde{F}_f$ denotes the fuzzified stock index values for the observations O, H, L, C, and F, respectively. Now, each LHS of the fuzzy relation is employed as inputs in the neural network architecture. Hence, the main objective of this neural network architecture is to yield F for the stock index data set in the form of fuzzified values $\tilde{F}_f$ (corresponding to each day). Later, these fuzzified values are employed for the defuzzification operation.

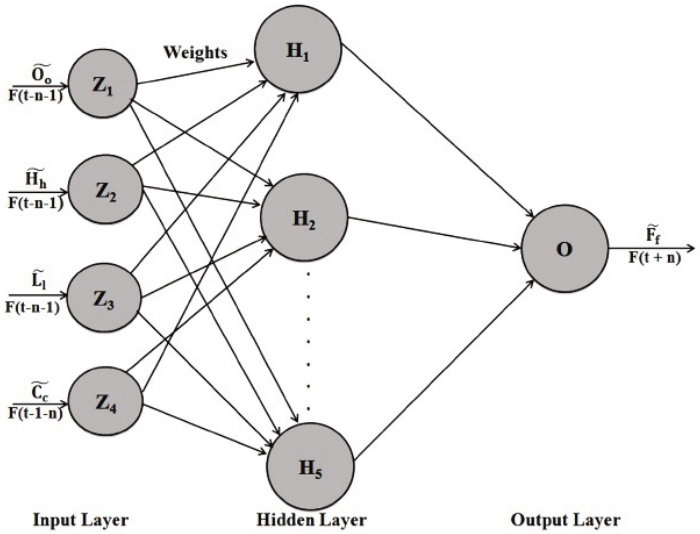


Fig. 2 Architecture of the proposed neural network

Table 1 Daily stock index price list of SBI (in rupee)

Date	O	H	L	C	F
6/4/2012	2064.00	2110.00	2061.00	2082.75	2079.44
6/5/2012	2100.00	2168.00	2096.00	2158.25	2130.56
6/6/2012	2183.00	2189.00	2156.55	2167.95	2174.13
6/7/2012	2151.55	2191.60	2131.50	2179.45	2163.53
6/10/2012	2200.00	2217.90	2155.50	2164.80	2184.55
6/11/2012	2155.00	2211.85	2132.60	2206.15	2176.40
6/12/2012	2210.00	2244.00	2186.00	2226.05	2216.51
6/13/2012	2212.00	2218.70	2144.10	2150.25	2181.26
6/14/2012	2168.45	2189.85	2147.00	2183.10	2172.10
6/17/2012	2216.00	2231.90	2081.60	2087.95	2154.36
...	...	...	...	...	...
7/29/2012	1951.25	2040.00	1941.00	2031.75	1991.00

4 Description of Data Set

To verify the proposed model, daily stock index price data set of SBI for the period 6/4/2012 – 7/29/2012 (format: mm-dd-yy) is used. This data set is collected from the website: `http://in.finance.yahoo.com`. A sample of this data set is shown in Table 1. This data set consists of five-factors, viz., *O*, *H*, *L*, *C*, and *F*. The main objective of this model is to forecast *F* for the historical data set.

5 Proposed Approach and Algorithm

This section first presents an approach for partitioning the universe of discourse into intervals followed by an algorithm for fuzzy-neuro hybridization.

5.1 EIBD Approach

In this subsection, we propose a new discretization approach referred to as *EIBD* for determining the universe of discourse of the historical time series data set, and partitioning it into different lengths of intervals. To explain this approach, the daily stock index price data set of SBI from the period $6/4/2012 - 7/29/2012$, shown in Table 1, is employed. Each step of the approach is explained below.

- Compute range (R) of a sample, $S = \{x_1, x_2, \ldots, x_n\}$ as:

$$R = Max_{value} - Min_{value}, \tag{5}$$

  where Max_{value} and Min_{value} are the maximum and minimum values of S respectively.
  From Table 1, Max_{value} and Min_{value} for the whole sample (S) are 2252.55 and 1931.50, respectively. Therefore, the range R for this data set is computed as:

$$R = 2252.55 - 1931.50 = 321.05$$

- Split the data range R as:

$$W = \frac{R}{Z} \tag{6}$$

  where Z is the size of the sample S.
  Based on Eq. 6, we can compute W as:

$$K = \frac{321.05}{200} = 1.6, \text{where sample size } Z = 200.$$

- Define the universe of discourse U of the sample S as:

$$U = [L_b, U_b], \tag{7}$$

  where $L_b = Min_{value} - K$, and $U_b = Max_{value} + K$.
  Based on Eq. 7, the universe of discourse U is:

$$U = [1929.9, 2254.15], \tag{8}$$

  where $L_b = 1931.50 - 1.6$, and $U_b = 2252.55 + 1.6$.

- Compute the length of the interval (L) as:

$$L = \frac{(U_b - L_b)}{\bar{A}} \times SD \tag{9}$$

where $\bar{A}$ = Mean of the whole sample S, and SD = Standard deviation of the whole sample S.
The L of the sample S is obtained as:

$$L = \frac{(2254.15 - 1929.9)}{2153.88} \times 63.48 = 9.73$$

Here, $\bar{A} = 2153.88$ and $SD = 63.48$.

- Partition the universe of discourse U into equal lengths of intervals as:

$$u_i = [L(i), U(i)], i = 1, 2, 3, \ldots; 1 \leq U(i) \leq U_b; u_i \in U; \tag{10}$$

where $L(i) = L_b + (i-1) \times L$, and $U(i) = L_b + i \times L$.
Based on Eq. 10, intervals for the universe of discourse U are:

$$a_1 = [1929.9, 1939.5], a_2 = [1939.5, 1949], \ldots, a_{34} = [2245.3, 2254.15]$$

- Allocate the elements to their corresponding intervals.

5.2 *Algorithm for the Big Data Time Series Forecasting Model*

In this subsection, an algorithm is presented, which is entitled as *Fuzzy-Neuro Forecasting Model for Big Data*. This algorithm integrates the *EIBD* approach and ANN. Basic steps of the algorithm are presented below:

- Partition the universe of discourse into different equal lengths of intervals based on the *EIBD* approach.
- Define linguistic terms for each of the interval.
- Fuzzify the big data time series.
- Establish the M-factors fuzzy relations based on Definition 0.4.
- Defuzzify the fuzzified time series values based on the proposed neural network architecture.

6 Fuzzy-Neuro Forecasting Model for Big Data: Detail Explanation

The proposed model is applied to forecast the daily stock index price of SBI from the period $6/4/2012 - 7/29/2012$. Each step of the model is elucidated below.

- *Partition the universe of discourse into equal lengths of intervals*: Define the universe of discourse U for the historical SBI data set. Based on Eq. 7, we can define the universe of discourse U as: $U = [1929.9, 2254.15]$. Then, based on the

EIBD approach, the universe of discourse U is partitioned into 34 equal lengths of intervals, which can be represented as a_i, for $i = 1,2,\ldots,n$; and $n \leq 34$.

- *Define linguistic terms for each of the interval*: Assume that the historical time series data set is distributed among n intervals (i.e., $a_1, a_2, \ldots,$ and a_n). Therefore, define n linguistic variables $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_n$, which can be represented by fuzzy sets, as shown below:

$$\begin{aligned}
\tilde{A}_1 &= 1/a_1 + 0.5/a_2 + 0/a_3 + \ldots + 0/a_{n-2} + 0/a_{n-1} + 0/a_n, \\
\tilde{A}_2 &= 0.5/a_1 + 1/a_2 + 0.5/a_3 + \ldots + 0/a_{n-2} + 0/a_{n-1} + 0/a_n, \\
\tilde{A}_3 &= 0/a_1 + 0.5/a_2 + 1/a_3 + \ldots + 0/a_{n-2} + 0/a_{n-1} + 0/a_n, \\
&\vdots \\
\tilde{A}_j &= 0/a_1 + 0/a_2 + 0/a_3 + \ldots + 0/a_{n-2} + 0.5/a_{n-1} + 1/a_n. \qquad (11)
\end{aligned}$$

Obtain the degree of membership of each day's time series value belonging to each fuzzy set $\tilde{A}_i$. Here, maximum degree of membership of fuzzy set $\tilde{A}_i$ occurs at interval a_i, and $1 \leq i \leq n$.

We define 34 linguistic variables $\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_{34}$ for the stock index data set, because the data set is distributed among 34 intervals (i.e., $a_1, a_2, \ldots,$ and a_{34}). All these defined linguistic variables are listed as below:

$$\begin{aligned}
\tilde{A}_1 &= 1/a_1 + 0.5/a_2 + 0/a_3 + \ldots + 0/a_{n-2} + 0/a_{n-1} + 0/a_{34}, \\
\tilde{A}_2 &= 0.5/a_1 + 1/a_2 + 0.5/a_3 + \ldots + 0/a_{n-2} + 0/a_{n-1} + 0/a_{34}, \\
\tilde{A}_3 &= 0/a_1 + 0.5/a_2 + 1/a_3 + \ldots + 0/a_{n-2} + 0/a_{n-1} + 0/a_{34}, \\
&\vdots \\
\tilde{A}_{34} &= 0/a_1 + 0/a_2 + 0/a_3 + \ldots + 0/a_{n-2} + 0.5/a_{n-1} + 1/a_{34}. \qquad (12)
\end{aligned}$$

- *Fuzzify the historical big data time series. If one day's time series value belongs to the interval a_i, then it is fuzzified into $\tilde{A}_i$, where $1 \leq i \leq n$.*

In order to fuzzify the historical time series data set, it is essential to obtain the degree of membership value of each observation belonging to each $\tilde{A}_i$ $(i = 1,2,\ldots,n)$ for each day. If the maximum membership value of one day's observation occurs at interval a_i, and $1 \leq i \leq n$, then the fuzzified value for that particular day is considered as $\tilde{A}_i$. For example, the stock index price of $6/4/2012$ for the observation O belongs to the interval a_{15} with the highest degree of membership value 1 (based on Eq. 12), so it is fuzzified to $\tilde{A}_{15}$. But for the convenience, this fuzzified value is represented with the help of its corresponding factor's tag, which is O. Hence, the fuzzified value $\tilde{A}_{15}$ can be represented as $\tilde{O}_{15}$. Similarly, the stock index price of $6/4/2012$ for the observation H belongs to the interval a_{19} with the highest degree of membership value 1 (based on Eq. 12), so it is fuzzified to $\tilde{H}_{19}$. In this way, we have fuzzified each observation of the historical time series data set. The fuzzified historical time series data set is presented in Table 2.

Table 2 Fuzzified stock index data set

Date	O	Fuzzified O	H	Fuzzified H	L	Fuzzified L	C	Fuzzified C	F	Fuzzified F
6/4/2012	2064.00	$\tilde{O}_{15}$	2110.00	$\tilde{H}_{19}$	2061.00	$\tilde{L}_{14}$	2082.75	$\tilde{C}_{16}$	2079.44	$\tilde{F}_{16}$
6/5/2012	2100.00	$\tilde{O}_{18}$	2168.00	$\tilde{H}_{25}$	2096.00	$\tilde{L}_{18}$	2158.25	$\tilde{C}_{24}$	2130.56	$\tilde{F}_{21}$
6/6/2012	2183.00	$\tilde{O}_{27}$	2189.00	$\tilde{H}_{28}$	2156.55	$\tilde{L}_{24}$	2167.95	$\tilde{C}_{25}$	2174.13	$\tilde{F}_{26}$
6/7/2012	2151.55	$\tilde{O}_{24}$	2191.60	$\tilde{H}_{28}$	2131.50	$\tilde{L}_{22}$	2179.45	$\tilde{C}_{27}$	2163.53	$\tilde{F}_{25}$
6/10/2012	2200.00	$\tilde{O}_{29}$	2217.90	$\tilde{H}_{31}$	2155.50	$\tilde{L}_{24}$	2164.80	$\tilde{C}_{25}$	2184.55	$\tilde{F}_{27}$
6/11/2012	2155.00	$\tilde{O}_{24}$	2211.85	$\tilde{H}_{30}$	2132.60	$\tilde{L}_{22}$	2206.15	$\tilde{C}_{29}$	2176.40	$\tilde{F}_{26}$
6/12/2012	2210.00	$\tilde{O}_{30}$	2244.00	$\tilde{H}_{33}$	2186.00	$\tilde{L}_{27}$	2226.05	$\tilde{C}_{31}$	2216.51	$\tilde{F}_{30}$
6/13/2012	2212.00	$\tilde{O}_{30}$	2218.70	$\tilde{H}_{31}$	2144.10	$\tilde{L}_{23}$	2150.25	$\tilde{C}_{24}$	2181.26	$\tilde{F}_{27}$
6/14/2012	2168.45	$\tilde{O}_{25}$	2189.85	$\tilde{H}_{28}$	2147.00	$\tilde{L}_{23}$	2183.10	$\tilde{C}_{27}$	2172.10	$\tilde{F}_{26}$
6/17/2012	2216.00	$\tilde{O}_{30}$	2231.90	$\tilde{H}_{32}$	2081.60	$\tilde{L}_{16}$	2087.95	$\tilde{C}_{17}$	2154.36	$\tilde{F}_{24}$
...	...	...	...	...	...	...	...	...	...	...
7/29/2012	1951.25	$\tilde{O}_{3}$	2040.00	$\tilde{H}_{12}$	1941.00	$\tilde{L}_{2}$	2031.75	$\tilde{C}_{11}$	1991.00	$\tilde{F}_{7}$

- *Establish the M-factors fuzzy relations between the fuzzified time series values*: Based on Definition 0.4, we can establish the M-factors fuzzy relations between the fuzzified time series values. For example, in Table 2, the fuzzified stock index values of the observations O, H, L, C, and F for the day $6/4/2012$ are: $\tilde{O}_{15}$, $\tilde{H}_{19}$, $\tilde{L}_{14}$, $\tilde{C}_{16}$, and $\tilde{F}_{16}$, respectively. Here, to establish the M-factors fuzzy relation among these fuzzified values, it is considered that the fuzzified value $\tilde{F}_{16}$, is caused by the previous four fuzzified values: $\tilde{O}_{15}$, $\tilde{H}_{19}$, $\tilde{L}_{14}$, and $\tilde{C}_{16}$. Hence, the M-factors fuzzy relation is represented in the following form:

$$\tilde{O}_{15}, \tilde{H}_{19}, \tilde{L}_{14}, \tilde{C}_{16} \rightarrow \tilde{F}_{16} \tag{13}$$

  Here, LHS of the M-factors fuzzy relation is called the previous state, whereas RHS of the M-factors fuzzy relation is called the current state.

  For the M-factors fuzzy relation as shown in Eq. 13, if we use current state's fuzzified value (i.e., $\tilde{F}_{16}$) for defuzzification operation, then the prediction scope of the model lies within the sample. However, for most of the real and complex problems, out of sample prediction (i.e., advance prediction) is very much essential. Therefore, in this model, the previous state's fuzzified values (i.e., $\tilde{O}_{15}$, $\tilde{H}_{19}$, $\tilde{L}_{14}$, and $\tilde{C}_{16}$) are used to obtain the forecasting results.

  The M-factors fuzzy relations obtained for the fuzzified daily stock index price are listed in Table 3. In this table, each symbol ? represent the *desired output* for corresponding day t in the symbol $<>$, which would be determined by the proposed model.
- *Defuzzify the fuzzified time series data set*: This model uses the BPNN algorithm to defuzzify the fuzzified time series data set. The neural network architecture which is used for this purpose is presented in earlier section. The proposed model is based on the M-factors fuzzy relations. The steps involve in the defuzzification operation are explained below.
  - For forecasting day $D(t+n)$, obtain the M-factors fuzzy relation, which can be represented in the following form:

$$\tilde{O}_o(t-n-1), \tilde{H}_h(t-n-1), \tilde{L}_l(t-n-1), \tilde{C}_c(t-n-1) \rightarrow ?\langle t+n \rangle, \tag{14}$$

Table 3 M-factors fuzzy relations for the fuzzified daily stock index data set

M-factors fuzzy relation
$\tilde{O}_{15},\tilde{H}_{19},\tilde{L}_{14},\tilde{C}_{16} \rightarrow ?\langle 6/4/2012\rangle$
$\tilde{O}_{18},\tilde{H}_{25},\tilde{L}_{18},\tilde{C}_{24} \rightarrow ?\langle 6/5/2012\rangle$
$\tilde{O}_{27},\tilde{H}_{28},\tilde{L}_{24},\tilde{C}_{25} \rightarrow ?\langle 6/6/2012\rangle$
$\tilde{O}_{24},\tilde{H}_{28},\tilde{L}_{22},\tilde{C}_{27} \rightarrow ?\langle 6/7/2012\rangle$
$\tilde{O}_{29},\tilde{H}_{31},\tilde{L}_{24},\tilde{C}_{25} \rightarrow ?\langle 6/10/2012\rangle$
$\tilde{O}_{24},\tilde{H}_{30},\tilde{L}_{22},\tilde{C}_{29} \rightarrow ?\langle 6/11/2012\rangle$
$\tilde{O}_{30},\tilde{H}_{33},\tilde{L}_{27},\tilde{C}_{31} \rightarrow ?\langle 6/12/2012\rangle$
$\tilde{O}_{30},\tilde{H}_{31},\tilde{L}_{23},\tilde{C}_{24} \rightarrow ?\langle 6/13/2012\rangle$
$\tilde{O}_{25},\tilde{H}_{28},\tilde{L}_{23},\tilde{C}_{27} \rightarrow ?\langle 6/14/2012\rangle$
$\tilde{O}_{30},\tilde{H}_{32},\tilde{L}_{16},\tilde{C}_{17} \rightarrow ?\langle 6/17/2012\rangle$
...
$\tilde{O}_{3},\tilde{H}_{12},\tilde{L}_{2},\tilde{C}_{11} \rightarrow ?\langle 7/29/2012\rangle$

where $t+n$ represents a day which we want to forecast. Here, $\tilde{O}_o(t-n-1),\tilde{H}_h(t-n-1),\tilde{L}_l(t-n-1),\tilde{C}_c(t-n-1)$ are the previous state's fuzzified values for the day, $D(t-n-1)$. Here, o, h, l, and c represent the indices of the fuzzy sets for the observations O, H, L, and C, respectively.

– Replace each previous state's fuzzified value of the fuzzy relation as shown in Eq. 14 with their corresponding indices as:

$$o,h,l,c \rightarrow ?\langle t+n\rangle \tag{15}$$

– Use the indices of Eq. 15 as inputs in the proposed BPNN architecture to compute the desired output ? for the corresponding day t+n. After sufficient number of epochs, output generated by the neural network would be in the form of f. Here, f represents the index of the forecasted fuzzy set for the observation F, which can be represented in the form of fuzzified value $\tilde{F}_f$.
– Find the interval, where the maximum membership value for the fuzzified value $\tilde{F}_f$ occurs. Let this interval be a_i, and its corresponding mid-value be M_i. The mid-value M_i for any interval can be computed by taking the mean of lower and upper bounds of the interval.
– Apply the following formula to calculate the forecasted value for day, $D(t+n)$ as:

$$Forecast(t+n) = \frac{M_i \times f}{j}. \tag{16}$$

Here, j represents the index of the actual fuzzified value $\tilde{A}_j$, which is, highly associated with the interval a_i (refer to Eq. 11).

During the learning process of the BPNN, a number of experiments were carried out to set additional parameters, viz., initial weight, learning rate, epochs, learning radius and activation function to obtain the optimal results, and we have chosen the

Table 4 Additional parameters and their values during the learning process of the BPNN

S. No.	Additional Parameter	Input Value
1	Initial weight	0.3
2	Learning rate	0.5
3	Epochs	10000
4	Learning radius	3
5	Activation function	Sigmoid

ones that exhibit the best behavior in terms of accuracy. The determined optimal values of all these parameters are listed in Table 4.

7 Performance Analysis Parameters

The performance of the proposed model is evaluated with the help of mean of the observed and predicted values, root mean square error (RMSE), average forecasting error rate (AFER), Theil's U Statistic, and correlation coefficient (CC). All these parameters are defined as follows:

- The mean can be defined as:

$$\bar{A} = \frac{\sum_{i=1}^{n} Act_i}{N} \tag{17}$$

- The *RMSE* can be defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Fore_i - Act_i)^2}{N}} \tag{18}$$

- The *AFER* can be defined as:

$$AFER = \frac{|Fore_i - Act_i|/Act_i}{N} \times 100\% \tag{19}$$

- The formula used to calculate Theil's U statistic [8] is:

$$U = \frac{\sqrt{\sum_{i=1}^{N} (Act_i - Fore_i)^2}}{\sqrt{\sum_{i=1}^{N} Act_i^2} + \sqrt{\sum_{i=1}^{N} Fore_i^2}} \tag{20}$$

- CC can be defined as:

$$CC = \frac{N \sum Act_i Fore_i - (\sum Act_i)(\sum Fore_i)}{\sqrt{N(\sum Act_i^2) - (\sum Act_i)^2} \sqrt{N(\sum Fore_i^2) - (\sum Fore_i)^2}} \tag{21}$$

Table 5 A sample of advance prediction of the daily stock index price list of SBI (in rupee)

Date	Final Price (F)	Forecasted Price
6/4/2012	2079.44	2079.30
6/5/2012	2130.56	2120.94
6/6/2012	2174.13	2188.98
6/7/2012	2163.53	2106.75
6/10/2012	2184.55	2262.31
6/11/2012	2176.40	2093.34
6/12/2012	2216.51	2198.75
6/13/2012	2181.26	2246.70
6/14/2012	2172.10	2096.44
6/17/2012	2154.36	2135.02
...	...	
7/29/2012	1991.00	2134.29

Here, each $Fore_i$ and Act_i is the forecasted and actual value of day i respectively, N is the total number of days to be forecasted. In Eq. 17, $\{A_1, A_2, \ldots, A_N\}$ are the observed values of the actual time series data set, and $\bar{A}$ is the mean value of these observations. Similarly, mean for the predicted time series data set is computed. For a good forecasting, the observed mean should be closed to the predicted mean. In Eqs. 18 and 19, smaller values of the RMSE and AFER indicate good forecasting accuracy. In Eq. 20, U is bound between 0 and 1, with values closer to 0 indicating good forecasting accuracy. In Eq. 21, the value of CC is such that $-1 < CC < +1$. A CC value greater than 0.5 is generally considered as good forecasting. However, the CC value greater than 0.8 is considered as the best forecasting.

8 Empirical Analysis

This section presents the forecasting results of the proposed model. This model is validated with the stock index data set of SBI, as mentioned in earlier. For obtaining the results, O, H, L, and C observations are employed as the main factors, whereas observation F is chosen as the main forecasting objective. Hence, the proposed model is trained with the M-factors fuzzy relations. We also conduct the experiment with the two-factors and three-factors.

8.1 Forecasting with the M-factors

Advance predicted values of the daily stock index price of SBI from the period $6/4/2012 - 7/29/2012$, for the M-factors fuzzy relations are presented in Table 5. The proposed model is also tested with the different orders of hidden nodes, viz., 6,

Table 6 Performance analysis of the model for the different orders of hidden nodes with the M-factors fuzzy relations

	Hidden nodes				
Statistics	5	6	7	8	9
AFER	1.93%	1.71%	1.27%	1.97%	1.82%

7, 8, and 9. The performance of the model is evaluated with the *AFER*. The *AFER* for the different orders of hidden nodes are compared in Table 6. In this case, the optimal *AFER* is 1.27% for the hidden nodes 7. The curves of the actual and the forecasted stock index prices with different orders of hidden nodes are shown in Fig. 3. It is obvious that the forecasted results are very close to that of actual values.

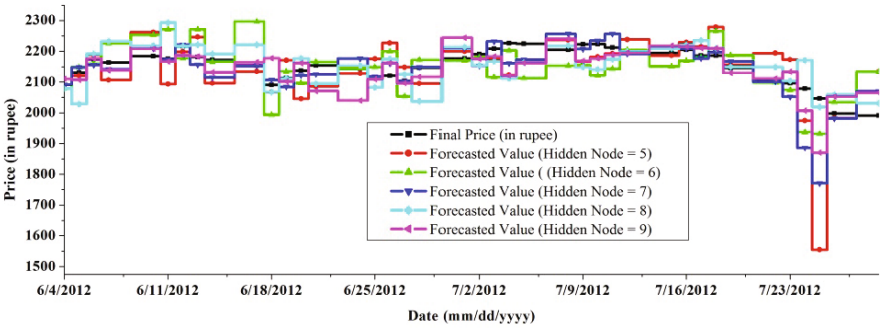


Fig. 3 Comparison curves of actual price and forecasted prices for the stock index price of SBI with different orders of hidden nodes

8.2 Forecasting with Two-factors

This subsection presents the forecasting results of the stock index price of SBI with the two-factors, viz., *O* and *H*. The proposed model is also verified with the different orders of hidden nodes, viz., 6, 7, 8, and 9. The performance of the model in terms of the *AFERs* are compared in Table 7. From the empirical analyzes between Tables 6 and 7, it is obvious that the proposed model doesn't able to take decision well with the two-factors as compared to the M-factors.

8.3 Forecasting with Three-factors

This subsection presents the forecasting results of the stock index price of SBI with the three-factors, viz., *O*, *H*, and *L*. The proposed model is also verified with the different orders of hidden nodes, viz., 6, 7, 8, and 9. Performance of the model in terms of the *AFER* is presented in Table 8. From the empirical analyzes between

Table 7 Performance analysis of the model for the different orders of hidden nodes with the two-factors fuzzy relations

	Hidden nodes				
Statistics	5	6	7	8	9
AFER	4.83%	4.40%	4.01%	3.75%	3.50%

Table 8 Performance analysis of the model for the different orders of hidden nodes with the three-factors fuzzy relations

	Hidden nodes				
Statistics	5	6	7	8	9
AFER	3.63%	3.20%	3.01%	2.95%	3.50%

Tables 7 and 8, it is obvious that the forecasting accuracy of the proposed model with respect to the three-factors is slightly better than the two-factors. However, from the overall empirical analyzes (from Tables 6-8), it is obvious that results are entirely consistent with the M-factors time series data set.

8.4 Statistical Significance

The forecasting results of stock index price of SBI with the M-factors in terms of the $AFER$ performs the best in comparison to the two-factors and the three-factors. To verify the statistical significance of the model, its performance is evaluated with various statistical parameters (as discussed in Section 7). Empirical analyzes are depicted in Table 9. From Table 9, it is obvious that the mean of observed values are close to the mean of predicted values. Forecasted results in terms of the $RMSE$ indicate very small error rate. In Table 9, the U values are closer to 0, which indicate the effectiveness of the proposed model. Hence, from the empirical analyzes, it is obvious that the proposed model is quite efficient in forecasting the daily stock index price of SBI (with the M-factors) with a very small error. The CC values between actual and predicted values also indicate the efficiency of the proposed model. However, the optimal $AFER$, $RMSE$, and CC values are found for the hidden nodes 7. Hence, it can be concluded that the application of 7 hidden nodes with the M-factors time series data set outperforms the two-factors and three-factors time series data sets.

Table 9 Performance analysis of the model for the different orders of hidden nodes with the M-factors fuzzy relations.

	Hidden nodes				
Statistics	5	6	7	8	9
Mean Observed	2152.02	2152.02	2152.02	2152.02	2152.02
Mean Predicted	2148.91	2143.68	2147.68	2154.29	2142.36
RMSE	51.35	48.26	36.41	53.37	52.39
U	0.01	0.01	0.01	0.01	0.01
CC	0.71	0.68	0.86	0.64	0.69

9 Conclusion and Discussion

This chapter presents a novel time series forecasting model for the big data time series using the M-factors fuzzy relations. In addition to this, this chapter also introduces the way to hybridize the two SC techniques, viz., fuzzy set and ANN, to resolve the domain specific problem. The proposed model establishes the linear association among the various fuzzified observations, and takes the decision from these fuzzy relations. Hence, the proposed model is applicable to the problems where massive fuzzy sets are involved. The proposed model is also verified with the different number of hidden nodes. The forecasting results are then analyzed with the various statistical parameters. These results clearly indicate that the integration of the M-factors in forecasting the big data time series improve the performance of the model.

In this model, we also introduce a new data discretization approach *EIBD*, which initially partitions the big data time series into various equal lengths of intervals. These evolved intervals are then used to fuzzify the sample data set. Later, these fuzzified time series values are employed to establish the M-factors fuzzy relations. Then, all these fuzzy relations are used as inputs in the designed BPNN architecture to take the final decisions.

The combination of two techniques always leads to the development of new architecture, which should be more advantageous and expert, providing robust, cost effective, and approximate solution, in comparison to conventional techniques. However, this hybridization should be carried out in a reasonable, rather than an expensive or a complicated, manner. In this study, two powerful SC techniques, viz., fuzzy set and ANN, are hybridized together in a very simple and cost effective way, so that the researchers in this domain can employ this model for further analysis of different kinds of big data time series.

References

1. Chen, C.L.P., Zhang, C.-Y.: Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. Information Sciences 275, 314–347 (2014)
2. Kumar, S.: Neural Networks: A Classroom Approach. Tata McGraw-Hill Education Pvt. Ltd., New Delhi (2004)

3. Ko, M., Tiwari, A., Mehnen, J.: A review of soft computing applications in supply chain management. Applied Soft Computing 10, 3–14 (2010)
4. Singh, P., Borah, B.: An efficient time series forecasting model based on fuzzy time series. Engineering Applications of Artificial Intelligence 26, 2443–2457 (2013)
5. Singh, P., Borah, B.: High-order fuzzy-neuro expert system for daily temperature forecasting. Knowledge-Based Systems 46, 12–21 (2013)
6. Singh, P., Borah, B.: Indian summer monsoon rainfall prediction using artificial neural network. Stochastic Environmental Research and Risk Assessment 27, 1585–1599 (2013)
7. Singh, P., Borah, B.: Forecasting stock index price based on M-factors fuzzy time series and particle swarm optimization. International Journal of Approximate Reasoning 55, 812–833 (2014)
8. Theil, H.: Applied Economic Forecasting. Rand McNally, New York (1996)
9. Wilson, I.D., Paris, S.D., Ware, J.A., Jenkins, D.H.: Residential property price time series forecasting with neural networks. Knowledge-Based Systems 15, 335–341 (2002)
10. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)

Learning Using Hybrid Intelligence Techniques

Sujata Dash

Abstract. This chapter focuses on a few key applications of hybrid intelligence techniques in the field of feature selection and classification. Hybrid intelligent techniques have been used to develop an effective and generalized learning model for solving these applications. The first application employs a new evolutionary hybrid feature selection technique for microarray datasets, which is implemented in two stages by integrating correlation-based binary particle swarm optimization (BPSO) with rough set algorithm to identify non-redundant genes capable of discerning between all objects. The other applications discussed are for evaluating the relative performance of different supervised classification procedures using hybrid feature reduction techniques. Correlation based Partial Least square hybrid feature selection method is used for feature extraction and the experimental results show that Partial Least Squares (PLS) regression method is an appropriate feature selection method and a combined use of different classification and feature selection approaches make it possible to construct high performance classification models for microarray data. Another hybrid algorithm, Correlation based reduct algorithm (CFS-RST) is used as a filter to eliminate redundant attributes and minimal reduct set is produced by rough sets. This method improves the efficiency and decreases the complexity of the classical algorithm. Extensive experiments are conducted on two public multi-class gene expression datasets and the experimental results show that hybrid intelligent methods are highly effective for selecting discriminative genes for improving the classification accuracy. The experimental results of all the applications indicate that, all the hybrid intelligent techniques discussed here have shown significant improvements in most of the binary and multi-class microarray datasets.

Sujata Dash
Sujata Dash, North Orissa University, Baripada, Odisha, India
e-mail: sujata238dash@gmail.com

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_3

1 Introduction

With the growth of high-throughput bio-technologies, feature selection has found its use in the reduction of huge quantity of generated data [1]. Three major types of feature selection techniques have been widely used for selecting and reducing genes or features from microarray data. The first type of technique is known as "filter" method which does not optimize the classification accuracy of the classifiers directly. But it attempts to select genes using some evaluation criterion such as correlation based feature selection, χ^2 - statistic [2], t-statistic [3], ReliefF [4], Information Gain [5] and Gain Ratio [6]. In this approach, gene selection and classification process are separated as shown in Figure 1. However, the selected gene subset is not adequate to increase the performance of the classifier because crucial information is being lost to discriminate the sample and identify the target gene [7]. Generally, genes are connected by various pathways and functioning a group. These selection methods often miss the important bio-pathways information. The second selection mechanism is "wrapper" technique which evaluates the identified gene subset according to their power to improve classification accuracy [8]. The classifier is wrapped in the feature selection process which is shown in Figure 2.

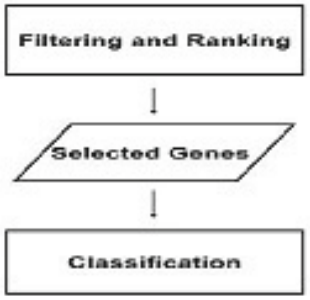


Fig. 1 Filter method

Currently, evolutionary algorithms such as Evolutionary Strategy $[ES]$, Genetic Algorithm (GA) and Binary Particle Swarm Optimization $(BPSO)$ algorithms have been introduced as the most advanced wrapper algorithms for the analysis of microarray datasets [9, 10, 11, 12]. Evolutionary algorithms select genes nonlinearly by generating gene subset randomly unlike conventional wrappers. In addition to this, evolutionary algorithms are efficient in exploring large searching space for solving combinatorial problems [13]. Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique, and was developed by Kennedy and Eberhart [14, 15].

PSO simulates the social behavior of organisms, such as bird flocking and fish schooling, to describe an automatically evolving system. In comparison with other stochastic optimization techniques like genetic algorithm (GAs), PSO have fewer complicated operations and fewer defining parameters, and can be coded in just

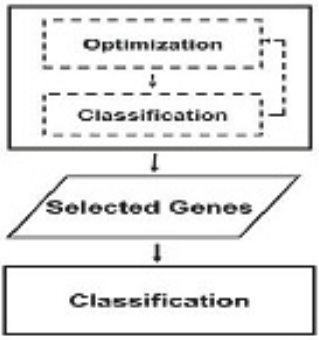


Fig. 2 Wrapper method

a few lines. Due to these advantages, the PSO has been successfully applied to many large scale problems in several engineering disciplines [17]. The third type of selection scheme is known as embedded method, which use the inductive learning algorithm as the feature selector as well as classifier. As illustrated in Figure 3, feature selection is actually a by-product of the classification process such as classification trees ID3 [18] and C4.5 [19]. But, the limitation of embedded methods is that they are generally greedy based [8], using only top ranked genes to perform sample classification in each step while an alternative split may perform better. Furthermore, additional steps are required to extract the selected genes from the embedded algorithms.

Each of the above techniques has their own limitations. To overcome the limitations of each technique while attempting to use their strengths, several intelligent hybrid techniques have been proposed. In [20], Yang et al. has mentioned that none of the filter algorithm is universally optimal and there is no tested method for selecting a filter for a particular dataset. They proposed a hybrid method which integrates various filters using a distance metric.

Computational intelligent techniques generally use a search mechanism to find the optimal solution for the problems. It differs from conventional computing techniques with respect to its tolerance to imprecision, vagueness, approximation, uncertainty and partial truth. However, all these techniques mimic the behavior and functions used by human beings and animals. Some of the intelligent techniques are neural network, evolutionary computation, particle swarm optimization and fuzzy computing. This chapter discusses some of the classification problems where the techniques mentioned above have been applied. All these techniques also can be used for feature extraction, feature selection and prototype selection.

Rough set has been used as a tool to detect the dependencies in the dataset and to reduce the dimension of the dataset using the data alone, requiring no additional information [21, 22, 23]. Over the years, Rough set has been accepted by many researchers and has been applied in many domains. In order to obtain most informative subset of genes or the reduct set from the original set, Rough set is

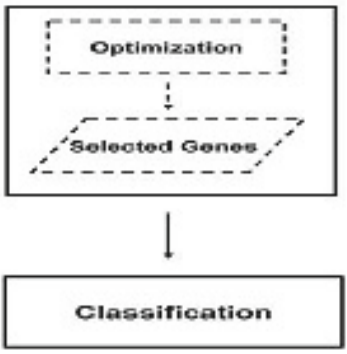


Fig. 3 Embedded method

applied on discretized dataset. All other attributes can be removed from the dataset with minimal information loss. A quick search of biological literatures shows that rough sets are still seldom used in bioinformatics. A major problem in using rough sets to deal with microarray gene expression data may be the huge amount of data and slow computational speed of rough sets algorithm. In the context of pattern classification, each label or class is considered as a rough set which contains some patterns in upper and lower approximation.

This chapter is organized in the following manner. Section 2 presents the topic on ”Redundant gene selection is using an intelligent hybrid PSO and Quick-Reduct algorithm [23],” with experimental results and discussion. Section 3 explains another application with experimental results based on ”Rough set aided hybrid gene selection for cancer classification [25]”. Section 4 describes the last application with experimental results and discussion which is based on ”Use of hybrid data mining technique (CFS+PLS) for improving classification accuracy of microarray data set [26]”. In the concluding section, conclusion of the chapter is followed by future scope and references.

2 Gene Selection Using Intelligent Hybrid PSO and Quick-Reduct Algorithm

Feature selection is an important technique to handle abundant noise, irrelevant features, imprecise and inconsistent information in real world problems. Rough sets [27, 28, 29] can address uncertainty and vagueness in the pattern which we find in inconsistent data. Rough sets have been successfully used as a feature selection method in pattern identification. It selects subsets of features, which predicts both the decision concepts and the original feature set. The optimal criterion of this feature selection technique is to select minimal reduct sets [30].

There are two types of searching operations generally used with rough set methods for feature selection, hill-climbing or greedy methods and stochastic methods [31]. The hill-climbing methods use to employ rough set attribute significance as heuristic

knowledge. They start with an empty set or attribute core followed by forward selection or backward elimination. Forward selection adds the most significant attribute or gene one at a time from the candidate set until the selected set is a reduct set. On the other hand, backward elimination starts with a full attribute set and removes attributes incrementally. Hill-climbing or heuristic methods are more efficient when deal with a little noise and a small number of interacting features, but are not assured of optimality. But Stochastic methods provide a more efficient solution at the expense of increased computational effort [31]. Therefore, stochastic feature selection method is the best option when optimal or minimal subset is required.

This chapter discuss a new feature selection mechanism by observing how particle swarm optimization (PSO) can be employed to select optimal feature subsets or rough set reducts [24]. PSO is a population based evolutionary computation technique developed by Kennedy and Eberhart [32] motivated from the simulation of social behavior of flock of birds. The objective was to simulate the graceful but unpredictable movement of birds. The PSO algorithm mimics the social behavior of flying birds and the way they exchange information to solve optimization problems. Each potential solution is seen as a particle with a certain velocity, and "flies" through the problem space. Each particle adjusts its flight according to its own flying experience and its companions' flying experience. The particle swarms find optimal regions of complex search spaces through the interaction of individuals in a population of particles. PSO has been successfully applied to a large number of complex combinatorial optimization problems; studies show that it often outperforms Genetic Algorithms [32]. PSO is a very efficient optimization tool for feature selection in which swarms behave like a particle and discover the best feature combinations as they fly within the problem space.

To find significant features with minimum redundancy, a novel integrated algorithm can be designed by combining a population based Particle Swarm Optimization (PSO) technique with multivariate filter algorithm i.e., correlation based feature selection (CFS) and supervised quick reduct algorithm $(CFS-PSO-QR)$ [24]. The fitness function of PSO explicitly measures the relevant features and feature redundancy simultaneously. $CFS-PSO$ determines the redundancy in feature set by applying the maximum feature inter-correlation measure, which is more reliable than the averaged inter-correlation. $CFS-PSO$ derives a compact feature set with high predictive ability due to the evolutionary PSO algorithm. Then supervised quick reduct (QR) algorithm has been used that enables $CFS-PSO$ to find minimal sets or reduct sets, which is a set of non-redundant features having the ability to discern between the objects. The efficiency of the proposed algorithm has been shown on two benchmark multi-class cancer datasets viz. Leukemia and Lung cancer. The approach which is applied in this work [24] is as follows:

$$Redu_{fi} = \max_{j=1:k,\, i\neq j} (su_{f_i f_j}) \tag{1}$$

where k is the size of selected feature set, and $su_{f_i f_j}$ is the correlation between f_i and f_j. Symmetrical uncertainty is applied to measure the correlation and then the merit $Merit_S$ of the set is defined as:

$$Merit_S = \frac{\sum_{i=1}^{k} su_{yf_i}}{\sum_{i=1}^{k} Redu_{f_i} + k + \delta} \tag{2}$$

where k is the size of selected feature set, δ is a parameter, su_{yf_i} is the correlation between feature f_i and the target y. The feature set having top score of $Merit_S$ is regarded as the best feature set and at the same time it has good predictive power and low redundancy. In order to obtain a compact gene set with high predictive power, an integrated algorithm i.e., correlation based feature selection by Particle Swarm Optimization $(CFS-PSO)$ method combined with supervised Quick-Reduct algorithm $(CFS-PSO-QR)$ is proposed. In $(CFS-PSO)$, binary PSO is chosen to search the optimal solution. Position of each particle which represent a gene, $x = (x_1, x_2, ..., x_m)$ provides a possible solution for feature selection, where m is the number of genes/features. If value of x is 1 then the corresponding gene is selected and vice versa. Most likely, PSO finds a global optimal gene set or a near global optimal gene set.

2.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique, and was developed by Kennedy and Eberhart in 1995 [14]. PSO simulates the social behavior of organisms, such as bird flocking and fish schooling, to describe an automatically evolving system. In comparison with other stochastic optimization techniques like genetic algorithm (GAs), PSO have fewer complicated operations and fewer defining parameters, and can be coded in just a few lines. Due to these advantages, the PSO has been successfully applied to many large scale problems in several engineering disciplines [33]. Given a design problem space D, PSO tries to find the global optimal solution by the movement of a swarm of particles. Each particle's position $x = (x_1, x_2, \cdots, x_m)$ is a possible solution, where m is the dimension of D. The movements of many particles, like bird flocking, have great search power. The particle updates its position x by the velocity $\upsilon = (\upsilon_1, \upsilon_2, \cdots, \upsilon_m)$ in a simple way:

$$X = X + \upsilon. \tag{3}$$

The moving path of particle x is decided by the variance of velocity υ, which is also calculated in a self-updating way. The value of velocity is revised by its particle's current optimal position $p = (p_1, p_2, \cdots, p_m)$ and the global optimal position of all particles $g = (g_1, g_2, \cdots, g_m)$. The velocity υ is updated as:

$$\upsilon_j = \omega\upsilon_j + c_1 rand()(p_j - x_j) + c_2 rand()(g_j - x_j). \tag{4}$$

where υ_j is the $j-th$ element of υ, $rand()$ is uniform random numbers between 0 and 1, c_1 and c_2 are two positive acceleration constants, usually $c_1 = c_2 = 2$, and ω is a inertia weight controlling the influence of a particles previous velocity and resulting in a memory effect. In different problems, PSO can also define some extra

constants, such as υ_{max}, υ_{min}, x_{max}, x_{min}, to restrain the position and velocity range of particles in the search space. For binary discrete search problems, Kennedy and Eberhart [15] proposed a binary PSO, where particles move in a state space restricted to 0 and 1 on each dimension. Different with normal PSO, in binary PSO, the velocity υ is used to control the probability of position x taking the value 0 and 1. For each value x_j of position x, where $j = 1 : m$, the updating equation is defined as:

$$s(\upsilon_j) = \frac{1}{1+\exp(-\upsilon_j)}$$

$$x_j = \begin{cases} 1, \ rand() < s(\upsilon_j); \\ 0, \ rand() \geq s(\upsilon_j). \end{cases} \tag{5}$$

where $s(\upsilon_j)$ is the probability of x_j taking the value 0 and 1 and $rand()$ is random number drawn from uniform sequence of $U(0,1)$.

2.2 Proposed Algorithm

The supervised $(CFS-PSO-QR)$ algorithm calculates the reduct set from the subset [34] obtained from $(CFS-PSO)$ filter. It deals with two parameters, conditional attribute and decision attribute and the evaluation of degree of dependency leads to the decision attribute. The algorithm starts with a null set and adds one at a time those attributes that results in increasing the rough set dependency metric until a maximum possible value for the dataset is obtained. The pseudo code of PSO is given here.

Algorithm PSO
Input:

m: swarm size;
c_1, c_2: positive acceleration constants;
W: inertia weight
Max-V: maximum velocity of particles
Max-Gen: maximum generation
Max-Fit: maximum fitness value

Output:
Pgbest: Global best position
Begin $Swarms\{x_{id}, \upsilon_{id}\} = Generate(m)$;
/* Initialize a population of particles with random positions and velocities on S dimensions*/

$Pbest(i) = 0;$
$i = 1, \cdots, m;$
$d = 1, \cdots, S;$
$Gbest = 0; Iter = 0;$

While (Iter<Max-Gen and Gbest< Max-Fit)
$\{for(every\ \ particle\ i)$
$\{Fitness(i) = Evaluate(i);$

$$IF(Fitness(i) > Pbest(i))$$

$\{Pbest(i) = Fitness(i); p_{id} = x_{id}\ ;\ d = 1, \cdots, S\}$
$IF(Fitness(i) > Gbest)$
$\{Gbest = Fitness(i); gbest = i;\}\}$
$For(every\ \ particle\ \ i)$
$\{For(every\ d)\{$
$\upsilon_{id} = w * \upsilon_{id} + c_1 * rand()(p_{id} - x_{id}) + c_2 * Rand() * (p_{id} - x_{id})$
$IF(\upsilon_{id} > Max - V)\{\upsilon_{id} = Max - V;\}$
$IF(\upsilon_{id} < -Max - V)\{\upsilon_{id} = -Max - V;\}$
$x_{id} = x_{id} + x_{id}$
}
}
$Iter = Iter + 1;$
}
/*rand () and Rand() are two random functions in the range $[0, 1]$*/ Return P_{gbest}
End.

The supervised CFS-PSO-quick reduct (CFS-PSO-QR) algorithm given in the following quick-reduct algorithm to calculate the reduct from the subset [34] generated from CFS-PSO filter. It has two parameters, conditional attribute and decision attribute and its evaluation of degree of dependency value leads to the decision attribute. It starts off with an empty set and adds in turn, one at a time, those attributes that result in the greatest increase in the rough set dependency metric, until this produces its maximum possible value for the dataset. According to the algorithm, the dependency of each attribute is calculated and the best candidate is chosen. The performance of supervised CFS-PSO-QR algorithm will be examined in our experiments.

The pseudo code of the supervised (CFS-PSO-QR) algorithm is given below.

Algorithm Supervised Quick-Reduct (CFS-PSO-QR)
Quick Reduct ((CFS-PSO)S, D)
$(CFS - PSO)_S = \{g_1, g_2, \cdots, g_k\};$
/* the set of all conditional features represent gene.*/
$D = \{d\};$
/* the set of decision features corresponds to class label of each sample */
$g_i \in (CFS - PSO)_S;$
$g_i = \{x_{1,i}, x_{2,i}, \cdots, x_{m,i}\}$
/* $i = 1, 2, \cdots$, n , where $x_{k,i}$ is the expression level of gene i at sample $k, k = 1, 2, \cdots, m$ */
$a.R \leftarrow \{\};$
$b.do$

$c.T \leftarrow R$;
$d.forg_i \in ((CFS-PSO)_s - R)$
$e.If \gamma R \bigsqcup \{x_i\}(D) > \gamma T(D)$
$f.T \leftarrow R \bigsqcup \{x_i\}$;
$g.R \leftarrow T$;
$h.until\ \gamma R(D) == \gamma(CFS-PSO)_s(D)$; $i.\ return\ R$.

2.3 Implementation and Results

Two benchmark multi-class cancer microarray data sets: Leukemia cancer data set and lung cancer data set, which are used by many researchers for gene selection and cancer classification, are used to measure the efficiency of the proposed method which is taken from http://www.gems-system.org.

The Leukemia training data set consists of 43 samples, 3 classes and 11226 genes, including 14 ALL, 12 MLL and 17 AML type of Leukemia data. The test data set consists of 29 samples, 3 classes and 11226 genes, including 11 ALL, 10 AML and 8 MLL type of Leukemia data. The information about number of samples, number of classes and number of attributes is given in Table 1. The samples are taken from 63 bone marrow samples and 9 peripheral blood samples.

The Lung cancer data set consists of 203 samples, 5 classes and 12601 genes including 139 AD, 17 NL, 6 SMCL, 21 SQ and 20 COID. Out of which 122 samples constitute training data set and 81 samples for test data set shown in Table 1. The training and test data sets contain 83/56 AD, 10/7 NL, 4/2SMCL, 13/8 SQ and 12/8 COID respectively.

After applying integrated filter algorithm (CFS+PSO) in training dataset, 12 genes are selected from Leukemia data set and 1210 genes from Lung cancer data set. In the next step, the filtered data sets obtained from previous step undergo through rough sets attribute reduct technique. Only 2 genes from Leukemia and 17 genes from Lung cancer dataset are selected from second stage of reduction which is shown in Table 1. The classification accuracy obtained from the combined feature selection method and 3 different classifiers are given in Table 2, 3, 4 and Table 5.

The integrated gene/feature search algorithm has increased the classification accuracy of both the datasets over the accuracy reported by [35]. The size of selected gene sets is also an important issue of gene selection. Comparing the genes selected in both the selection techniques i.e., from (CFS-PSO) and (CFS-PSO-QR), which is given in Table 1, it is understood that (CFS-PSO-QR) computes the most compact gene set. By combining the evaluation feedbacks of multiple filtering algorithms the system does not simply improves classification accuracy of training dataset greedily, but considers other characteristics of the data as well. The over fitting problem can then be replaced by a better generalization of the identified gene and gene subsets. By measuring the characteristics of each candidate gene from different aspects, we can reduce the possibility of identifying false positive gene while finding more compact gene subset. This gene subset can be useful for validating the proposed hybrid learning model.

Table 1 Cancer dataset information and number of genes selected in two reduction steps

Dataset	# classes	# samples	# genes	CFS+ PSO	CFS + PSO+ Rough
Leukemia (Train/ Test)	3	43/29	11226	12	2
Lung (Train/Test)	5	122/81	12601	1210	17

Table 2 Classification accuracy for leukemia training and test dataset using reduced subset obtained from (CFS+PSO)

	Classification accuracy for each class				
Classifier	ALL	MLL	AML	Overall classification accuracy	Overall classification accuracy for test data
5-knn	85.71	66.667	94.118	83.7209	82.7586
Naive Bayes	92.857	66.667	88.235	83.7209	86.2069
J48	71.429	16.667	76.471	58.1395	72.4138

Table 3 Classification accuracy for leukemia training and test dataset using reduced subset obtained from (CFS+PSO+QR)

	Classification accuracy for each class				
Classifier	ALL	MLL	AML	Overall classification accuracy	Overall classification accuracy for test data
5-knn	71.429	75.00	88.235	79.0698	86.2069
Naive Bayes	50.00	83.333	88.235	74.4186	79.3103
J48	91.667	50.00	82.353	72.093	82.7586

Table 4 Classification accuracy for lung cancer training and test dataset using reduced feature obtained from (CFS+PSO)

	Classification accuracy for each class				
Classifier	ALL	MLL	AML	Overall classification accuracy	Overall classification accuracy for test data
5-knn	71.429	75.00	88.235	79.0698	86.2069
Naive Bayes	50.00	83.333	88.235	74.4186	79.3103
J48	91.667	50.00	82.353	72.093	82.7586

Table 5 Classification accuracy for lung cancer training and test dataset using reduced feature obtained from (CFS + PSO + QR)

Classifier	Classification accuracy for each class					Overall classification accuracy	Overall classification accuracy for test data
	AD	NL	SMCL	Q	COID		
5-knn	95.818	70.00	50.00	76.923	58.333	86.0656	83.9506
Naive Bayes	93.976	80.00	50.00	84.615	83.33	89.3443	82.716
J48	89.157	80.00	0	53.846	75.00	80.3279	81.4815

3 Rough Set Aided Hybrid Gene Selection for Cancer Classification

In this application, again, a hybrid Supervised Correlation based-Quick Reduct algorithm is proposed for attribute/ gene reduction from gene expression dataset. Correlation based Feature subset selection (CFS) is used as a filter method to remove redundant genes/attributes by evaluating the worth of a subset of genes/attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them [26]. In this case, the integrated algorithm is used to find the minimal reduct sets from microarray datasets. The predictive ability of the reduced dataset is evaluated by three classifiers. The proposed method improves the efficiency and decreases the complexity of the algorithm. However, the integrated Rough set based feature selection algorithm is applied on two public multi-class gene expression datasets and the experimental results show that this hybrid intelligent method is successful for selecting high discriminative genes for classification task.

3.1 Rough Set

The concept of Rough sets, proposed by Zdzislaw Pawlak [21], considers vagueness from a different point of view. In rough set theory, vagueness is not described by means of set membership but in terms of boundary regions of a set of objects. If the boundary region is empty, then the set is a crisp, otherwise it is rough or inexact. The existence of a non-empty boundary region implies our lack of sufficient knowledge to define the set precisely, with certainty. Let $A = (A_1, A_2, \cdots, A_m)$ be a non-empty finite set of attributes and $U = (a_1, a_2, \cdots, a_m)$ be a non-empty finite set of m-tuples, known as the objects of universe of discourse. $V(a_i)$ denote the set of all values for the attributes a_i. Then an information system is defined as an ordered pair $I = (U, AU\{d\})$ where d is the decision attribute such that for all $i = 1, 2, \cdots, m$, there is a function f_i

$$f_i : U \longrightarrow V(a_i) \tag{6}$$

If $P \subseteq A$, then the associated equivalence relation is:

$$IND(P) = \{(x,y) \in UXU \mid \forall a \in P,\ f_a(x) = f_a(y)\} \tag{7}$$

Then the partition of universal set U generated by $IND(P)$ is denoted as U/P. If $(x,y) \in IND(P)$, then x and y are indiscernible by the attributes from P. The equivalence classes of the P-indiscernibility relation are denoted by $[x]p$. Let $X \subseteq U$, the P-lower approximation $\underline{PX}$ and P-upper approximation $\overline{PX}$ of set X can be defined as:

$$\underline{PX} = \{x \in U \mid [x]p \subseteq X\} \tag{8}$$

$$\overline{PX} = \{x \in U \mid [x]p \bigcap X \subseteq \neq \phi\} \tag{9}$$

Let P, $Q \subseteq A$ be the equivalence relations over U, then the positive, negative and boundary regions of fuzzy set can be defined as:

$$POS_p(Q) = \bigcup_{x \in U/Q} \underline{PX} \tag{10}$$

$$NEG_p(Q) = U - \bigcup_{X \in U/Q} \overline{PX} \tag{11}$$

$$BND_p(Q) = \bigcup_{X \in U/Q} PX - \bigcup_{X \in U/Q} PX \tag{12}$$

The positive region of the partition U/Q with respect to P is denoted as,$POS_P(Q)$ is the set of all objects of U. The partition Q depends on P in a degree k $(0 \leq k \leq 1)$ denoted by $P \Rightarrow_k Q$ such that

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|U|} \tag{13}$$

where P is a set of condition attributes, Q is the decision attribute and $\gamma_P(Q)$ is the quality of classification. If the value of dependency parameter $k = 1$ then Q depends totally on P; if $0 < k < 1$, Q depends partially on P; and if $k = 0$ then Q does not depend on P. The goal of reduction of attributes is to eliminate redundant features and obtain a reduct sets which will provide quality classification as the original one. The reduct sets are defined as:

$$Red(C) = \{R \subseteq C \mid \gamma_R(D) = \gamma_C(D),\ \forall B \subset R,\ \gamma_B(D) \neq \gamma_C(D)\} \tag{14}$$

3.2 Gene Selection Based on Rough Set Method

The objective of the method is to identify marker genes from gene expression dataset. Rough set theory formalizes this problem as decision system $T = (U,A,D)$, where the universe of discourse $U = (a_1,a_2,\cdots,a_m)$ is a set of samples, the conditional attributes $A = (A1,A2,,\cdots,A)$ is a set of genes and the decision attributes

$D = \{d\}$ represents the class label of each sample. Many genes in the dataset are highly correlated and this redundancy increases computational cost and decreases classification accuracy. Thus correlation based feature selection (CFS) is applied to reduce the dimension of gene space. CFS evaluates a subset of features by considering the individual predictive ability of each feature along with degree of redundancy between them [34]. Therefore CFS can be stated as:

$$CFS_S = \frac{\overline{kr}_{cf}}{\sqrt{k+k(k-1)}r_{ff}} \tag{15}$$

where CFS_S is the score of a subset of features S which contains k features, $\overline{r}_{cf}$ is the average of the correlation between feature to class $(f \in S)$, and $\overline{r}_{ff}$ is the average of the correlation between feature to feature. The difference between univariate filter algorithms and CFS are that univariate evaluate scores for each feature independently whereas CFS finds a heuristic merit of the feature subsets and reports the best subset of features it evaluates. The best subset obtained from the filter process is then used to construct the reduct set by calculating the degree of

$$IND(P) = \{(x,y) \in U \times U \mid \forall a \in P,\ f_a(x) = f_a(y)\} \tag{16}$$

dependency which is a basis to find decision attributes. To achieve the goal, supervised CFS-quick reduct (CFS-QR) algorithm is proposed to calculate the reduct from the subset S generated [33], [36] from CFS filter which is a multi variate one.

3.3 Supervised Correlation Based Reduct Algorithm (CFS-RST)

The supervised (CFS-RST) algorithm.

Quick Reduct (CFS_S, D) $CFS_S = g_1, g_2, \cdots, g_k$, the set of all conditional features or genes. $D = \{d\}$, the set of decision attributes corresponds to class label of each sample. $a.R \leftarrow \{\}$;
$b.do$
$c.T \leftarrow R$;
$d.for g_i \in ((CFS)_s - R)$
$e.If \gamma R \bigsqcup \{g_i\}(D) > \gamma T(D)$
$f.T \leftarrow R \bigsqcup \{g_i\}$;
$g.R \leftarrow T$;
$h.until\ \gamma R(D) == \gamma(CFS)_(D)$; $i.\ return\ R$.

3.4 Implementation and Results

The same two benchmark multi-class cancer microarray data sets: Leukemia cancer data set and lung cancer data set which were used in first application are used in this application to measure the efficiency of the proposed modified Quick-Reduct algorithm. Initially, a simple preprocessing method [13] is applied to discretize the domain of each attribute which is a pre-requisite of Rough set method. Gene

expression values greater than $\mu + \sigma/2$ are transformed to state 1, those who lies between $\mu - \sigma/2$ and $\mu + \sigma/2$ are transformed to state 0 and values smaller than $\mu - \sigma/2$ are transformed to state -1. These three states correspond to the over-expression, baseline, and under-expression. Then proposed method is employed for searching the informative genes for classification purpose.

The modified Quick-Reduct feature selection algorithm reduces the attributes of both the datasets to a large extent. Leukemia and Lung datasets are left with only 6 and 13 genes respectively which is shown in Table 6.

The reduct set is then classified using three different classifiers such as: KNN, J48 and Naive Bayes. The classification performance of training data is validated with test data. The experimental results for both the datasets is shown in Tables 7, 8, 9 and 10 with and without applying the proposed reduction algorithm.

Therefore, it is evident from the experimental results that Rough set effectively identifies the most significant bio-markers whose characteristics represent the original dataset.

Table 6 Number of genes selected in training data set using CFS + Best First and CFS + Rough

Dataset	CFS + Best First	CFS + Rough Set
Leukemia	150	6
Lung	473	13

Table 7 Experimental results of leukemia training and test data with feature reduction

	Classification accuracy for each class				
Classifier	ALL	MLL	AML	Overall classification accuracy	Overall classification accuracy for test data
5-knn	85.71	58.3	100	83.7209	75.8621
Naive Bayes	86.71	75	100	88.3721	79.3103
J48	78.571	58.3	94.1	79.0698	79.0698

Table 8 Experimental results of entire leukemia dataset

	Classification accuracy for each class			
Classifier	ALL	MLL	AML	Overall classification accuracy
5-knn	79.16	92.85	85	86.11
Naive Bayes	95.83	71.42	95	86.11
J48	100	82.143	90	90.28

Table 9 Experimental results of lung cancer training and test data with feature reduction

Classifier	Classification accuracy for each class					Overall classification accuracy	Overall classification accuracy for test data
	AD	NL	SMCL	SQ	COID		
5-knn	97.59	80.00	75.00	61.53	95	90.9836	77.778
Naive Bayes	96.38	80.00	50.00	76.92	83.33	90.1639	82.716
J48	90.36	50.00	50.00	61.53	83.33	81.9672	79.0123

Table 10 Experimental results obtained from entire lung dataset

Classifier	Classification accuracy for each class					Overall classification accuracy
	AD	SQ	NL	SMCL	COID	
5-knn	99.27	76.19	76.471	0	90	90.64
Naive Bayes	78.417	85.714	76.471	33.333	95	79.3103
J48	99.281	76.19	70.84	0	90	92.6108

4 Hybrid Data Mining Technique (CFS + PLS) for Improving Classification Accuracy of Microarray Data

A major task in biomedical problems in recent years has been the categorization of gene expression samples using classification tools, such as cases and controls. This is achieved by training a classifier using a labeled training set taken from the two populations, and then that classifier is being used to predict the labels of new samples. Such prediction method has improved the diagnosis and treatment of several diseases. Supervised learning offers an effective means to differentiate positive from negative samples: a collection of samples with known type labels is used to train a classifier that is then used to classify new samples. Microarrays allow simultaneous measurement of tens of thousands of gene expression levels per sample. Because typical microarray studies usually contain less than one hundred samples, the number of features (genes) in the data far exceeds the number of samples. This asymmetry of the data poses a serious challenge for standard learning algorithms-that can be overcome by selecting a subset of the features and using only them in the classification. This feature selection step offers several advantages such as improved performance of classification algorithms, improved generalization ability of the classifier to avoid over-fitting, fewer features, making classifiers more efficient in time and space and more focused analysis of the relationship between a modest number of genes and the disease in question. In principle, many dimensionality reduction algorithms for supervised learning can be applied to the classification of gene expression data. Various hybrid schemes have been presented and all of them reported improved classification accuracy. There is no conclusion from previous studies so far which confirms superiority of any particular scheme for microarray data classification.

In this application we have developed a novel feature selection technique based on the Partial Least Squares (PLS) algorithm [37, 38, 39, 40]. PLS aims to obtain a low dimensional approximation of a matrix that is 'as close as possible' to a given vector. SIMPLS is a multivariate feature selection method based on PLS that incorporates feature dependencies [41]. In the first step, we implemented two different dimensionality reduction schemes: (i) SIMPLS as the dimensionality reduction algorithm and (ii) an alternative and novel hybrid feature selection scheme which consecutively applied correlation based feature selector method [40] on the original data sets followed by the SIMPLS regression algorithm. Then in the second step, the two sets of filtered data with new features resulting from the two feature selection schemes described in the first step were separately fed into four supervised classification algorithms namely, Support Vector Machine using Polynomial kernel function, Support Vector Machine using RBF kernel function, Multilayer Perceptron and Radial Basis Function Network (RBFN). Three different expression profile datasets comprising a total of 215 samples were collected and used for training and testing. We then used these two schemes our results show that the use of some SIMPLS variants leads to significantly better classification than that obtained with standard filters.

A similar procedure was employed in [41] in order to combine information from two different datasets of gene expression. Quite recently, Cao et al. [42] used PLS-SVD (a variant of PLS that uses singular value decomposition) together with Lasso Penalty in order to integrate data coming from different sources for classification. The combination of PLS and linear regression techniques was further studied in [43] . Fort and Lambert-Lacroix [44] described a classification using PLS with penalized logistic regression; like [45] , this study ran the t-test filter before applying PLS. All the above studies used PLS for classification, and when feature selection was involved, it was implicitly used. For example, in [46], where a penalizing process was applied to reduce the number of genes, the threshold parameter λ, which implicitly determines the number of features, was found using cross validation. The SIMPLS method is unique in that it focuses solely on feature selection; it does not propose a new classification procedure. As a result, it can be used as a pre-processing stage with different classifiers. Thus, we evaluated the performance of SIMPLS with different classifiers, and compared it with a hybrid feature selector method and not to the PLS-based classification methods mentioned above.

4.1 SIMPLS and Dimension Reduction in the Classification Framework

PLS regression is especially appropriated to predict a univariate or multivariate continuous response using a large number of continuous predictors. Suppose we have a $n \times p$ data matrix X. The centered data matrix XC is obtained by centering each column to zero mean. Y denotes a univariate continuous response variable and Y the $n \times 1$ vector containing the realizations of Y for the n observations. The centered

vector YC is obtained by subtracting the empirical mean of Y from Y. From now on, Y denotes a categorical variable taking values 1 to K, with $k \geq 2$. $Y_1, \cdots, Y_n$ denote the n realizations of Y. In this framework, PLS can be seen as a dimension reduction method: $t_1, \cdots, t_n \in R_n$ represent the observed m new components. Although the algorithm with orthogonal components has been designed for continuous responses, it is known to lead to good classification accuracy when it is applied to a binary response (K = 2), especially for high-dimensional data as microarray data [41], [42].The same can be said for the SIMPLS algorithm: a binary response can be treated as a continuous response, since no distributional assumption is necessary to use the SIMPLS algorithm. If the response is multi-categorical $(K > 2)$, it cannot be treated as a continuous variable. The problem can be circumvented by dummy coding. The multi-categorical random variable Y is transformed into a K-dimensional random vector $y \in 0, 1k$ as follows.

$$y_{i1} = 1 \ \ if Y_i = k, \qquad (17)$$
$$y_{ik} = 0 \ \ else,$$

Y denotes the $n \times K$ matrix containing y_i in its $i-th$ row, for $i = 1, \cdots, n$. In the following, Y denotes the $n \times 1$ vector $Y = (Y_1, \cdot, Y_n)T$, if Y is binary $(K = 2)$ or the $n \times K$ matrix as defined above if Y is multi-categorical $(K > 2)$. In both cases, the SIMPLS algorithm outputs a p x m transformation matrix A containing the $a_1, \cdots, a_m \in RP$ in its columns. The $n \times m$ matrix T containing the values of the new components for the n observations is computed as

$$T = X_C A. \qquad (18)$$

These new components can be used as predictors for classification. In this paper, we attempt to improve predictive accuracy by building a hybrid classification scheme for microarray data sets. In the first step, we implement SIM-Partial Least-Squares (SIMPLS) regression [42, 44] as the dimensionality reduction algorithm, on the original data sets. Then in the second step, the filtered data with new features resulting from the feature reduction scheme described in the first step is fed into supervised classification algorithms such as Polynomial Support Vector Machine (SVM) [20], radial SVM [20], Multilayer Perceptron [20] and Radial Basis Function Network (RBFN) [20] to compare the results of the classifiers.

4.2 Partial Least Squares Regression

Partial least squares (PLS) regression aims to reduce the data dimensionality with a similar motivation, but differs from PCA by adopting a different objective function to obtain PLS components. Whereas PCA maximizes the variance of each coordinate and whereas both PCA and latent factor analysis will not take into account the values of the target (dependent) attribute, the PLS regression model attempts to find a small number of linear combinations of the original independent variables which

maximize the covariance between the dependent variable and the PLS components. (PLS uses the entire data set: input and target attributes.) So the ith PLS component is given by

$$\omega_1 = arg \max_{\omega^T \omega=1} cov\{\omega^T x, y\}, \tag{19}$$

subject to constraint

$$t_i^T t_j = 0, \ where\ i \neq j\ , t_k = \omega_k^T x. \tag{20}$$

The PLS method can be illustrated by examining the following relations. Assuming X is an $n \times m$ matrix representing a data set of n instances with p independent variables, then if the number of PLS components is K, then the matrix X can be written as the summation of K matrices generated by outer products between vector ti (which is often known as the score vector) and pi T (which is often called the load vector). The optimal number of PLS components, K, is usually determined by applying cross-validation methods on training data.

$$X = TP^T + E = \sum_{i=1}^{K} t_i p_i^T + E \tag{21}$$

In effect, the relation in the PLS model projects the data vectors X from the original p-dimensional space into a (much lower than p) K-dimensional space. In the same way, when PLS components are used in the regression, the relation between dependent variable y and PLS component ti can be written as

$$Y = TBQ + F \tag{22}$$

Where T is PLS components matrix, B is the coefficients vector so that TB is orthogonal, Q is the regression coefficients matrix, F is the residual matrix and $|F|$ is to be minimized. Partial least squares regression can be regarded as an extension of the multiple linear regression model. It has the advantage of being more robust, and therefore it provides a good alternative to the traditional multiple linear regression and principal component methods. The original PLS method was proposed by Wold [47] in the late 1960s and initially applied in the field of econometrics. Since then the method had been adopted in other research disciplines and been widely applied in many scientific analyses. SIMPLS is an algorithm for partial least squares regression proposed by de Jong [48]. Compared to conventional nonlinear iterative partial least squares (NIPALS)-PLS, SIMPLS runs faster and is easier to interpret. In SIMPLS, the PLS components are calculated directly as linear combinations of the original variables, which avoids the construction of deflated data matrices. In this paper, we use the SIMPLS algorithm by de Jong [48], which can be seen as a generalization for multi-categorical response variables of the algorithm.

4.3 Implementation and Results

In this study, the dimensionality reduction scheme is implemented as follows. Each column of the training set is normalized, so that each column has a mean of zero and variance of one. The values of the binary target attribute are set to either 0 or 1. Specifying the number of components for the Partial Least Square Regression, then a PLS model for a training data set is built by feeding the original training set into the SIMPLS algorithm. The output scores of the PLS algorithm are regarded as the values of input variables and forms the training set for the classification algorithms.

Determining the Optimal Number of PLS Components: Biologists often want statisticians to answer questions like 'which genes can be used for tumor diagnosis'? Thus, gene selection remains an important issue and should not be neglected. Dimension reduction is sometimes wrongly described as a black box which looses the information about single genes. In the following, we will see that PLS performs gene selection intrinsically. In this section, only binary responses are considered: Y can take values 1 and 2. We denote as $YC = (YC_1, \cdots, YC_n)T$ the vector obtained by centering $Y = (Y_1, \cdots, Y_n)T$ to zero mean:

$$Y_{Ci} = \begin{cases} -n_2/n, \text{ if } Y_i = 1; \\ n_1/n, \quad \text{if } Y_i = 2. \end{cases}$$

where $n_1 \quad n_2$ are the numbers of observations.

To perform PLS dimension reduction, it is not necessary to scale each column of the data matrix X to unit variance. However, the first PLS component satisfies an interesting property with respect to gene selection if X is scaled. In this section, the columns of the data matrix X are supposed to be have been scaled to unit variance and, as usual in the PLS framework, centered to zero mean. $a = (a_1, \cdots, a_p)T$, denotes the $p \times 1$ vector defining the first PLS component as calculated by the SIMPLS algorithm.

A classical gene selection scheme consists of ordering the p genes according to BSS_j/WSS_j and selecting the top-ranking genes. For data sets with binary responses, we argue that aj^2 can also be seen as a scoring criterion for gene j and we prove that the ordering of the genes obtained using $BSS_j/WSS_j$is the same as the ordering obtained using aj^2. As a consequence, the first PLS component calculated by the SIMPLS algorithm can be used to order and select genes and the ordering is the same as the ordering produced by one of the most widely accepted selection criteria. Up to a constant, the BSS / WSS-statistic equals the F-statistic which is used to test the equality of the means within different groups. Since BSS / WSS is obtained by a strictly monotonic transformation of aj^2, aj^2, can be seen as a test statistic itself. This PLS-based procedure for gene selection is much faster than the computation of BSS / WSS for each gene.

Three widely used microarray gene expression datasets are chosen for our experiments: ALL-AML, lung cancer, and colon tumor. The data is taken from http://sdmc.lit.org.sg/GEDatasets/Datasets.html. Table 11 summarizes these datasets. We conducted the experiments on these three data sets by applying Partial Least

Square (PLS) method for feature reduction and Polynomial Support Vector Machine (SVM) , Radial SVM , Multilayer Perceptron and Radial Basis Function Network(RBFN) for classification of the reduced datasets. We evaluated the performance of feature reduction with four classifiers, using 10-fold Cross Validation. We performed 10-fold Cross Validation on both the feature reduction process and the classification step.

The dimensionality reduction scheme is implemented as follows. Each column of the training set is normalized, so that each column has a mean of zero and variance of one. The values of the binary target attribute are set to either 0 or 1. Specifying the number of components for the Partial Least Square Regression, then a PLS model for a training data set is built by feeding the original training set into the SIMPLS algorithm. The output scores of the PLS algorithm are regarded as the values of input variables and forms the training set for the classification algorithms. Table12 shows optimal number of components selected by SIMPLS algorithm.

In two-stage dimensionality reduction scheme, irrelevant genes were filtered out by correlation based feature selector method(CFS) [32] in the first step and in the second step, dimension of the data is further reduced by applying SIMPLS, a variant of PLS method. We processed the data using the above scheme, then applied the learning algorithms. These experimental results showed that, in, going from the SIMPLS scheme in Table 13 to the hybrid scheme in Table 14, only a marginal increase in classification accuracy of Lung cancer data set has been obtained. SIMPLS a variant of PLS is a supervised procedure which uses the information about the class of the observations to construct the new components. Unlike sufficient dimension reduction, PLS can handle all the genes simultaneously and performs gene selection intrinsically. In other word, PLS is a very fast and competitive tool for classification problems with high dimensional microarray data as regards to prediction accuracy.

Table 11 Microarray datasets

Dataset	# of genes	# of instances	# of positive samples	#of negative samples
Leukemia	7129	72	47(ALL)	25(AML)
colon cancer	2000	62	22	40
Lung cancer	12533	181	31(MPM)	150(ADCA)

Table 12 The optimal number of PLS components

Dataset	RBFN	Polynomial SVM	RBF SVM	MLP
Leukemia	04	50	8	20
colon cancer	8	40	20	40
Lung cancer	20	50	50	50

Table 13 Predictive error (%) of classification algorithms, using SIMPLS dimensionality reduction scheme

Dataset	RBFN	Polynomial SVM	RBF SVM	MLP
Leukemia	0	0.45	28.22	0.41
colon cancer	10.95	0	23.33	0.31
Lung cancer	11.55	0	16	0.95

Table 14 Predictive error (%) of classification algorithms, using a hybrid dimensionality reduction scheme

Dataset	RBFN	Polynomial SVM	RBF SVM	MLP
Leukemia	2.86	3.88	31.11	4.75
colon cancer	32.46	17.13	33.89	22.53
Lung cancer	8.65	1.91	10.95	0.75

5 Conclusion

A number of hybrid intelligent feature selection applications like (CFS+PSO+QR), (CFS+QR) and SIMPLS have been discussed in this chapter. This chapter has also discussed the shortcomings of conventional hill-climbing rough set approaches to feature selection. Often these heuristic techniques fail to find optimal reductions. On the other hand, complete searches are not feasible for even medium-sized datasets. So, stochastic approaches provide a promising feature selection mechanism. The algorithms discussed provide robust optimization techniques which can be applied in various research fields. These methods are capable of dealing with ill-defined objective functions and constraints of the problem. The evaluation of the degree of dependency in Rough set and individual particles in PSO are important for the working of the algorithm which also require understanding of the domain knowledge. Other evolutionary algorithms like GA, Ant-colony optimization, simulated annealing and Tabu search can be used for finding optimal solutions.

An efficient feature reduction method, SIMPLS a variant of PLS has been used for reducing the dimension of the datasets. This supervised procedure only requires the information about the class of the observations to construct the new components. Unlike sufficient dimension reduction, PLS can handle all the genes simultaneously and performs gene selection intrinsically. In other word, PLS is a very fast and competitive tool for classification problems with high dimensional microarray data as regards to prediction accuracy.

All the methodology discussed above is tested with microarray dataset. These gene expression data sets have very unique characteristics like high dimensionality but low sample size which is very different from all the previous data used for classification. Only a very small fraction of genes or subsets of genes are informative for a certain task.

6 Scope for Future Work

This chapter has discussed how hybrid intelligent techniques can be used in the field of feature reduction and classification. Only a few representative techniques have been considered for finding optimal subsets of genes which can predict the disease. There is immense scope to use intelligent techniques for a number of applications. Combined with the Partial Least-Squares (PLS) regression method, which is proved to be an appropriate feature selection method, the learning algorithms are capable of building classification models with high predictive accuracies from microarray data. As the study shows that our feature reduction scheme improves classification accuracies, one question immediately arises: will there be better hybrid schemes for the feature selection process for building supervised classification models? Since the number of instances in the studied microarray data is small and the performances of many classification algorithms are sensitive to the number of training data, another interesting question is raised: when comparing predictive performances of various classification algorithms on microarray data, what is the impact of adopting different methodologies such as tenfold cross-validation, leave-one-out cross-validation and bootstrap. Particle Swarm Optimization (PSO) has the ability to quickly converge and also a promising method for rough set reduction. It can be further improved with other stochastic search techniques such as ant colony optimization, cat swarm optimization and bee colony optimization. It is also possible to use integrated fuzzy-rough technique to carry out learning.

References

1. Saeys, Y., Lnza, I., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23 (19), 2507–2517 (2007)
2. Somorjai, R.L., Dolenko, B., Baumgartner, R., Crow, J.E., Moore, J.H.: Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. Bioinformatics 19, 1484–1491 (2003)
3. Wang, Y., Makedon, F., Ford, J., Pearlman, J.: Hykgene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. Bioinformatics 21, 1530–1537 (2005)
4. Jafari, P., Azuaje, F.: An assessment of recently published gene expression data analyses: reporting experimental design and statistical factors. BMC Med Inform Decis Mak. 6(27) (2006)
5. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of relieff and rrelieff. Machine Learning 53, 23–69 (2003)
6. Su, Y., Murali, T., Pavlovic, V., Schaffer, M., Kasif, S.: Rankgene: identification of diagnostic genes based on expression data. Bioinformatics 19, 1578–1579 (2003)
7. Kohavi, R., John, G.: Wrapper for feature subset selection. Artificial Intelligence 97, 273–324 (1997)
8. Blum, A., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence 97, 245–271 (1997)
9. Li, L., Weinberg, C., Darden, T., Pedersen, L.: Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 17, 1131–1142 (2001)

10. Ooi, C., Tan, P.: Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. Bioinformatics 19, 37–44 (2003)
11. Jirapech-Umpai, T., Aitken, S.: Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. BMC Bioinformatics 6(146) (2005)
12. Liu, J., Cutler, G., Li, W., Pan, Z., Peng, S., Hoey, T., Chen, L., Ling, X.: Multiclass cancer classification and biomarker discovery using GA-based algorithm. Bioinformatics 21, 2691–2697 (2005)
13. Kudo, M., Sklansky, J.: Comparison of algorithms that select features for pattern classifiers. Pattern Recognition 33, 25–41 (2000)
14. Kennedy, J., Eberhart, R.C.: Prticle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, Australia, vol. 4, pp. 1942–1948 (1995)
15. Kennedy, J., Eberhart, R.: A discrete binary version of the particle swarm algorithm. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, Orlando, FL, USA, vol. 5, pp. 4104–4108 (1997)
16. Juan, C.: A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Transactions on Systems, Man and Cybernetics 34, 997–1006 (2004)
17. Deng, X.: Research on building crowd evacuation model based on multi-agent particle swarm optimization algorithm. Journal of Convergence Information Technology 8(4), 17–25 (2013)
18. Quinlan, J.R.: Induction of decision trees. Machine Learning 1, 81–106 (2004)
19. Quinlan, J.R.: Programs for machine learning. Morgan Kaufmann, CA (1993)
20. Yang, Y.H., Xiao, Y., Segal, M.R.: Identifying differentially expressed genes from microarray experiments via statistic synthesis. Bioinformatics 21(7), 1084–1093 (2005)
21. Pawlak, Z.: Rough set approach to knowledge-based decision support. European Journal of Operational Research 99, 48–57 (1997)
22. Mitra, S., Hayashi, Y.: Bioinformatics with Soft Computing. IEEE Transactions on Systems, Man and Cybernetics 36(5), 616–635 (2006)
23. Grzymala-Busse, J.W.: LERS-a system for learning from examples based on rough sets. In: Slowinski, R. (ed.) Intelligent Decision Support, pp. 3–18. Kluwer Academic Publishers, Dordrecht (1992)
24. Dash, S., Patra, B.: Redundant gene selection based on genetic and quick-reduct algorithm. International Journal on Data Mining and Intelligent Information Technology Applications 3(2) (2013)
25. Dash, S., Patra, B., Ttripathy, B.K.: A hybrid data mining technique for improving the classification accuracy of microarray data set. International Journal of Information Engineering and Electronic Business 2, 43–50 (2012)
26. Dash, S., Patra, B.: Rough set aided gene selection for cancer classification. In: Proceedings of 7th International Conference on Computer Sciences and Convergence Information Technology. IEEE Xplore, Seoul (2012)
27. Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
28. Pawlak, Z.: Rough sets: Theoretical aspects of reasoning about data. Kluwer Academic Publishing, Dordrecht (1991)
29. Pawlak, Z.: Rough set approach to knowledge-based decision support. European Journal of Operational Research 99, 48–57 (1997)
30. Swiniarski, R.W., Skowron, A.: Rough set methods in feature selection and recognition. Pattern Recognition Letters 24(6), 833–849 (2003)

31. Vafaie, H., Imam, I.F.: Feature selection methods: genetic algorithms vs. greedy-like search. In: Proceedings of International Conference on Fuzzy and Intelligent Control Systems (1994)
32. Kennedy, J., Spears, W.M.: Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. In: Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 39–43 (1998)
33. Juan, C.: A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Transactions on Systems, Man and Cybernetics 34, 997–1006 (2004)
34. Jensen, R., Shen, Q.: Semantics-preserving dimensionality reduction: rough and fuzzy-rough based approaches. IEEE Transactions on Knowledge and Data Engineering 16 (12), 1457–1471 (2004)
35. Hall, M.A.: Correlation-based feature selection for machine learning. Ph.D. Thesis. Department of Computer Science, University of Waikato (1999)
36. Jensen, R.: Combining rough and fuzzy sets for feature selection. Ph.D. Dissertation. School of Informatics, University of Edinburgh (2004)
37. Ding, H., Peng, C.: Minimum redundancy feature selection from microarray gene expression data. Journal of Bioinformatics and Computational Biology 3(2), 185–205 (2003)
38. Wold, H.: Soft modeling: the basic design and some extensions. Systems Under Indirect Observation 2, 1–53 (1982)
39. Wold, H.: Partial least squares. Encyclopedia of the Statistical Sciences 6, 581–591 (1985)
40. Wold, S., Ruhe, H., Wold, H., Dunn, W.J.: The collinearity problem in linear regression-The partial least squares (PLS) approach to generalized inverse. SIAM Journal of Scientific and Statistical Computations 5, 735–743 (1984)
41. Tan, A.C., Gilbert, D.: Ensemble machine learning on gene expression data for cancer classification. Applied Bioinformatics 2, 575–583 (2003)
42. Huang, X., Pan, W., Han, X., Chen, Y., Miller, L.W.: Borrowing information from relevant microarray studies for sample classification using weighted partial least squares. Comput. Biol. Chem. 29, 204–211 (2005)
43. Cao, K.A., Roussouw, D., Robert-Granie, C., Besse, P.: A Sparse PLS for variable selection when integrating omics data. Statistical Applications in Genetics and Molecular Biology 7 (2008)
44. Ding, B., Gentleman, R.: Classification using generalized partial least squares. Bioconductor Project (2004)
45. Fort, G., Lambert-Lacroix, S.: Classification using partial least squares with penalized logistic regression. Bioinformatics 21, 1104–1111 (2005)
46. Nguyen, D.V., Rocke, D.M.: Tumor classification by partial least squares using microarray gene expression data. Bioinformatics 18, 39–50 (2002)
47. Wold, H.: Soft modeling: the basic design and some extensions. Systems Under Indirect Observation 2, 1–53 (1982)
48. De Jong, S.: SIMPLS: an alternative approach to partial least squares regression. Chemometrics and Intelligent Laboratory Systems 2(4), 251–263 (1993)

Neutrosophic Sets and Its Applications to Decision Making

Pinaki Majumdar

Abstract. This chapter introduces a new emerging tool for uncertain data processing which is known as neutrosophic sets. A neutrosophic set has the potentiality of being a general framework for uncertainty analysis in data sets also including big data sets. Here useful techniques like distance and similarity between two neutrosophic sets have been discussed. These notions are very important in the determination of interacting segments in a data set. Also the notion of entropy has been introduced to measure the amount of uncertainty expressed by a neutrosophic set. Further the notion of neutrosophic sets have been generalized and combined with soft sets to form a new hybrid set called interval valued neutrosophic sets. Some important properties of these sets under various algebraic operations have also been shown here.

1 Introduction

The first successful attempt towards incorporating non-probabilistic uncertainty, i.e. uncertainty which is not caused by randomness of an event, into mathematical modeling was made in 1965 by L. A. Zadeh [20] through his remarkable theory on fuzzy sets (FST). A fuzzy set is a set where each element of the universe belongs to it but with some grade or degree of belonging-ness which lies between 0 and 1 and such grades are called membership value of an element in that set. This gradation concept is very well suited for applications involving imprecise data such as natural language processing or in artificial intelligence, handwriting and speech recognition etc. Although Fuzzy set theory is very successful in handling uncertainties arising from vague-ness or partial belongingness of an element in a set, it cannot model all sorts of uncertainties pre-vailing in different real physical problems such as

Pinaki Majumdar
Department of Mathematics, M.U.C Womens College, Burdwan, West-Bengal, India
e-mail: pmajumdar2@rediffmail.com

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_4

problems involving incomplete information. Further generalization of this fuzzy set was made by K. Atanassov [1] in 1986, which is known as Intuitionistic fuzzy sets (IFS). In IFS, instead of one membership grade, there is also a non-membership grade attached with each element. Further there is a restriction that the sum of these two grades is less or equal to unity. In IFS the degree of non-belongingness is not independent but it is dependent on the degree of belongingness. FST can be considered as a special case of an IFS where the degree of non-belongingness of an element is exactly equal to 1 minus the degree of belongingness. IFS have the ability to handle imprecise data of both complete and incomplete in nature. In applications like expert systems, belief systems and information fusion etc., where degree of non-belongingness is equally important as degree of belongingness, intuitionistic fuzzy sets are quite useful. There are of course several other generalizations of Fuzzy as well as Intuitionistic fuzzy sets like L-fuzzy sets and intuitionistic L- fuzzy sets, interval valued fuzzy and intuitionistic fuzzy sets etc that have been developed and applied in solving many practical physical problems [2, 5, 6, 16].

Recently a new theory has been introduced which is known as neutrosophic logic and sets. The term neutro-sophy means knowledge of neutral thought and this neutral represents the main distinction between fuzzy and intuitionistic fuzzy logic and set. Neutrosophic logic was introduced by Florentin Smarandache [14] in 1995. It is a logic in which each proposition is estimated to have a degree of truth (T), a degree of indeterminacy (I) and a degree of falsity (F). A Neutrosophic set is a set where each element of the universe has a degree of truth, indeterminacy and falsity respectively and which lies between [0, 1]*, the non-standard unit interval. Unlike in intuitionistic fuzzy sets, where the incorporated uncertainty is dependent of the degree of belongingness and degree of non-belongingness, here the uncertainty present, i.e. the indeterminacy factor, is independent of truth and falsity values. Neutrosophic sets are indeed more general than IFS as there are no constraints between the degree of truth, degree of inde-terminacy and degree of falsity. All these degrees can individually vary within [0, 1]*.

In 2005, Wang et. al. [17] introduced an instance of neutrosophic set known as single valued neutrosophic sets which was motivated from the practical point of view and that can be used in real scientific and engineering applications. Here the degree of truth, indeterminacy and falsity respectively of any element of a neutrosophic set lies in standard unit interval [0, 1]. The single valued neutrosophic set is a generalization of classical set, fuzzy set, intuitionistic fuzzy set and paraconsistant sets etc.

The organization of the rest of this chapter is as follows: In section 2, discussions about the single valued neutrosophic sets (SVNS) have been done. Here several operations on them have been defined and some of their basic properties are studied. The distance and similarity between two SVNSs are discussed in section 3. The notion of entropy of a SVNS has also been discussed in this section. In section 4, a special hybrid neutrosophic set is defined. In this new set a combination of interval valued neutrosophic sets and soft sets are done to form interval valued neutrosophic set (IVNSS). These hybrid sets have greater powers of expressing and handling uncertainty than its predecessors. Section 5 concludes this chapter.

2 Single Valued Neutrosophic Multisets

This section clearly discusses single valued neutrosophic multisets as defined by [17]. A single valued neutrosophic set has been defined in [17] as follows. In addition, the definitions of complement and containment are also defined below.

Definition 0.1. Let X be a universal set. A Neutrosophic set A in X is characterized by a truth-membership function t_A, a indeterminacy-membership function i_A and a falsity-membership function f_A, where $t_A, i_A, f_A : X \to [0,1]$, are functions and $\forall x \in X, x \equiv x(t_A(x), i_A(x), f_A(x)) \in A$, is a single valued neutrosophic element of A.

A single valued neutrosophic set A (SVNS in short) over a finite universe $X = \{x_1, x_2, x_3, \cdots, x_n\}$ is represented as below:

$$A = \sum_{i=1}^{n} \frac{x_i}{< t_A(x_i), i_A(x_i), f_A(x_i) >}$$

Example 0.1. Assume that $X = \{x_1, x_2, x_3\}$, where x_1 is capacity, x_2 is trustworthiness and x_3 is price of a machine, be the universal set. The values of x_1, x_2, x_3 are in $[0,1]$. They are obtained from the questionnaire of some domain experts, their option could be a degree of 'good service', a degree of 'indeterminacy' and a degree of 'poor service'. A is a single valued Neutrosophic set of defined by

$$A = \frac{< 0.3, 0.4, 0.5 >}{x_1} + \frac{< 0.5, 0.2, 0.3 >}{x_2} + \frac{< 0.7, 0.2, 0.2 >}{x_3}$$

The following is a graphical representation of a single valued neutrosophic set. The elements of a single valued neutrosophic set always remain inside and on a closed unit cube.

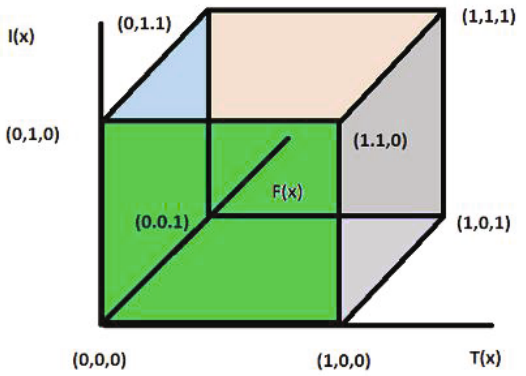


Fig. 1 A Single Valued Neutrospphic Set

Definition 0.2. The complement of a SVNS A is denoted by A^c and is defined by $t_{A^c}(x) = f_A(x)$; $i_{A^c}(x) = 1 - i_A(x)$ and $f_{A^c}(x) = t_A(x)\ \forall x \in X$.

Definition 0.3. A SVNS A is contained in the other SVNS B, denoted as $A \subset B$, if and only if $t_A(x) \leq t_B(x)$; $i_A(x) \leq i_B(x)$ and $f_A(x) \geq f_B(x)$ $\forall x \in X$. Two sets will be equal, i.e. $A = B$ if and only if $A \subset B$ and $B \subset A$.

Let us denote the collection of all SVNS in X as $N(X)$. Several operations like union and intersection has been defined on SVNSs and they satisfy most of the common algebraic properties of ordinary sets.

Definition 0.4. The union of two SVNS A and B is a SVNS C, written as $C = A \cup B$, and is defined as $t_C(x) = Max(t_A(x), t_B(x))$; $i_C(x) = Max(i_A(x), i_B(x))$ and $f_C(x) = Min(f_A(x), f_B(x))$ $\forall x \in X$

Definition 0.5. The intersection of two SVNS A and B is a SVNS C, written as $C = A \cap B$, and is defined as $t_C(x) = Min(t_A(x), t_B(x))$; $i_C(x) = Min(i_A(x), i_B(x))$ and $f_C(x) = Max(f_A(x), f_B(x))$ $\forall x \in X$

For practical purpose, throughout the rest of this chapter, we have considered only SVNS over a finite universe. In addition, two operators, namely 'truth favorite' and 'falsity favorite' to remove indeterminacy in the SVNS and transform it into an IFS or a paraconsistant set is defined.

Definition 0.6. The truth favorite of a SVNS A is again a SVNS B written as $B = \Delta A$, which is defined as follows:

$T_B(x) = Min(T_A(x) + I_A(x), 1)$
$I_B(x) = 0$
$F_B(x) = F_A(x), \quad \forall x \in X$

Definition 0.7. The falsity favorite of a SVNS A is again a SVNS B written as $B = \nabla A$, which is defined as follows:

$T_B(x) = T_A(x)$
$I_B(x) = 0$
$F_B(x) = Min(F_A(x) + I_A(x), 1) \quad \forall x \in X$

Example 0.2. Consider the SVNS A as defined in example 1. The following is an example of truth and falsity favorite respectively of the SVNS as defined in the said example.

$$A = \frac{<0.3, 0.4, 0.5>}{x_1} + \frac{<0.5, 0.2, 0.3>}{x_2} + \frac{<0.7, 0.2, 0.2>}{x_3}$$

$$B = \Delta A = \frac{<0.7, 0.0, 0.5>}{x_1} + \frac{<0.7, 0.0, 0.3>}{x_2} + \frac{<0.9, 0.0, 0.2>}{x_3}$$

$$C = \nabla A = \frac{<0.3, 0.0, 0.9>}{x_1} + \frac{<0.5, 0.0, 0.5>}{x_2} + \frac{<0.7, 0.0, 0.4>}{x_3}$$

The following theorem states the important algebraic properties of the operations defined in this section. Proofs of the results can be easily derived from definitions and hence left for readers.

Theorem 0.1. *If A,B,C are three single valued neutrosophic sets, then the following properties holds:*

1. $A \cup B = B \cup A;\ A \cap B = B \cap A$
2. $A \cup (B \cup C) = (A \cup B) \cup C;\ A \cap (B \cap C) = (A \cap B) \cap C$
3. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C);\ A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
4. $A \cup A = A;\ A \cap A = A;\ \Delta\Delta A = A;\ \nabla\nabla A = A$
5. $A \cup (A \cap B) = A;\ A \cap (A \cup B) = A$
6. $(A \cup B)^c = A^c \cap B^c;\ (A \cap B)^c = A^c \cup B^c$

3 Distance, Similarity and Entropy of Single Valued Neutrosophic Multisets

Single valued neutrosophic sets [17] are special instance of neutrosophic sets which motivated from practical point of view that can be used in real scientific and engineering applications. Again distance and similarity are key concepts in a number of fields such as linguistics, psychology, computational intelligence etc. where comparison between two different patterns or images is required. On the other hand 'entropy' of a set is a measure of capability of expressing uncertainty which is present in the data. This section is devoted to the study of these three concepts namely, distance, similarity and entropy of single valued neutrosophic sets [11].

3.1 Distance between Two Neutrosophic Sets

This section introduces the notion of distance between two single valued neutrosophic sets A and B defined over the finite universe $X = \{x_1, x_2, x_3, \cdots, x_n\}$.

Definition 0.8. Let $A = \sum_{i=1}^{n} \frac{x_i}{<t_A(x_i), i_A(x_i), f_A(x_i)>}$ and $B = \sum_{i=1}^{n} \frac{x_i}{<t_B(x_i), i_B(x_i), f_B(x_i)>}$ be two single valued neutrosophic sets in $X = \{x_1, x_2, x_3, \cdots, x_n\}$. Then
The Hamming distance between A and B is defined as follows:

$$d_N(A,B) = \sum_{i=1}^{n} \{|t_A(x_i) - t_B(x_i)| + |i_A(x_i) - i_B(x_i)| + |f_A(x_i) - f_B(x_i)|\} \quad (1)$$

The normalized Hamming distance between A and B is defined as follows:

$$l_N(A,B) = \frac{1}{3n} \sum_{i=1}^{n} \{|t_A(x_i) - t_B(x_i)| + |i_A(x_i) - i_B(x_i)| + |f_A(x_i) - f_B(x_i)|\} \quad (2)$$

The Euclidian distance between A and B is defined as follows:

$$e_N(A,B) = \sqrt{\sum_{i=1}^{n}\{(t_A(x_i) - t_B(x_i))^2 + (i_A(x_i) - i_B(x_i))^2 + (f_A(x_i) - f_B(x_i))^2\}} \quad (3)$$

The normalized Euclidian distance between A and B is defined as follows:

$$q_N(A,B) = \sqrt{\frac{1}{3n}\sum_{i=1}^{n}\{(t_A(x_i) - t_B(x_i))^2 + (i_A(x_i) - i_B(x_i))^2 + (f_A(x_i) - f_B(x_i))^2\}} \quad (4)$$

Now for equations 1 4 the following inequations holds:

1. $0 \le d_N(A,B) \le 1$
2. $0 \le l_N(A,B) \le 1$
3. $0 \le e_N(A,B) \le \sqrt{(3n)}$
4. $0 \le q_N(A,B) \le 1$

Example 0.3. Let $X = \{a,b,c,d\}$ be the universe and A,B be two single valued neutrosophic sets in X defined as follows:

$$A = \{\frac{a}{<0.5,0.2,0.9>}, \frac{b}{<0.8,0.4,0.2>}, \frac{c}{<0.3,0.8,0.7>}, \frac{d}{<0.6,0.3,0.5>}\}$$

$$B = \{\frac{a}{<0.7,0.4,0.2>}, \frac{b}{<0.5,0.5,0.3>}, \frac{c}{<0.1,0.2,0.3>}, \frac{d}{<0.8,0.1,0.6>}\}$$

Then the distance between A and B is given as $d_N(A,B) = 0.33$. Similarly, the other three distances will be $l_N(A,B) = \frac{0.33}{12} = 0.0275$, $e_N(A,B) \cong 1.15$ and $q_N(A,B) \cong 0.33$. Then the following result can be easily proved.

Proposition 0.1. *The distances d_N, l_N, e_N, and q_N defined above are metric.*

Definition 0.9. The minimum or sure cardinality of a SVNS A is denoted as *min* $\sum count(A)$ or c^l and is defined as $c^l = \sum_{i=1}^{n} t_A(x_i)$. The maximum cardinality of A is denoted by $max\sum count(A)$ or c^{ll} and is defined as $c^{ll} = \sum_{i=1}^{n}\{t_A(x_i) + (1 - i_A(x_i))\}$. The cardinality of A is defined by the interval $[c^l, c^{ll}]$. Similarly for A^c, the minimum and maximum cardinality is defined as $min\sum count(A^c) = \sum_{i=1}^{n} f_A(x_i)$ and $max\sum count(A^c) = \sum_{i=1}^{n}\{f_A(x_i) + (1 - i_A(x_i))\}$.

Example 0.4. For the SVNS A given in example 3, the minimum and maximum cardinality is computed as below:

$c^l = \sum_{i=1}^{n} t_A(x_i) = 0.5 + 0.8 + 0.3 + 0.6 = 2.2$

$c^{ll} = \sum_{i=1}^{n}\{t_A(x_i) + (1 - i_A(x_i))\} = 1.3 + 1.4 + 0.5 + 1.3 = 4.5$

3.2 Similarity Measure between Two Single Valued Neutrosophic Sets

This section presents the notion of similarity between two SVNS. Various methods have adopted for calculating this similarity. The first method is based on distances defined in the previous section. The second one is based on a matching function and the last one is based on membership grades.

In general a similarity measure between two SVNS is a function defined as $S : N(X)^2 \to [0,1]$ which satisfies the following properties:

(i) $S(A,B) \in [0,1]$
(ii) $S(A,B) = 1 \Leftrightarrow A = B$
(iii) $S(A,B) = S(B,A)$
(iV) $A \subset B \subset C \Rightarrow S(A,C) \leq S(A,B) \wedge S(B,C)$

But individual measures may satisfy more properties in addition to (i) - (iv). Now similarity can be calculated using several techniques. Three common techniques adopted here are distance based measure, matching function based measure and membership grade based measure.

3.2.1 Distance Based Similarity Measure

Similarity is inversely proportional with the distance between them. Using the distances defined in equations (1)(4), measures of similarity s^1 between two SVNS A and B as follows:

$$s^1(A,B) = \frac{1}{1+d(A,B)} \tag{5}$$

For example if Hamming distance d_N is used, then the associated measure of similarity will be denoted by S^1_N and is defined as

$$s^1_N(A,B) = \frac{1}{1+d_N(A,B)} \tag{6}$$

For example, the similarity measure between the two SVNS defined in Example 3 will be

$$s^1(A,B) = \frac{1}{1+0.33} \cong 0.75$$

Proposition 0.2. *The distance based similarity measure, s^1, between two SVNS A and B satisfies the following properties:*

1. $0 \leq s^1(A,B) \leq 1$
2. $s^1(A,B) = 1$ *if and only if* $A = B$
3. $s^1(A,B) = s^1(B,A)$
4. $A \subset B \subset C \Rightarrow s^1(A,C) \leq s^1(A,B) \wedge s^1(B,C)$

Proof. The results (i) to (iii) are trivial and can be derived directly from definition. The Proof (iv) is furnished below. Let $A \subset B \subset C$. Then for all $x \in U$, we have

$t_A(x) \le t_B(x) \le t_C(x)$; $\quad i_A(x) \le i_B(x) \le i_C(x)$ and
$f_A(x) \ge f_B(x) \ge f_C(x)$. Again,
$|t_A(x) - t_B(x)| \le |t_A(x) - t_C(x)|$ and $|t_B(x) - t_C(x)| \le |t_A(x) - t_C(x)|$.
Similarly, we have
$|i_A(x) - i_B(x)| \le |i_A(x) - i_C(x)|$ and $|i_B(x) - i_C(x)| \le |i_A(x) - i_C(x)|$ and
$|f_A(x) - f_B(x)| \le |f_A(x) - f_C(x)|$ and $|f_B(x) - f_C(x)| \le |f_A(x) - f_C(x)|$.
Thus, we get
$d(A,B) \le d(A,C) \Rightarrow s^1(A,B) \ge s^1(A,C)$ and
$d(B,C) \le d(A,C) \Rightarrow s^1(B,C) \ge s^1(A,C)$.
It implies that
$s^1(A,C) \le s^1(A,B) \wedge s^1(B,C)$
This is true for all the distance functions defined in equations (1) to (4). Hence the proof. □

3.2.2 Similarity Measure Based on Matching Function

Consider a universe where each element x_i has a weight w_i. Then we require a new measure of similarity different from the one discussed earlier. A weighted similarity measure s^w between SVNS $A$and B can be defined using a matching function as follows:

$$s^w(A,B) = \frac{\sum_{i=1}^{n} w_i(t_A(x_i).t_B(x_i) + i_A(x_i).i_B(x_i) + f_A(x_i).f_B(x_i))^2}{\sum_{i=1}^{n} w_i\{(t_A(x_i)^2 + i_A(x_i)^2 + f_A(x_i)^2) \times (t_B(x_i)^2 + i_B(x_i)^2 + f_B(x_i)^2)\}} \tag{7}$$

Consider the example 3 as discussed earlier. Further let the elements a,b,c,d of the universe X have weights 0.1, 0.3, 0.5 and 0.2 respectively. Then the weighted similarity measure between the two SVNS will be

$$s^w(A,B) = \frac{0.1 \times 0.3721 + 0.3 \times 0.4356 + 0.5 \times 0.16 + 0.2 \times 0.6561}{0.1 \times 0.759 + 0.3 \times 0.4956 + 0.5 \times 0.1708 + 0.2 \times 0.707}$$

$$= \frac{0.37911}{0.45138} \cong 0.84$$

Proposition 0.3. *The weighted similarity measure s^w between two SVNS A and B satisfies the following properties:*

1. $0 \le s^w(A,B) \le 1$
2. $s^w(A,B) = 1$ *if* $A = B$
3. $s^w(A,B) = s^w(B,A)$

Proof. The above properties trivially follows from the definition of similarity measure and Cauchy-Schawarz inequality. □

3.2.3 Similarity Measure Based on Membership Degrees

This section discusses another similarity measure between two SVNS. Another measure of similarity s^2 between two SVNS A and B could be defined as follows:

$$s^2(A,B) = \frac{\sum_{i=1}^{n}\{Min\{t_A(x_i),t_B(x_i)\} + Min\{i_A(x_i),i_B(x_i)\} + Min\{f_A(x_i),f_B(x_i)\}\}}{\sum_{i=1}^{n}\{Max\{t_A(x_i),t_B(x_i)\} + Max\{i_A(x_i),i_B(x_i)\} + Max\{f_A(x_i),f_B(x_i)\}\}} \tag{8}$$

For example, the similarity measure s^2 between the two SVNS A and B defined as in example 3 is given as:

$$s^2(A,B) \cong \frac{3.8}{7.1} = 0.535$$

Proposition 0.4. *The distance based similarity measure s^2, between two SVNS A and B satisfies the following properties:*

1. $0 \le s^2(A,B) \le 1$
2. $s^2(A,B) = 1$ *if and only if* $A = B$
3. $s^2(A,B) = s^2(B,A)$
4. $A \subset B \subset C \Rightarrow s^2(A,C) \le s^2(A,B) \le s^2(B,C)$

Proof. Properties (1) and (3) readily follows from definition and hence left to the reader. □

Proof. 2 It is clear that, if $A = B$, then $s^2(A,B) = 1$.
Conversely, let $s^2(A,B) = 1$, Therefore,

$$\frac{\sum_{i=1}^{n}\{Min\{t_A(x_i),t_B(x_i)\} + Min\{i_A(x_i),i_B(x_i)\} + Min\{f_A(x_i),f_B(x_i)\}\}}{\sum_{i=1}^{n}\{Max\{t_A(x_i),t_B(x_i)\} + Max\{i_A(x_i),i_B(x_i)\} + Max\{f_A(x_i),f_B(x_i)\}\}} = 1$$

$\Rightarrow \sum_{i=1}^{n}\{Min\{t_A(x_i),t_B(x_i)\} + Min\{i_A(x_i),i_B(x_i)\} + Min\{f_A(x_i),f_B(x_i)\}\}$
$= \sum_{i=1}^{n}\{Max\{t_A(x_i),t_B(x_i)\} + Max\{i_A(x_i),i_B(x_i)\} + Max\{f_A(x_i),f_B(x_i)\}\}$
$\Rightarrow \sum_{i=1}^{n}\{[Min\{t_A(x_i),t_B(x_i)\} - Max\{t_A(x_i),t_B(x_i)\}] + [Min\{i_A(x_i),i_B(x_i)\} -$
$Max\{i_A(x_i),i_B(x_i)\}] + [Min\{f_A(x_i),f_B(x_i)\} - Max\{f_A(x_i),f_B(x_i)\}]\} = 0$
Thus for each x_i, it is clear that
$Min\{t_A(x_i),t_B(x_i)\} - Max\{t_A(x_i),t_B(x_i)\} = 0$
$Min\{i_A(x_i),i_B(x_i)\} - Max\{i_A(x_i),i_B(x_i)\} = 0$
$Min\{f_A(x_i),f_B(x_i)\} - Max\{f_A(x_i),f_B(x_i)\} = 0$
Therefore, it is clear that
$t_A(x_i) = t_B(x_i)$; $i_A(x_i) = i_B(x_i)$ and $f_A(x_i) = f_B(x_i)$ $\forall 1 \le i \le n$
It implies that $A = B$ □

Proof. 4 Let us consider $A \subset B \subset C$. Therefore
$t_A(x) \le t_B(x) \le t_C(x)$; $i_A(x) \le i_B(x) \le i_C(x)$ and $f_A(x) \ge f_B(x) \ge f_C(x)$ $\forall x \in U$
It implies that
$t_A(x) + i_A(x) + f_B(x) \ge t_A(x) + i_A(x) + f_C(x)$ and
$t_B(x) + i_B(x) + f_A(x) \ge t_C(x) + i_C(x) + f_A(x)$

Therefore,

$$s^2(A,B) = \frac{t_A(x)+i_A(x)+f_B(x)}{t_B(x)+i_B(x)+f_A(x)} \geq \frac{t_A(x)+i_A(x)+f_C(x)}{t_C(x)+i_C(x)+f_A(x)} = s^2(A,C)$$

Again, similarly:

$t_B(x)+i_B(x)+f_C(x) \geq t_A(x)+i_A(x)+f_C(x)$ and
$t_C(x)+i_C(x)+f_A(x) \geq t_C(x)+i_C(x)+f_B(x)$
Therefore,

$$s^2(B,C) = \frac{t_B(x)+i_B(x)+f_C(x)}{t_C(x)+i_C(x)+f_B(x)} \geq \frac{t_A(x)+i_A(x)+f_C(x)}{t_C(x)+i_C(x)+f_A(x)} = s^2(A,C)$$

It implies that $S^2(A,C) \leq S^2(A,B) \wedge S^2(B,C)$ □

3.2.4 Entropy of a Single Valued Neutrosophic Set

Entropy can be considered as a measure of uncertainty expressed by a set, whether it is fuzzy or intuitionistic fuzzy or vague etc. Here the SVNS are also capable of handling uncertain data, therefore as a natural consequence we are also interested in finding the entropy of a single valued neutrosophic set. Entropy as a measure of fuzziness was first mentioned by Zadeh [20] in 1965. Later De Luca-Termini [4] axiomatized the non-probabilistic entropy. According to them the entropy E of a fuzzy set A should satisfy the following axioms:

1. $E(A) = 0$ if and only if $A \in 2^X$
2. $E(A) = 1$ if and only if $\mu_A(x) = 0.5 \forall x \in X$
3. $E(A) \leq E(B)$ if and only if A is less fuzzy than B; i.e., if $\mu_A(x) \leq \mu_B(x) \leq 0.5 \quad \forall \quad x \in X$ or if $\mu_A(x) \geq \mu_B(x) \geq 0.5 \quad \forall \quad x \in X$
4. $E(A^c) = E(A)$

Several other authors have investigated the notion of entropy. Kaufmann [7] proposed a distance based measure of fuzzy entropy; Yager [19] gave another view of entropy or the degree of fuzziness of any fuzzy set in terms of lack of distinction between the fuzzy set and its complement. Kosko [8] investigated the fuzzy entropy in relation to a measure of subset hood. Szmidt & Kacprzyk [15] studied the entropy of intuitionistic fuzzy sets etc.. Several applications of fuzzy entropy in solving many practical problems like image processing, inventory, economics can be found in literatures [3, 13]. The following definition introduces the notion of entropy of a SVNS:

Definition 0.10. The entropy of SVNS is defined as a function $E_N : N(X) \to [0,1]$ which satisfies the following axioms:

1. $E_N(A) = 0$ if A is a crisp set.
2. $E_N(A) = 1$ if $(t_A(x), i_A(x), f_A(x)) = (0.5, 0.5, 0.5) \forall x \in X$.
3. $E_N(A) \geq E_N(B)$ if A is more uncertain than B, i.e., $t_A(x) + f_A(x) \leq t_B(x) + f_B(x)$ and $|i_A(x) - i_{A^c}(x)| \leq |i_B(x) - i_{B^c}(x)|$
4. $E_N(A) = E_N(A^c) \forall A \in N(X)$

Now notice that in a SVNS the presence of uncertainty is due to two factors, firstly due to the partial belongingness and partial non-belongingness and secondly due to the indeterminacy factor. Considering these two factors an entropy measure E_1 of a single valued neutrosophic sets A is proposed as follows:

$$E_1(A) = 1 - \frac{1}{n} \sum_{x_i \in X} (t_A(x_i) + f_A(x_i)) \times |i_A(x_i) - i_{A^c}(x_i)| \tag{9}$$

Proposition 0.5. *The entropy measure E_1 satisfies all the axioms given in definition 10.*

Proof. 1. *For a crisp set A, A^c is also crisp and $i_A(x) = 0 \forall x \in X$. Hence, $E_1(A) = 0$ holds.*
2. *If A be such that $(t_A(x), i_A(x), f_A(x)) = (0.5, 0.5, 0.5) \forall x \in X$, then $t_A(x) + f_A(x) = 1$ and $i_A(x) - i_{A^c} = 0.5 - 0.5 = 0 \forall x \in X$. It implies that $E_1(A) = 1$*
3. *The axim 3 holds from definition.*
4. *$E_1(A) = E_1(A^c)$ holds obviously from definition.*

Thus E_1 is an entropy function defined on $N(X)$ □

Example 0.5. Let $X = \{a, b, c, d\}$ be the universe and A be a single valued neutrosophic set in X defined as below:

$$A = \{\frac{a}{< 0.5, 0.2, 0.9 >}, \frac{b}{< 0.8, 0.4, 0.2 >}, \frac{c}{< 0.3, 0.8, 0.7 >}, \frac{d}{< 0.6, 0.3, 0.5 >}\}$$

Then the entropy of A will be $E_1(A) = 1 - 0.52 = 0.48$

4 Interval Valued Neutrosophic Soft Sets

In this section the notion of interval valued Neutrosophic soft sets (IVNSS) [10] has been intro-duced which is a hybrid structure, combining interval Neutrosophic sets and soft sets. Neutroso-phic sets quantify the notion of 'indeterminacy' and a soft set provides parameterization tool. Both of these are very useful tool for modeling uncertainty. This new set is more capable in ex-pressing uncertain situation than other variants of soft or fuzzy sets. It has been shown that this structure is the most general structure available till now containing classical sets, fuzzy sets, intuitionistic fuzzy sets, soft sets etc as special cases. Some basic operations are also defined on IVNSS and a few important properties of it are studied in this section of this chapter. An application of interval valued Neutrosophic soft sets in decision making has also been given at the end of this chapter.

Fig. 2 Relation between interval neutrosophic set and other sets

4.1 Soft Set

First we define a soft set. Soft set was invented by Molodtsov [12] in 1999 and it is a parametric representation of the universe. Let U be an initial universal set and E be a set of parameters. Let $P(U)$ denote the power set of U. A pair (F,A) is called a soft set over U if and only if F is a mapping given by $F : A \to P(U)$, where $A \subset E$.

As an illustration, consider the following example. Suppose a soft set (F,A) describes attractiveness of the shirts which the authors are going to wear for a party. Let $U =$ the set of all shirts under consideration $= \{x_1,x_2,x_3,x_4,x_5\}$; $A = \{$colorful, bright, cheap, warm$\} = \{e_1,e_2,e_3,e_4\}$. Let $F(e_1) = \{x_1,x_2\}, F(e_2) = \{x_1,x_2,x_3\}, F(e_3) = \{x_4\}, F(e_4) = \{x_2,x_5\}$. So, the soft set (F,A) is a subfamily $\{F(e_i), i = 1,2,3,4\}$ of $P(U)$. Here $F(e_i)$ is called an e_i-approximation of (F,A).

4.2 Interval Valued Neutrosophic Soft Sets

In this section, the notion of interval valued neutrosophic soft sets (IVNSS) is introduced and its properties are studied. Before, the concept is introduced, the definition of interval neutrosophic set [18] is presented.

Let X be a universal set and $x \in X$. An interval neutrosophic set (INS) A in X is characterized by a truth-membership function T_A, a indeterminacy membership function I_A and a falsity membership function F_A and $\forall x \in X$, $x \equiv x(T_A(x), I_A(x), F_A(x)) \in A$ and $T_A(x), I_A(x), F_A(x)$ are closed subintervals of [0, 1].

Example 0.6. Assume that $X = \{x_1,x_2,x_3\}$, where x_1 is capacity, x_2 is trustworthiness and, x_3 is price of a machine, be the universal set. The truth, indeterminacy and falsity values of x_1,x_2,x_3 are closed subintervals of $[0,1]$. They are obtained from the questionnaire of some domain experts, their option could be a degree of 'good service', a degree of indeterminacy and a degree of 'poor service'. A is an interval neutrosophic set of X defined by

$$A = \frac{< [0.2,0.4],[0.3,0.5],[0.3,0.5] >}{x_1} + \frac{< [0.5,0.7],[0.0,0.2],[0.2,0.3] >}{x_2} + \frac{< [0.6,0.8],[0.2,0.3],[0.2,0.3] >}{x_3}$$

Interval neutrosophic set are most general in nature because by adjusting the values of T_A, I_A and f_A, we get a varity of other sets as special cases. The following Figure 2 shows the relationship among interval neutrosophic sets and other sets.

Definition 0.11. Let U be an initial universal set and E be a set of parameters. Let $IN(U)$ denote the set of all interval valued neutrosophic sets on U. A pair (G,A) is called an interval valued neutrosophic soft set (IVNSS) over U if G is a mapping given by $G : A \to IN(U)$, where $A \subset E$. Thus, for all $e \in A$, $G(e) = < x, T_e^G(x), I_e^G(x), F_e^G(x) > \in IN(U)$.

Example 0.7. Let the universal set be $X = \{x_1, x_2, x_3\}$ where x_1 is capacity, x_2 is trustworthiness and, x_3 is price of a machine which is tested against certain parameters $E = \{e_1, e_2, e_3\}$, where e_1 is high altitude, e_2 is high temperature and, e_3 is high pressure. The values of x_1, x_2, x_3 are closed subintervals of $[0,1]$. They are obtained from the questionnaire of some domain experts; their option could be a degree of 'good service', a degree of indeterminacy and a degree of 'poor service' with respect to each parameter. Then G is an interval valued neutrosophic soft set of X defined as $G = \{G(e_1), G(e_2), G(e_3)\}$, where

$$G(e_1) = \frac{< [0.2,0.4],[0.3,0.5],[0.3,0.5] >}{x_1} + \frac{< [0.5,0.7],[0.0,0.2],[0.2,0.3] >}{x_2} + \frac{< [0.6,0.8],[0.2,0.3],[0.2,0.3] >}{x_3}$$

$$G(e_2) = \frac{< [0.3,0.6],[0.1,0.5],[0.3,0.6] >}{x_1} + \frac{< [0.2,0.7],[0.1,0.3],[0.2,0.5] >}{x_2} + \frac{< [0.4,0.7],[0.4,0.6],[0.2,0.5] >}{x_3}$$

$$G(e_3) = \frac{< [0.5,0.9],[0.1,0.3],[0.2,0.4] >}{x_1} + \frac{< [0.4,0.7],[0.1,0.2],[0.1,0.4] >}{x_2} + \frac{< [0.5,0.8],[0.2,0.4],[0.1,0.5] >}{x_3}$$

A matrix representation of an IVNSS is given as $G = (G_1, G_2, G_3)$, where G_1, G_2 and G_3 are column matrices such that

$$G_1 = \begin{pmatrix} ([0.2,0.4],[0.3,0.5],[0.3,0.5]) \\ ([0.3,0.6],[0.1,0.5],[0.3,0.6]) \\ ([0.5,0.9],[0.1,0.3],[0.2,0.4]) \end{pmatrix} \quad G_2 = \begin{pmatrix} ([0.5,0.7],[0.0,0.2],[0.2,0.3]) \\ ([0.2,0.7],[0.1,0.3],[0.2,0.5]) \\ ([0.4,0.7],[0.1,0.2],[0.1,0.4]) \end{pmatrix}$$

$$G_3 = \begin{pmatrix} ([0.6,0.8],[0.2,0.3],[0.2,0.3]) \\ ([0.4,0.7],[0.4,0.6],[0.2,0.5]) \\ ([0.5,0.8],[0.2,0.4],[0.1,0.5]) \end{pmatrix}$$

Definition 0.12. An interval valued neutrosophic soft set (IVNSS) (G,A) is said to be null or empty if and only if it satisfies the following conditions for all $x \in X$ and for all $e \in A$. It is denoted as $\tilde{\phi_N}$.

1. inf $T_e^G(x) = supT_e^G(x) = 0$
2. inf $I_e^G(x) = supI_e^G(x) = 1$
3. inf $F_e^G(x) = supF_e^G(x) = 1$

Definition 0.13. An interval valued neutrosophic soft set (IVNSS) (G,A) is said to be absolute if and only if it satisfies the following conditions for all $x \in X$ and for all $e \in A$. It is denoted as $\tilde{A_N}$.

1. inf $T_e^G(x) = supT_e^G(x) = 1$
2. inf $I_e^G(x) = supI_e^G(x) = 0$
3. inf $F_e^G(x) = supF_e^G(x) = 0$

4.2.1 Set Theoretic Operations on Interval Valued Neutrosophic Soft Set

This section discusses various set theoretic operations on interval valued neutrosophic soft sets and study some basic properties.

Definition 0.14. An IVNSS (P,A) is said to be a subset of another IVNSS (G,B) if and only if it satisfies the following conditions. It is denoted as $(P,A) \subset_N (G,B)$.

1. $A \subset B$ and
2. for all $x \in X$ and for all $e \in A$

   a. $infT_e^P(x) \leq infT_e^G(x)$; $\quad supT_e^P(x) \leq supT_e^G(x)$
   b. $infI_e^P(x) \geq infI_e^G(x)$; $\quad supI_e^P(x) \geq supI_e^G(x)$
   c. $infF_e^P(x) \geq infF_e^G(x)$; $\quad supF_e^P(x) \geq supF_e^G(x)$

Definition 0.15. Two IVNSS (P,A) and (G,B) is said to be equal, denoted by $(P,A) =_N (G,B)$ if and only if $(P,A) \subset_N (G,B)$; $(G,B) \subset_N (P,A)$; $(F,A) \subset_N (G,B)$ and $(G,B) \subset_N (F,A)$.

Definition 0.16. The complement of an IVNSS (P,A), denoted by $(P,A)^c = (P^c,A)$ is defined as below. For all $x \in X$ and for all $e \in A$

1. $T_e^{P^c}(x) = F_e^P(x)$
2. inf $I_e^{P^c}(x) = 1 - supI_e^P(x)$
3. sup $I_e^{P^c}(x) = 1 - infI_e^P(x)$
4. $F_e^{P^c}(x) = T_e^P(x)$

Definition 0.17. The union of two IVNSS (P,A) and (G,B) denoted by $(P,A) \cup_N (G,B)$ is an IVNSS (H,C) defined by

1. $C = A \cup B$ and

2. for all $e \in (A \cap B)$
   $\inf T_e^H(x) = max(infT_e^P(x), infT_e^G(x)); supT_e^H(x) = max(supT_e^P(x), supT_e^G(x))$
   $\inf I_e^H(x) = min(infI_e^P(x), infI_e^G(x)); supI_e^H(x) = min(supI_e^P(x), supI_e^G(x))$
   $\inf F_e^H(x) = min(infF_e^P(x), infF_e^G(x)); supF_e^H(x) = min(supF_e^P(x), supF_e^G(x))$
3. $\forall e \in (A-B), H(e) = P(e)$
4. $\forall e \in (B-A), H(e) = G(e)$

Definition 0.18. The intersection of two IVNSS (P,A) and (G,B) denoted by $(P,A) \cap_N (G,B)$ is an IVNSS (H,C) defined by

1. $C = A \cap B$ and
2. for all $e \in C$
   $\inf T_e^H(x) = min(infT_e^P(x), infT_e^G(x)); supT_e^H(x) = min(supT_e^P(x), supT_e^G(x))$
   $\inf I_e^H(x) = max(infI_e^P(x), infI_e^G(x)); supI_e^H(x) = max(supI_e^P(x), supI_e^G(x))$
   $\inf F_e^H(x) = max(infF_e^P(x), infF_e^G(x)); supF_e^H(x) = max(supF_e^P(x), supF_e^G(x))$

Theorem 0.2. *The following properties of union and intersection are satisfied by the interval valued neutrosophic soft sets. The proofs directly follows from definition.*

1. $(F,A) \cup_N (G,B) = (G,B) \cup_N (F,A)$
2. $(F,A) \cap_N (G,B) = (G,B) \cap_N (F,A)$
3. $((F,A) \cup_N (G,B)) \cup_N (H,C) = (F,A) \cup_N ((G,B) \cup_N (H,C))$
4. $((F,A) \cap_N (G,B)) \cap_N (H,C) = (F,A) \cap_N ((G,B) \cap_N (H,C))$
5. $((F,A) \cup_N (G,B)) \cap_N (H,C) = ((F,A) \cup_N (G,B)) \cap_N ((F,A) \cup_N (H,C))$
6. $((F,A) \cap_N (G,B)) \cup_N (H,C) = ((F,A) \cap_N (G,B)) \cup_N ((F,A) \cap_N (H,C))$
7. $(F,A) \cup_N (F,A) = (F,A)$
8. $(F,A) \cap_N (F,A) = (F,A)$
9. $(F,A) \cup_N \tilde{\phi}_N = (F,A)$
10. $(F,A) \cap_N \tilde{\phi}_N = \tilde{\phi}_N$
11. $(F,A) \cup_N \tilde{A}_N = \tilde{A}_N$
12. $(F,A) \cap_N \tilde{A}_N = (F,A)$

Next few more definitions regarding interval valued intuitionistic soft sets are presented. Let (X,E) be the soft universe. Then to remove the indeterminacy in IVNSS and transform it into an interval valued intuitionistic fuzzy soft sets, the following unique operations are defined.

Definition 0.19. (Truth-Favorite) The truth-favorite of IVNSS (G,A) is an IVNSS (H,B) written as $(H,B) = \Delta(G,A)$ or $H = \Delta G$ and is defined as below. This is used to evaluate the maximum degree of truth membership.

1. $B = A$
2. For all $x \in X$ and for all $e \in (A = B)$
   $infT_e^H(x) = min(infT_e^G(x) + infI_e^G(x), 1)$
   $supT_e^H(x) = min(supT_e^G(x) + supI_e^G(x), 1)$
   $infI_e^H(x) = 0 = supI_e^H(x)$
   $infF_e^H(x) = infF_e^G(x)$
   $supF_e^H(x) = supF_e^G(x)$

Definition 0.20. (False-Favorite) The false-favorite of IVNSS (G,A) is an IVNSS (H,B) written as $(H,B) = \nabla(G,A)$ or $H = \nabla G$ and is defined as below. This is used to evaluate the maximum degree of false membership.

1. $B = A$
2. For all $x \in X$ and for all $e \in (A = B)$
   $infT_e^H(x) = infT_e^G(x)$
   $supT_e^H(x) = supT_e^G(x)$
   $infI_e^H(x) = 0 = supI_e^H(x)$
   $infF_e^H(x) = min(infF_e^G(x) + infI_e^G(x), 1)$
   $supF_e^H(x) = min(supF_e^G(x) + supI_e^G(x), 1)$

Example 0.8. The truth-favorite $\Delta G = (\Delta G_1, \Delta G_2, \Delta G_3)$ and false-favorite $\nabla G = (\nabla G_1, \nabla G_2, \nabla G_3)$ IVNSS of the IVNSS (G,A), given in example 7 is furnished below.

$$\Delta G_1 = \begin{pmatrix} ([0.5,0.9],[0,0],[0.3,0.5]) \\ ([0.4,1.0],[0,0],[0.3,0.6]) \\ ([0.6,1.0],[0,0],[0.2,0.4]) \end{pmatrix}; \quad \Delta G_2 = \begin{pmatrix} ([0.5,0.9],[0,0],[0.2,0.3]) \\ ([0.3,1.0],[0,0],[0.2,0.5]) \\ ([0.5,0.9],[0,0],[0.1,0.4]) \end{pmatrix}$$

$$\Delta G_3 = \begin{pmatrix} ([0.8,1.0],[0,0],[0.2,0.3]) \\ ([0.8,1.0],[0,0],[0.2,0.5]) \\ ([0.7,1.0],[0,0],[0.1,0.5]) \end{pmatrix}; \quad \nabla G_1 = \begin{pmatrix} ([0.2,0.4],[0,0],[0.6,1.0]) \\ ([0.3,0.6],[0,0],[0.4,1.0]) \\ ([0.5,0.9],[0,0],[0.3,0.7]) \end{pmatrix}$$

$$\nabla G_2 = \begin{pmatrix} ([0.5,0.7],[0,0],[0.2,0.5]) \\ ([0.2,0.7],[0,0],[0.3,0.8]) \\ ([0.4,0.7],[0,0],[0.2,0.6]) \end{pmatrix}; \quad \nabla G_3 = \begin{pmatrix} ([0.6,0.8],[0,0],[0.4,0.6]) \\ ([0.4,0.7],[0,0],[0.6,1.0]) \\ ([0.5,0.8],[0,0],[0.3,0.9]) \end{pmatrix}$$

The relationship between interval valued neutrosophic soft sets and other sets are shown below. An interval valued neutrosophic soft set will be a

1. Soft set if for all $x \in X$ and for all $e \in E$, $I_e^F(x) = \phi$, $infT_e^F(x) = supT_e^F(x) = 0$ or 1; $infF_e^F(x) = supF_e^F(x) = 0$ or 1 and $supT_e^F(x) + supF_e^F(x) = 1$.
2. Fuzzy soft set if for all $x \in X$ and for all $e \in E$, $I_e^F(x) = \phi$, $infT_e^F(x) = supT_e^F(x) \in [0,1]$; $infF_e^F(x) = supF_e^F(x) \in [0,1]$ and $supT_e^F(x) + supF_e^F(x) = 1$.
3. Interval valued fuzzy soft set if for all $x \in X$ and for all $e \in E$, $I_e^F(x) = \phi$, $infT_e^F(x), supT_e^F(x), infF_e^F(x), supF_e^F(x) \in [0,1]$, $supT_e^F(x) + infF_e^F(x) = 1$ and $infT_e^F(x) + supF_e^F(x) = 1$.
4. Intuitionistic fuzzy soft set if for all $x \in X$ and for all $e \in E$, $I_e^F(x) = \phi$, $infT_e^F(x) = supT_e^F(x) \in [0,1]$; $infF_e^F(x) = supF_e^F(x) \in [0,1]$ and $supT_e^F(x) + supF_e^F(x) \leq 1$.
5. Interval valued intuitionistic fuzzy soft set if for all $x \in X$ and for all $e \in E$, $I_e^F(x) = \phi$, $infT_e^F(x), supT_e^F(x), infF_e^F(x), supF_e^F(x) \in [0,1]$, and $supT_e^F(x) + supF_e^F(x) \leq 1$.

Therefore, interval valued neutrosophic soft sets represent the most general case.

4.3 An Application of IVNSS in Decision Making

Consider a set of three machines $X = \{x_1, x_2, x_3\}$, whose performances are tested against certain parameters $E = \{e_1, e_2, e_3\}$, where e_1 is high altitude, e_2 is high temperature, and e_3 is high pressure. The values of x_1, x_2 and x_3 are subintervals of [0, 1]. They are obtained from some domain experts; their option could be a degree of 'good performance', a degree of indeterminacy and a degree of 'poor performance' with respect to each parameter. Then $F = (F_1, F_2, F_3)$ is an interval valued neutrosophic soft set of X defined as follows:

$$F_1 = \begin{pmatrix} ([0.4,0.7],[0.3,0.6],[0.3,0.5]) \\ ([0.4,0.6],[0.2,0.5],[0.4,0.6]) \\ ([0.4,0.9],[0.2,0.3],[0.1,0.4]) \end{pmatrix}; \quad F_2 = \begin{pmatrix} ([0.5,0.7],[0.1,0.2],[0.2,0.3]) \\ ([0.4,0.7],[0.2,0.5],[0.2,0.5]) \\ ([0.3,0.7],[0.1,0.2],[0.1,0.4]) \end{pmatrix}$$

$$F_3 = \begin{pmatrix} ([0.2,0.8],[0.2,0.3],[0.2,0.3]) \\ ([0.2,0.7],[0.4,0.6],[0.2,0.5]) \\ ([0.4,0.8],[0.2,0.4],[0.1,0.5]) \end{pmatrix}$$

Now to take decision that which machine is the best among the three, one has to compute 'parametric score' and 'total score' of an element in an IVNSS. The parametric score and total score of an element is defined as below.

Let $x \in X$ be an element of an IVNSS A such that $A = \{A(e), e \in E\}$. Now for $x \in A(e)$, $x \equiv x(T_A(x), I_A(x), F_A(x)) \in A$ and $T_A(x), I_A(x), F_A(x) \subseteq [0,1]$. The parametric score $S_e(x)$ of an element x in A is defined as below:

$$S_e(x) = \frac{[(T_A^+(x) - F_A^+(x)) + (T_A^-(x) - F_A^-(x))]}{2} \times (1 - \frac{I_A^+(x) + T_A^-(x)}{2}) \qquad (10)$$

where $T_A^+(x) = supT_e^A(x)$, $T_A^-(x) = infT_e^A(x)$, $I_A^+(x) = supI_e^A(x)$, $I_A^-(x) = infI_e^A(x)$, $F_A^+(x) = supF_e^A(x)$ and $F_A^-(x) = infF_e^A(x)$.

The total score, $S_T(x)$, of an element x in A is defined as below:

$$S_T(x) = \sum_{e \in E} S_e(x) \qquad (11)$$

The computed result for the above case is presented in the following table 1. From the computational result, it is clear that the second machine x_2 has the highest total score and hence will be selected.

Table 1 Computational result of parametric and total score

S	x_1	x_2	x_3
e_1	0.0825	0.2975	0.1875
e_2	0.0	0.13	0.05
e_3	0.3	0.2125	0.21
Total score	0.3825	0.64	0.4475

5 Conclusion

The recently proposed notion of neutrosophic sets is a general framework for studying uncertainties arising in data processing due to indeterminacy factors. From the philosophical point of view, it has been shown that a neutrosophic set generalizes a classical set, fuzzy set, interval valued fuzzy set etc. A single valued neutrosophic set is an instance of neutrosophic sets which can be used in real scientific problems. Interval valued neutrosophic sets are again further generalization of neutrosophic and interval valued neutrosophic sets which additionally has parameterization tool in it. Similarity and entropy measures of neutrosophic sets have several real applications involving analysis of data, including also big data sets. This is an emerging field of study and the author is confident that the discussions in this chapter will be helpful to future researchers inter-ested in this area of research.

Acknowledgements. The present work is partially supported by University Grants Commission (ERO) under minor research project (Science)(Project No. F.PSW-19/12-13).

References

1. Atanassov, K.: Intuitionistic Fuzzy Sets. Fuzzy Sets and Systems 20, 87–96 (1986)
2. Atanassov, K., Stoeva, S.: Intuitionistic L Fuzzy Sets. Cybernetics and System Research 2, 539–540 (1984)
3. Cheng, C.C., Liao, K.H.: Parameter Optimization based on Entropy Weight and Triangular Fuzzy Number. International Journal of Engineering and Industries 2(2), 62–75 (2011)
4. De Luca, A., Termini, S.: A Definition of a Non-Probabilistic Entropy in the Setting of Fuzzy Sets Theory. Information and Control 20, 301–312 (1972)
5. Goguen, J.A.: L Fuzzy Sets. Journal of Mathematical Analysis and Applications 18, 145–174 (1963)
6. Jiang, Y., Tang, Y., Chen, Q., Liu, H.: Interval-Valued Intuitionistic Fuzzy Soft Sets and their Properties. Computers and Mathematics with Applications 60(3), 906–918 (2010)
7. Kaufmann, A.: Introduction to the Theory of Fuzzy Subsets. Academic Press, New York (1975)
8. Kosoko, B.: Fuzzy Entropy and Conditioning. Information Sciences 40(2), 165–174 (1986)
9. Maji, P.K., Biswas, R., Roy, A.R.: Soft Set Theory. Computers and Mathematics with Applications 45, 555–562 (2003)

10. Majumdar, P.: A Study of Several Types of Sets Expressing Uncertainty and Some Applications on them. Ph.D. Thesis, Visva Bharati University, India (2013)
11. Majumdar, P., Samanta, S.: On Similarity and Entropy of Neutrosophic Sets. Journal of Intelligent and Fuzzy Systems 26, 1245–1252 (2014)
12. Molodtsov, D.: Soft Set Theory-First Results. Computers and Mathematics with Applications 37, 19–31 (1999)
13. Pasha, E.: Fuzzy Entropy as Cost Function in Image Processing. In: Proceeding of the 2nd IMTGT Regional Conference on Mathematics, Statistics and Applications. Universiti Sains, Malaysia, Penang. (2006)
14. Smarandache, F.: A Unifying Field in Logics, Neutrosophy: Neutrosophic Probability, Set and Logic. American Research Press (1999)
15. Szmidt, E., Kacprzyk, J.: Entropy for Intuitionistic Fuzzy Sets. Fuzzy Sets and Systems 118, 467–477 (2001)
16. Tuskan, I.: Interval Valued Fuzzy Sets based on Normal Forms. Fuzzy Sets and Systems 20, 191–210 (1986)
17. Wang, H., Smarandache, F., Zhang, Y., Sunderraman, R.: Single Valued Neutrosophic Sets. In: Proceedings of 10th International Conference on Fuzzy Theory & Technology, Salt Lake City, Utah (2005)
18. Wang, H., Smarandache, F., Zhang, Y.Q., Sunderraman, R.: Interval Neutrosophic Sets and Logic: Theory and Applications in Computing, Hexis, AZ (2005)
19. Yagar, R.R.: On the Measure of Fuzziness and Negation, Part I: Membership in the Unit Interval. International Journal of General Systems 5, 189–200 (1979)
20. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)

Part II
Architecture for Big Data Analysis

An Efficient Grouping Genetic Algorithm for Data Clustering and Big Data Analysis

Sayede Houri Razavi, E. Omid Mahdi Ebadati, Shahrokh Asadi, and Harleen Kaur

Abstract. Clustering as a formal, systematic subject in dissertations can be considered the most influential unsupervised learning problem; so, as every other problem of this kind, it deals with finding the structure in a collection of unlabeled data. One of the matters associated with this subject is undoubtedly determination of the number of clusters. In this chapter, an efficient grouping genetic algorithm is proposed under the circumstances of an anonymous number of clusters. Concurrent clustering with different number of clusters is implemented on the same data in each chromosome of grouping genetic algorithm in order to discern the accurate number of clusters. In subsequent iterations of the algorithm, new solutions with different clusters number or distinct accuracy of clustering are produced by application of efficient crossover and mutation operators that led to significant improvement of clustering. Furthermore, a local search by a special probability is applied in each chromosome of each new population in order to increase the accuracy of clustering.These special operators will lead to the successful application of the proposed method in the big data analysis. To prove the accuracy and the efficiency

Sayede Houri Razavi
Department of Knowledge Engineering and Decision Science,
University of Economic Science, Tehran, Iran
e-mail: houririazavi@ues.ac.ir, houririazavi@gmail.com

E. Omid Mahdi Ebadati
Department of Mathematics and Computer Science,
University of Economic Science, Tehran, Iran
e-mail: omidit@gmail.com, ebadati@ues.ac.ir

Shahrokh Asadi
Department of Industrial Engineering,
Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran
e-mail: s.asadi@aut.ac.ir

Harleen Kaur
Department of Computer Science, Hamdard University, New Delhi, India
e-mail: harleen_k1@rediffmail.com, harleen@jamiahamdard.ac.in

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_5

of the algorithm, its tested on various artificial and real data sets in a comparable manner. Most of the datasets consisted of overlapping clusters, but the algorithm could detect the proper number of all data sets with high accuracy of clustering. The consequences make the best evidence of the algorithms successful performance of finding an appropriate number of clusters and accomplishment of the best clusterings quality in comparison with others.

1 Introduction

Clustering is the process of organizing unbalanced data and objects into groups in order to reveal their relationship and structures. This idea is found in many different fields such as machine learning, pattern recognition, biology, ecology, social sciences, marketing and psychology. The aim of clustering is to divide a set of objects into clusters that objects within clusters, have the most similarity between each other and the most dissimilarity with objects of other clusters [1, 2, 3, 4]. One of the most important problems in clustering is to detect proper number of clusters, especially when the clusters overlap with others, or when data have higher dimensions [4]. Up to now, a variety of methods have been suggested to detect the appropriate number of clusters. Gordon divided these methods into two categories: local and global methods. The local methods are planned to test the hypothesis that a pair of clusters should be amalgamated. They are suitable for assessing only hierarchical nested partitions. In fact, in local methods, selecting the proper number of clusters is equivalent to deciding in which level to cut the tree. Just as Gordon indicated that determining the number of clusters in the local methods is complex because of the multiple strict tests involved in the lives for merging clusters [5, 6]. On the other hand, in global methods, the number of clusters is required as an input. The quality of clustering with a specific number of clusters is measured by a criterion and the optimal estimate of the number of clusters is obtained by comparing the values of the criterion calculated with a range of values of k. However, the computational load to specify the number of clusters by artificial comparison will dramatically increase when the data set is large. Since partitioning clustering algorithms are iterative, inappropriate selection of initial partition and the criterion function leads to converging to local minimum and incorrect clustering results [6].

Currently, Evolutionary Algorithms are widely applied in different optimization problems. Especially, clustering is done by different evolutionary approaches such as evolutionary programming, genetic algorithms, particle swarm optimization, ant colony algorithms [7] and bee colony algorithms [8]. Dong et al. proposed a fuzzy clustering based on evolutionary programming. They could improve fuzzy c-means with the benefits of the global search strategy of evolutionary algorithms. During the algorithm, the number of clusters changes dynamically to find the proper number of clusters [9]. Some of researchers applied genetic algorithms for clustering. Liu et al. [10] developed a genetic algorithm to automatically detect the number of clusters. They applied a variable-length genetic algorithm and used Davies-Boldin index [11]

as a fitness function. They also designed a noising selection and division-absorption mutation to keep the balance of population. Dongxia et al. [12] designed a dynamic niching genetic algorithm with data attraction to preserve the diversity of the population. In fact, in each generation, niches dynamically evolve an appropriate number of clusters by data attraction, without using cluster validity function or a variance-covariance matrix. Chang et al. applied a genetic algorithm with variable-length chromosomes that uses a message-based similarity measure. The algorithm updates the clusters centers by means of message exchange between data points and candidate centers, so the algorithm is able to detect the appropriate number of clusters [13]. In order to do an automatic clustering, He and Tan [6] designed a two-stage genetic algorithm. First, they searched for the proper number of clusters and then they transferred to global optimal cluster centers. They also overcame the sensitivity of clustering algorithms to initial partition by means of a maximum attribute range partition approach. Integration of particle swarm optimization and genetic algorithm was applied by Kuo et al. for dynamic clustering. The algorithm can automatically cluster data without a pre-specified number of clusters. The algorithm tries to increase global search capabilities and escapes from the local optimum. The results showed its validity and stability [14]. Gene transposon based clone selection algorithm is the name of the algorithm proposed by Liu et al. for automatic clustering. The algorithm was a new immune computing algorithm which evolves a number of clusters and cluster centers (variables) to satisfy values based on clonal selection. The length of antibody string for finding the proper number of clusters is changed by a genetic operation [15].

The Grouping Genetic Algorithm (GGA) is a class of evolutionary algorithms that was introduced to overcome perceived drawbacks of traditional GAs with respect to grouping problems (problems in which a number of items must be assigned to a set of predefined groups) [16, 17]. The GGA is somehow different from standard GAs. In the GGA, the encoding, crossover and mutation operators of traditional GAs are modified to be applied in grouping problems with a high performance. The GGA has been successfully applied to a number of problems in different fields such as telecommunications [18], manufacturing [19], and industrial engineering [20, 21, 22, 23].

Clustering is one of the grouping problems which, recently, for the first time, has been done with GGA by Agustin-Blas et al. [24]. They did a successful clustering which was also able to detect a proper number of clusters in most cases. But during crossover operation, some of the objects were randomly assigned to clusters which led to the sudden decrease of fitness value and disability of crossover to improve solutions. While the mutation operation was performed, they divided a cluster into two. They did it randomly which led to an inappropriate clustering and wouldnt create any improvements. Since an elitist replacement was used, a local search and an island model, the algorithm was able to make a successful clustering in most cases.

Stochastic methods, such as genetic algorithms perform a more comprehensive search of the model space in comparison to deterministic search techniques. However, it gets difficult or time consuming for all the model parameters to converge

within a given margin of error. One of the most crucial tasks of an evolutionary algorithm is Scalability, it is how deal with complex search space resulted from high dimensionality. A number of enhanced EAs have been introduced that attempts at manipulating the search space, not necessarily aiming at scalability issues thought. Some of the major improved variants on traditional EA are based on GA [25]. Some of GA based methods tried to have a balanced and diverse population [26, 27]. These methods avoided redundancy by replacing similar individuals with different individual in localized search spaces. Tsutsui [28] also aimed at working on genetically different populations. Some others used special learning-based approaches by applying differential evolution implementation [29]. Application of an efficient grouping genetic helps in building diverse and balanced population. Indeed, this special approach could be a new solution for clustering of big data sets.

In this chapter the GGA is applied for clustering with new crossover and mutation operators that overcome mentioned drawbacks. Clustering with different number of clusters were simultaneously carried out parallel to searching the optimal number of clusters, and finding the best part of the data. The proposed algorithm was tested on different artificial and real data sets and compared it with the other algorithms. Results indicated better performance in finding the proper number of clusters. Furthermore, the proposed clustering method presented in this article showed higher quality than the others.

The remainder of the chapter is organized as follows: next section expresses the problem definition. Section 3 fully demonstrates all steps of the proposed approach that consists of encoding, fitness function, selection operator, crossover operator, mutation operator, replacement and local search. Section 4 describes the applied clustering evaluation measure. Then, the experimental evaluation to identify a number of clusters, clustering accuracy and comparison between the proposed algorithm and the other clustering algorithms are presented in Section 5. Finally, conclusions are pointed to in Section 6.

2 Problem Definition

Clustering divides data into clusters that are meaningful, useful, or both. Clusters, or conceptually meaningful categories of objects that have familiar characteristics, play a significant role in how people analyze or describe the world. Essentially, human beings are skilled at assigning objects into groups. For instance, even young children can quickly label the objects of scene as buildings, vehicles, people, animal, plants, etc. Todays, researchers are trying to make machines to cluster objects like or better than humans. This made us (humans) to increase our understanding in different fields such as biology, psychology, medicine or business and etc., and be able to summarize or compress the date to achieve useful information [30, 31, 32, 33, 34]. Clustering divides data objects into groups only based on information found in the data that describe the objects and their relationships. The aim of clustering is to divide a set of objects into clusters with greater similarity within groups and greater differences between groups.

If U has partitioned with n objects or data points, $X = \{x_1, x_2, , x_n\}$, the aim of clustering is dividing these points to k clusters, $U = \{C_1, C_2, ..., C_k\}$,where C_i is the ith cluster and $C_1 \bigcup C_2 \bigcup ... \bigcup C_k = U$ and $C_1 \bigcap C_2 \bigcap ... \bigcap C_k = \phi$, in a way that the points within one cluster have the most similarity and the points in different clusters have the least similarity. To do so, a fitness function applied to demonstrate the quality of clustering. In fact, by calculating a fitness value for each solution, the quality of different solutions could be compared with each other. Determination of the best estimate of clusters' number is a fundamental problem in clustering. This could have a great effect on the clustering results. However, it is difficult to choose an appropriate number of clusters because of the lack of prior domain knowledge, especially when clusters are not well-separated or when the data has high dimensions [1].

The classical method of estimating clusters' number involves the use of some validity measures. The evaluation of a special validity function for clustering result is performed within a range of the values of cluster number for each given cluster number and then an optimal number is chosen. The clusters' number search by this method depends on the selected clustering algorithm. Some other methods of determining the clusters' number are based on the idea of cluster removal and merging. In the progressive clustering, the clusters' number is over specified. After convergence, bogus clusters are eliminated and harmonic clusters are merged. The real challenge of this method is the definition of bogus and harmonic clusters. In these circumstances, there is no way to guarantee the chosen of proper clusters' number [35].

Since the global optimum of the validity would correspond to the best solution with respect to the functions, GA-based clustering algorithms have been reported to estimate the proper number of clusters. In these algorithms the validity functions are regarded as the fitness function to evaluate the fitness of individuals. Two examples of GA-based clustering were seen in [36, 37]. In both cases they try to estimate the proper number of clusters number, but they were making assumptions about the shape of data sets. So, when the data set violates the assumption will be unsatisfactory. In [38], a fuzzy GA was proposed for automatic detection of clusters' number without any assumption about the shape of data sets. This algorithm overcomes the problem of previous works, but when the clusters are overlapping, it prefers to class these clusters into one.

Despite the successful application of standard GA to lots of problems in various fields, as mentioned, there are some difficulties in applying GAs in grouping problems like clustering. First, the encoding pattern of the standard GA is not suitable for grouping problems. In fact, it is highly redundant for a grouping problem. For example, chromosomes CBBA and BAAC both have the same grouping which assign the second and the third elements in a one group together and the first and forth elements in two other separate groups. This kind of encoding pattern increases the dimensions of the search space and hampers the GA performance. Second, applying the standard GA operation will cause some problems. For example the crossover operation may produce offspring chromosome that have no quality. Furthermore, the application of standard mutation operator may be too destructive

when the GA nears a good solution. The Standard mutation operator will greatly reduce the quality of a solution by injecting new groups into a highly successful chromosome. The grouping genetic algorithm (GGA) was introduced by Falkenauer [16, 17] to solve these difficulties of applying GAs in grouping problems.

Stochastic search methods such as evolutionary algorithms are known to be better explorer of the search space in comparison to deterministic methods. However, in the area of big data analysis, suitability of evolutionary algorithms like most other search methods and learning algorithms is naturally questioned. Big data put new computational challenges, including very high dimensionality and sparseness of data. Evolutionary algorithms' superior exploration skills should make them promising candidates for big data analysis. High dimensional problems introduce added complexity to the search space.

The proposed model implements efficient grouping genetic operators to introduce diversity by expanding the scope of the search process and reducing less promising members of the population. Furthermore, the algorithm attempts to deal with the problem of high dimensionality of big data analysis by ensuring broader and more exhaustive search and preventing premature death of potential solutions.

Clustering has been done with GGA by Agustin-Blas et al. [24] for the first time. They did a successful clustering which was also able to detect a proper number of clusters in most cases. But during crossover operation, some of the objects were randomly assigned to clusters which led to the sudden decrease of fitness value and disability of crossover to improve solutions. While the mutation operation was performed, they divided a cluster into two. They did it randomly which led to an inappropriate clustering and wouldnt create any improvements.

In this chapter, an efficient GGA is applied for clustering with new crossover and mutation operators that overcome mentioned drawbacks of previous work. Clustering with different number of clusters were simultaneously carried out parallel to searching the optimal number of clusters, and finding the best part of the data. The proposed method without any predefinition of data set's shape is able to detect the proper number of clusters even in inseparable clusters. Indeed, it gains the best cluster number and accuracy of clustering with application of a new and GGA by use of an efficient crossover and mutation operators and the application of a local search.

3 The Proposed Algorithm

The GGA is different from the standard GA in several ways. First, the chromosomes consist of two parts, an element part and a group part. The group part also has a variable length in each chromosome. Second, the genetic operators work with the group section of the chromosome, altering the composition of the groups. This leads to, alteration of the main chromosome. Third, the GGA operators of crossover and mutation do not work in the same direction as the classical operators. In this chapter, an efficient GGA for clustering was applied. The general framework of the proposed algorithm is shown in Fig. 1.

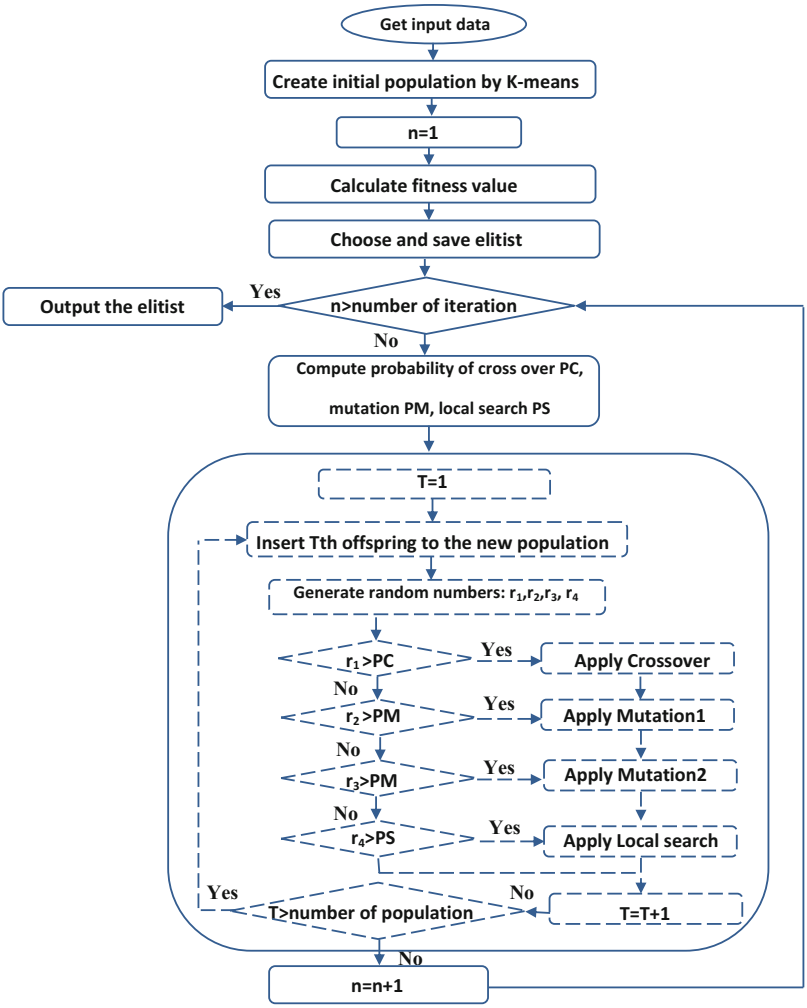
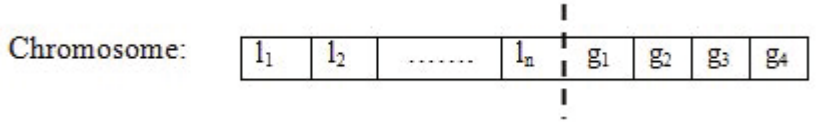


Fig. 1 General frame work of proposed algorithm

3.1 Encoding

A variety of encoding patterns have been proposed based on the characteristics of problems in GAs. Falkenauer presented a special encoding pattern of grouping problems. His encoding pattern consists of two parts: the object part and the group part.

The proposed GGA for clustering in this chapter is similar to the classical one initially proposed by Falkenauer. Each chromosome in the encoding of this grouping genetic algorithm is a representation of a solution for clustering of a fixed data set. In fact, each chromosome is a different solution for a specific problem. The chromosome is made of two parts; the first part is the object section, whereas the second part is called the group section. As mentioned, the solution of a clustering problem is going to divide a set of data points into separated groups. The first part of the chromosome is a representation of this data point and each record of this part holds the value of group which the object is assigned. The group part is a representative of the groups of that solution. Since the number of data points of a special problem is fixed, all the chromosomes of the population have object parts with the same length. The length of group part of each chromosome is variable and equal to the number of its groups. A simple example of a chromosome for a clustering problem with n objects and k clusters is as follows:



L_j represents the cluster to which the j_{th} object is assigned and g_i is indicator of i_{th} cluster. In a formal way:

$$l_j = gi \Leftrightarrow x_j \in C_i$$

As you see each part of object section is related to one of the objects of the data set and the length of object section is fixed and equal to the number of data set objects. The length of group section for each chromosome is variable and depends on the number of clusters of that solution (chromosome), because the algorithm doesnt receive the number of clusters as an input and finds the proper clusters number itself. Therefore, the algorithm tests different solutions with different number of clusters in order to find the best number of clusters.

To clarify the GGA encoding, you can see a sample chromosome in Fig. 2. The solution consists of four clusters, so the length of group part is four and since the objects number of the data set is 17, the length of object part is 17. Note that all the chromosomes for finding the best clustering for this data set have an object part by the length of 17, but the length of group part differs from one to another. The initialization of the algorithm is done in two steps:

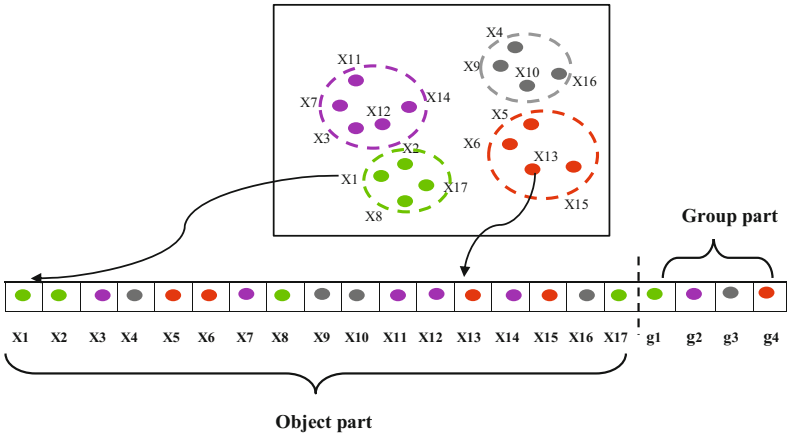


Fig. 2 Representation of a sample chromosome

- First, it chose a random number for each chromosome as the number of clusters. In most of the algorithms, the number of clusters is a random number chosen from the interval of 2 to $\sqrt{n}$, for example [6]. The same interval was tested at first, but then was replaced $\sqrt{n}$ by $\lceil \sqrt{\frac{n}{2}} \rceil$ since it leads to better results. Under this condition, maximum number of clusters in a data set including 50 elements will be 5, and it seems to be acceptable. Under this condition, not only better results were achieved, but also running time was greatly reduced.
- Second, it prepared the object part of initial population by means of K-means algorithm. Fig. 1 shows a K-means clustering, which is implemented on a chromosome (In real situations, usually number of clusters are used instead of clusters colors). In fact, the initial clustering was done by K-means and then it improved during the algorithm. At the end, the chromosome, which had done the best clustering, will be chosen.

3.2 Fitness Function

A fitness function is defined to be able to evaluate the quality of each chromosome. To calculate the fitness value, it should have a distance measure. In traditional clustering problems, the most popular used distance is given by a norm defined by a symmetric and positive matrix A:

$$d^2(x_i,x_j) = \|x_i - x_j\|_A = (x_i - x_j).A.(x_i - x_j)^T \quad (1)$$

Where T stands for transpose operation. In fact matrix A defines the shape and the size of the set of vectors sited at a distance of a given vector under study x_i. The most popular form of a norm A distance is the simplest case when $A = I$, in this situation the generated distance is called Euclidean distance and is defined as follows:

$$d^2{}_E(x_i,x_j) = \|x_i - x_j\|^2 = (x_i - x_j).(x_i - x_j)^T \tag{2}$$

The Mahalanobis distance is a suitable alternative for cases in which not all the clusters are ellipsoids with the same orientation and size. The definition is as follows:

$$d^2{}_M(x_i,x_j) = \|x_i - x_j\|_{\sum^{-1}} = (x_i - x_j).\sum^{-1}.(x_i - x_j)^T \tag{3}$$

Where$\sum$ stands for the covariance matrix, The Mahalanobis distance considers correlations between the different features of the objects involved in the distance.

After choosing the distance measure, its time to choose the fitness function. One of the most commonly used fitness functions for genetic clustering algorithms consists of minimizing the sum of the squared Euclidean distances of the objects from their respective cluster centers. Nevertheless, this approach may cause k to be larger than practical value due to the monotonically minimization of the distance between each point and the cluster center [39]. It is known that many clustering validity criteria have also been utilized as objective functions. These days, there are several validity indices; including Calinski-Harabas index, Davies-Bouldin index [11], Maulik-Bandyopadhyay index, Dunn index and Silhouette index [40].

Saitta et al. [40] say Dunn index is heavily computational and has problem in dealing with noisy data. It is beneficial for identifying clean clusters in small and medium data sets. Davies-Bouldin index presents good results for distinct groups. However, it is not designed to cope with overlapping clusters. The Maulik-Bandyopadhyay index has the particularity of being dependent on a user specified parameter. So, the Silhouette index was chosen. The Silhouette index is based on a combination of the internal cluster cohesion, a_j and the external cluster isolation, b_j. By using the Silhouette index as a fitness function, the algorithm was able to evaluate the quality of a particular solution, and also the quality of each of the clusters. Even more, it allowed evaluating clustering accuracy of each particular object x_i. Thus, the silhouette index is defined for the j_{th} object x_j,

$$s_j = \frac{b_j - a_j}{max(a_j, b_j)} \tag{4}$$

Where the parameter a_j is the average distance between j_{th} object and other objects in its cluster and b_j is the minimum average distance between j_{th} object and other objects in each of the other clusters. The silhouette coefficient for a given cluster C_j is defined in the following way:

$$S_j = \frac{1}{n(C_j)} \sum_{x_j \in C_j} s_j \tag{5}$$

Here $n(C_j)$ is the number of objects that are in the j_{th} cluster. At the end the silhouette coefficient for a given partition U of the data is:

$$S(U) = \frac{1}{k}\sum_{j=1}^{k} S_j \tag{6}$$

In this definition, $S(U)$ is in the interval between [-1, 1], those numbers which are near to -1 indicate a wrong clustering while numbers near to 1 are interpreted as proper clustering and the objective for a good partition is to maximize S.

3.3 Selection Operator

In the proposed algorithm, a rank-based wheel selection mechanism was used as the selection method, like the one described in [19]. At the beginning, the chromosomes are sorted in a list based on their fitness values. The position of each chromosome in the list is called the rank of the chromosome, and denoted $R_i, i = 1, ..., \xi$, where is the number of chromosomes in the population. The algorithm considered a rank in which the best chromosome x is assigned $R_x = \xi$, the second best y, $R_y = \xi - 1$, and so on. A selection value associated with each chromosome is then defined, as follows:

$$f_i = \frac{2.R_i}{\xi.(\xi+1)} \tag{7}$$

Depending on the position of the chromosomes in the ranking list, the values are normalized between 0 and 1. This rank-based selection mechanism is static, considering that probability of survival (given by f_i) do not depend on the generation, but on the position of the chromosome in the list. As an example, suppose a population consisting 5 chromosomes ($\xi = 5$), in which chromosome 1 is the best quality one ($R_1 = 5$), chromosome 2 is the second best ($R_2 = 4$), and so on. For example, the selected value of the best chromosome is calculated as follows:

$$f_1 = \frac{2.R_1}{\xi.(\xi+1)} = \frac{2\times 5}{5.(5+1)} \tag{8}$$

In this case, the fitness related to the chromosomes are {0.33, 0.26, 0.2, 0.13, 0.06 }, and the associated intervals for the roulette wheel are {0 - 0.33, 0.34 - 0.6, 0.61- 0.8, 0.81 - 0.93, 0.94- 1 }.

3.4 Crossover Operator

The GGA crossover operator described in [17], unlike traditional ones, operates only on the group part of the chromosome, and then, according to the alteration of group part, the object part will be updated. As mentioned previously, the group

part of the solution contains the identifiers of the groups to which the objects are assigned. GGA Clustering has been recently done by Agustin-Blas et al. [24] for the first time; they chose several clusters of both parents and then the rest of the objects were randomly assigned to one of the chosen clusters. Doing so has a bad effect on the quality of clustering and decreased the value of the fitness function suddenly. In this chapter, two approaches are presented to overcome this problem, the first approach repairs the solutions after crossover, but the problem becomes so time-consuming. The second approach uses another crossover operator such as one which was used as follows and was implemented in [19]. The crossover operator process is as follows:

1. First, select two chromosomes by the selection operator, and then choose two crossing points in the first parent group part.
2. Insert the objects belonging to the selected groups of the first chromosome into the offspring, and the selected group numbers into the group part of the offspring.
3. Insert the rest of the objects with their clusters number in the second parent, and add all the group numbers of the second chromosome in the group part.
4. Remove empty clusters, if any (some of the clusters of the second parent may be empty).
5. Modify the labels of the current groups in the offspring in order to numerate them from 1 to k.

Fig. 3 shows an example of crossover procedure implemented in this chapter. As you see, first two parents are selected and two randomly selected crossing points are shown with $\downarrow$, in this situation the gray cluster is chosen, so in step two, the assigned objects to gray cluster are inserted into the offspring and the gray cluster is added to the group part. In the next Step, the rest of the objects were inserted from parent 2 into their color in parent 2 and all clusters of parent two were also added to the group part of the offspring. Since some of the clusters of parent two might be empty, they were omitted in this step, although no empty cluster would exist here. In this example, since the clusters colors of parents were different nothing would be done in step 5. Finally, you can see the offspring in Fig. 3. Note that colors were used as identifiers of the clusters for better perception of the problem, but in real conditions numbers will be used as identifiers.

An adaptive crossover probability was implemented in order to apply crossover operator with high probability in the first stages of the algorithm, and moderated probability in the last ones in order to properly explore the search space. Thus, it is defined in the following way:

$$P_c(j) = P_{ci} + \frac{j}{TG}(P_{ci} - P_{cf}) \tag{9}$$

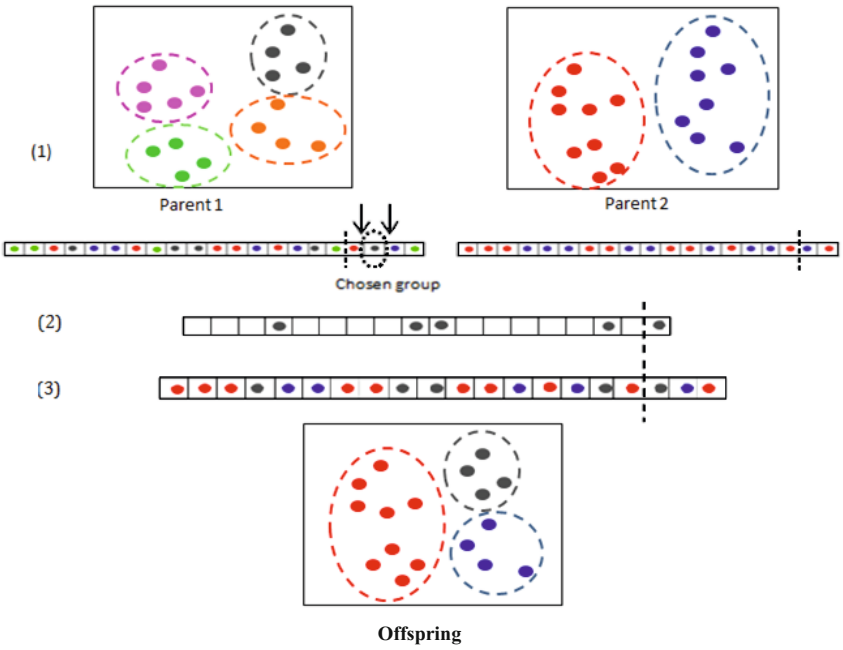


Fig. 3 An example of crossover operation

Where $P_c(j)$ is the crossover probability applied in the j_{th} iteration, TG stands for the total number of iterations of the algorithm, $P_{ci}$and P_{cf} are the initial and final values of probability considered, respectively.

3.5 Mutation Operators

With a standard GA, the mutation operator generally involves the alteration of a small percentage of randomly selected genes. For grouping problems, this type of mutation may be too disruptive. If an item is transferred to a new group which it shares no similarities with its members, then the quality of the mutated solution may be greatly impacted. Falkenauer proposed mutation strategies which involve creating a new group, eliminating an existing group, or shuffling a few items among their respective groups [17]. In this chapter the mutation process is made of two mutation operators that are independent of each other.

- Mutation by creating new clusters: In this part a cluster was divided into two clusters. The selection of the initial cluster to be divided is dependent on the clusters size, with more probability of division given to larger clusters. To do so, the new generated cluster will keep its label in the group part of the chromosome, whereas the other will be assigned a new label $(k+1)$. Agustin-Blas et al. [24]

assigned the observations to the new cluster with equal probability. In fact objects are randomly divided between two clusters. In this situation, it wont be useful to compare the quality of clustering with other chromosomes in order to be a chance of local optimum. This problem was solved by applying the K-means algorithm as the divider of the objects. Fig. 4 shows an example of mutation by creating new cluster. In the beginning an identifier was chosen for the new cluster, here green color is the identifier of the new cluster, and added it to the group part. Then the chromosome was updated by changing the label of the objects which were assigned to the new cluster.

- Mutation by merging clusters: after choosing two clusters using this operator, they are merged and converted into one. As in mutation by creating new cluster, the probability of choosing the first cluster depends on the size [24] used such method for mutation, but they chose both clusters according to their size and didnt pay any attention to other important factors such as the location of the clusters. So they might choose two clusters with completely different locations which may lead to a low quality clustering. Here a method was applied to resolve this problem. After the selection of the first cluster, its time to choose the second one. In order to find the best cluster to merge with the first one, combinations of the first cluster with each of the other clusters were tested and then the fitness value of each solution was calculated. At the end, the cluster (the second cluster) which led to the best fitness value was chosen.

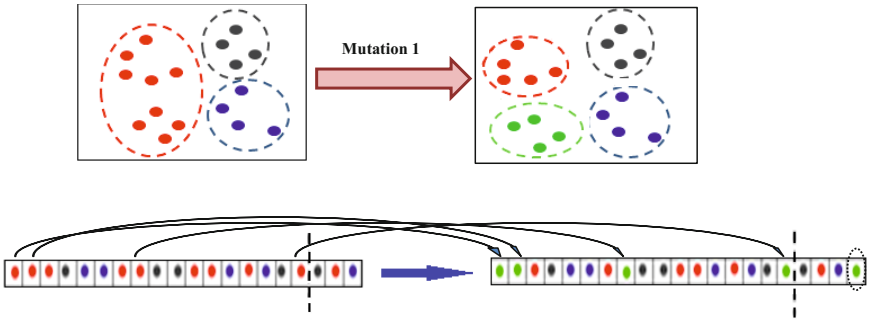


Fig. 4 An example of mutation by creating new cluster

After clusters selection, the identifier of one cluster was deleted from the group part and turned the identifier of objects belonging to the cluster to the other chosen cluster. If numbers were used as identifiers, the number of larger ones could be deleted from the group part and turn the number of objects belonging to the cluster to the other chosen cluster. Fig. 5 shows the process of clusters merging. As you see the red and green clusters are the selected clusters. In the next step, the identifier of green cluster was omitted from group part and assigned the objects of green cluster to the red one. Then the new solution is shown.

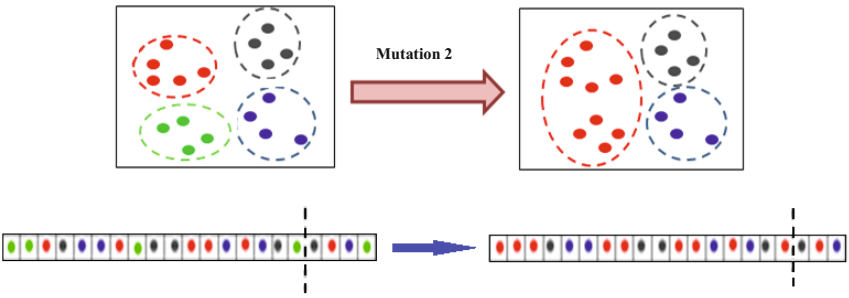


Fig. 5 An example of mutation by merging clusters

Again, an adaptive mutation probability was implemented in order to apply mutation operators. The two mutation operators were implemented respectively, with independent probabilities of the application. In this case, probability of mutation is smaller in the first iterations of the algorithm and larger in the last ones, in order to have more opportunities to escape from local minimums in the last stages of the algorithm; the mutation probability is as follows:

$$P_m(j) = P_{mi} + \frac{j}{TG}(P_{mf} - P_{mi}) \tag{10}$$

Where $P_m(j)$ is the mutation probability of a given iteration j, TG stands for the total number of iterations of the algorithm and P_{mf} and P_{mi} are the final and initial values of probability considered, respectively.

3.6 Replacement and Elitism

In the proposed algorithm, the population of a given generation $j+1$ is obtained by the application of the selection crossover, and mutation operators described above by the replacement of the chromosomes in the population at iteration j. An elitist selection method is also applied; the best chromosome in iteration j is automatically passed on to the population of iteration $j+1$, ensuring that the best solution obtained so far is always preserved by the algorithm.

3.7 Local Search

The proposed algorithm uses a local search procedure to try to find local optimums in a close neighborhood of a chromosome. The proposed local search is based on slight modifications of the current chromosomes, as far as they produce an increase in the associated fitness values. Here for each object of the chromosome, the fitness value of the solution was calculated while the object is assigned to each cluster and then the cluster, which gains the highest fitness value will be chosen as the object to be assigned, although the best cluster may be the one which the object

belongs to at first. An example of this process is shown in Fig. 6. As you see in Fig. 6, the object X_3 was assigned to each of the clusters that were in the group part, and then the fitness function of each of these solutions was calculated and showed by $a_1, ..., a_4$. Afterwards by comparing these values, the maximum value was chosen and assigned X_3 to that cluster. Note that this was done for all the objects of the chromosome. Since this is a quite time-consuming operation, it is applied to a given chromosome with a small probability.

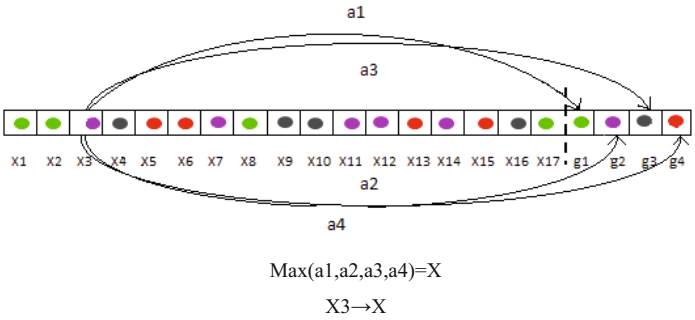


Fig. 6 Example of local search

4 Validation of Clustering

Sometimes there are available external (known) results in the clustering problem that can help evaluate the quality of algorithms. Here the Rand index was chosen as the evaluation function. The Rand index or Rand measure [41] in statistics, and particularly in data clustering, calculates the similarity between the obtained partition (the result of the algorithm) and the known optimal solution, i.e. it is a measure of the percentage of the correct decisions made by the algorithm. From a mathematical standpoint, Rand index is related to the accuracy, but is applicable even when class labels are not used. Given a set of n elements, $S = \{o_1, ..., o_n\}$ and two partitions of S to compare , $X = \{x_1, ..., x_r\}$, a partition of S into r subsets, and $Y = \{y_1, ..., y_s\}$, a partition of S into s subsets, defines the following:

- a, the number of pairs of elements in S that are in the same set in X and in the same set in Y
- b, the number of pairs of elements in S that are in different sets in X and in different sets in Y
- c, the number of pairs of elements in S that are in the same set in X and in different sets in Y
- d, the number of pairs of elements in S that are in different sets in X and in the same set in Y

The Rand index (RI) is:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \tag{11}$$

Intuitively, a+b can be considered as the number of agreements between X and Y and c+d as the number of disagreements between X and Y. The value of RI is 1when two partitions match perfectly and 0 when the two partitions are selected at random.

5 Experiments and Evaluation

In this section, the experimental results of both artificial and real data sets are presented and then they are compared with three other algorithms. The results consist of both detecting proper number of clusters and the quality of clustering solutions. In the first part the data sets are introduced and then in the second part the efficiency of the algorithm will be proved.

5.1 Data Sets

The experiments were implemented on three artificial data sets and two real data sets from UCI Repository. The chosen data sets are equal to those used in Agustin-Blas et al. [24] for comparison.

Artificial data sets:

Data set 1: spherical clusters

The first data set is a 2-dimensional clustering problem, defined by 300 objects, randomly generated by using a Gaussian distribution from 8 classes with equal probability, with the following means and covariance matrices:

$\mu_1 = (1,-3), \mu_2 = (-2,-1), \mu_3 = (-1,1), \mu_4 = (1,-2), \mu_5 = (1,0), \mu_6 = (3,-3),$
$\mu_7 = (2.7,-1), \mu_8 = (3,1)$

$$\Sigma_1 = ... = \Sigma_8 = \begin{bmatrix} 0.35^2 & 0 \\ 0 & 0.35^2 \end{bmatrix}$$

Data set 2: structured clusters

The second data set is again another 2-dimensional clustering problem, defined by 400 objects, randomly generated by using a Gaussian distribution from 3 classes with probabilities, $p_1 = 0.5, p_2 = 0.33$ and $p_3 = 0.17$. The classes means are: $\mu_1 = (-1,-1), \mu_2 = (2,-1)$ and $\mu_3 = (0,2)$, and the covariance matrices are as follows:

$$\Sigma_1 = \begin{bmatrix} 1^2 & 0 \\ 0 & 0.8^2 \end{bmatrix} \Sigma_2 = \begin{bmatrix} 0.6^2 & 0 \\ 0 & 0.4^2 \end{bmatrix} \Sigma_3 = \begin{bmatrix} 0.3^2 & 0 \\ 0 & 0.5^2 \end{bmatrix}$$

In this data set, classes have different distributions and are not spherical.

Data set 3: unbalanced clusters

The final artificial data set is a 2-dimensional clustering problem, defined by 200 objects, randomly generated by using a Gaussian distribution from 9 classes with equal probabilities, with means: $\mu_1 = (1,-1), \mu_2 = (-1.5,0), \mu_3 = (0,1),$

Table 1 Main features of data sets

Data set	M	P	K
Spherical data	300	2	8
Structured data	400	2	3
Unbalanced data	200	2	9
Iris	150	4	3
Wine	178	13	3

Table 2 An example for unbalanced data set

	1	2	3	4	5	...	196	197	198	199	200
X	0.9296	0.9734	1.0343	0.9818	-1.5014	...	1.4876	1.8225	1.2526	1.7133	1.6211
Y	-0.9359	-1.0430	-0.7663	-0.8906	-0.2232	...	-0.3381	0.0053	0.1613	-0.0246	-0.1919

$\mu_4 = (-1,1), \mu_5 = (2,-1), \mu_6 = (-2,-1), \mu_7 = (-0.5,2), \mu_8 = (-1,-1), \mu_9 = (1.5,0)$ and the covariance matrices are as follows:

$$\Sigma_1 = ... = \Sigma_9 = \begin{bmatrix} 0.2^2 & 0 \\ 0 & 0.2^2 \end{bmatrix}$$

In this data set, there are three groups of clusters form different clusters structures, that two of them overlapping. The aim is to check out that the algorithms are able to correctly separate all the cases.
Real data sets:

- Iris: This is maybe one of the best known data sets found in the clustering literature. It consists of 150 data points divided into three clusters. Each cluster has 50 points. The data set represents different categories of iris described by four feature values in centimeters: the sepal length, sepal width, petal length and the petal width. This data set has three classes, namely, Setosa, Versicolor and Virginica among which the last two classes overlap in large amounts while the first class is linearly separable.
- Wine: The wine recognition data consists of 178 instances with 13 features. It is formed by three classes that consist of 59, 47 and 78 data objects and each observation presents a class of wines from different regions of Italy. The analysis provided the quantities of 13 features corresponded to chemical attributes.

Main features of the data sets are shown in Table1. *M* is size, *P* is the number of attributes (dimension of data) and *K* is the real number of clusters of data sets.

At this point, the implementation of the algorithm on one of the datasets will be described step by step. The unbalanced data was chosen, since it's two dimensional and has fewer data points. After the application of Gaussian distribution function with mentioned circumstances, the dataset will be like the one in Table2.

The size of the population and the number of iterations could be inputs of the algorithm. In this example, the number of population was 30 ($n = 30$) and the number of iterations was 10 ($j = 10$). In the beginning, the initial population will be built. Each of the chromosomes is a clustering solution for this data set. The initial population in the first iteration will be built by application of k-means clustering. For each chromosome, a random number between 2 to 10 ($\lceil \sqrt{\frac{200}{2}} \rceil$) will be chosen. Then the k-means will be called and this random number is the input of k-means as the number of clusters. The answer of k-means forms the object part of the chromosome. Now the first chromosome is built. This procedure will repeat until the building of whole populations. In these circumstances, the length of group part of each chromosome will be equal to this random number (number of clusters) and since the number of data points is 200, the length of object part of each chromosome will be 200 too. In the next step, the fitness value of each chromosome will be calculated and then the best chromosome with highest fitness value will be saved as elitist. Now is the time to build the new population, the elitist of previous population will be the first offspring of the new population. The next offsprings will be chosen by the selection operator. After the selection of each offspring the crossover and mutation operators and a local search will be applied by a special probability as mentioned in previous sections. So, one offspring may be changed by all the mentioned operators or sent to the new population without any change. In this example, the initial probability for the application of crossover operator was 0.7, the initial probability for the application of the first mutation operator was 0.08, the initial probability for the application of the second mutation operator was 0.01 and the initial probability for the application of local search was 0.08. As mentioned, these probabilities will be changed for each new population. Finally, after the production of the last population, its' elitist will be the main solution of clustering for the desired dataset. As in this example the elitist chose 9 clusters for the data set and its rand index was 0.9978.

5.2 Results

First part of the results is related to detecting proper number of clusters. To verify the search ability of a number of clusters of the algorithm (EGGAC), the proposed algorithms results was compared with k-means, fuzzy c-means and GGAC in Table 3. Since k-means and c-means are not able to detect a number of clusters, the related columns are empty. Quantities of GGAC column are related to the results of Agustin-Blas et al. [24] algorithm by using S index as fitness function. The experiments results are shown in the last column. The proposed algorithm showed better results than the others. Note that the proposed algorithm had the same result in all run times, so the correct ratio of detecting proper number of clusters is 100 for all data sets.

For evaluating a clustering method, it is necessary to define a measure of agreement between two partitions of one dataset. In the clustering literature, measures of agreement between partitions are referred to as the external indices. Several such

indices are rand index, adopted rand index, Jaccard index and F-measure. In this chapter, the rand index was used as the evaluation method which was described in section 4. In order to evaluate the clustering accuracy of EGGAC, the mean value of rand index (RI) of the algorithm was recorded for 20 run times. The k-means and fuzzy c-means were also applied to all of the datasets. The number of clusters (k) which is a real number, was given as an input, in these cases.

Table 3 Result or detecting cluster numbers by of K-means, C-means, GGAC, EGGAC

Data set	numbers of clusters	K-means	C-means	GGAC	EGGAC
Spherical data	8	-	-	7	8
Structured data	3	-	-	3	3
Unbalanced data	9	-	-	9	9
Iris	3	-	-	3	3
Wine	3	-	-	3	3

Finally the results were compared with the results of GGAC in Agustin-Blas et al. [24]. The results of clustering accuracy of four algorithms are shown in Table 4.

Table 4 Result or detecting cluster numbers by of K-means, C-means, GGAC, EGGAC

Data set	K-means	C-means	GGAC	EGGAC
Spherical data	0.9493	0.9555	0.9578	**0.9984**
Structured data	0.9401	0.9408	0.9511	**0.9645**
Unbalanced data	0.9427	0.9658	0.9936	**0.9978**
Iris	0.8805	0.8805	**0.8995**	0.8939
Wine	0.7037	0.7122	0.7220	**0.7943**

As you see, clustering accuracy of NCGGA excels others. The values of RI of EGGAC in all data sets are greater than those of others except in iris data set. Although in this case the best result when $S(U) = 0.6330$ and $RI = 0.9341$ is better than the one which was gained by GGAC when S(U)=0.5325. It is worth mentioning that the best result for wine data set occurred when $S(U) = 0.4983$ and $RI = 0.8247$. Since wine data set has the highest dimensions among others and the results of the proposed algorithm tested on it have more improvements than others, It would be a good idea to apply the algorithm to other high dimensional data sets. Finally, you see the clustering results on 2 dimensional data sets in Fig. 7.

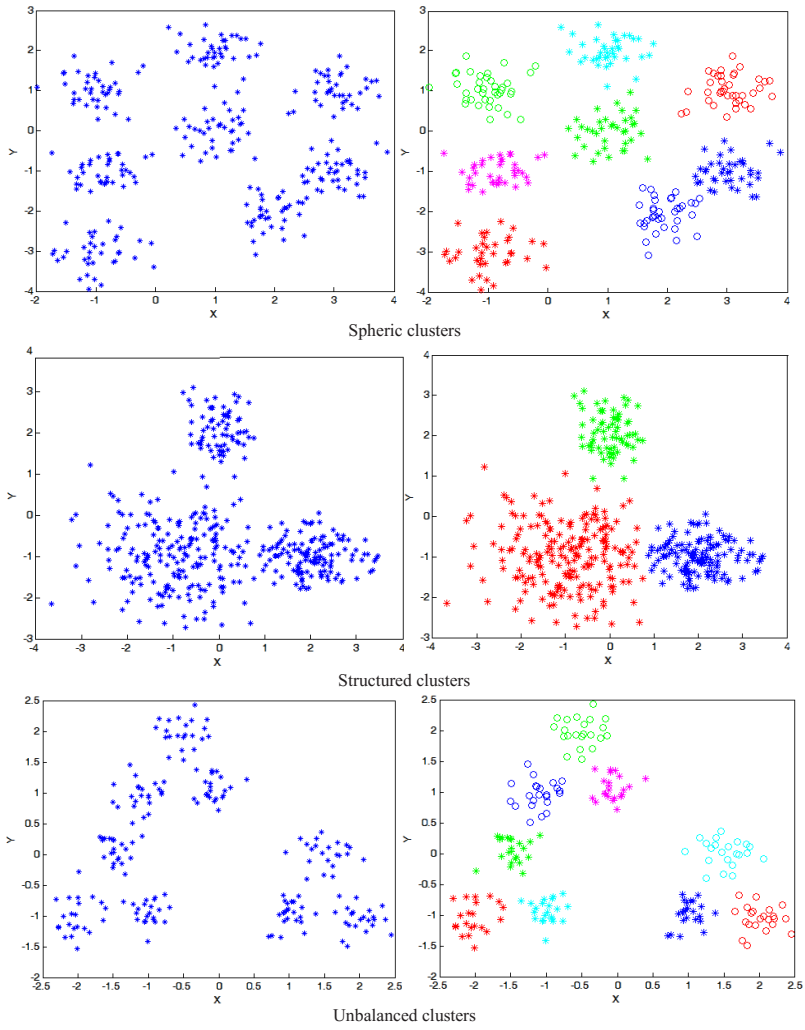


Fig. 7 Results of clustering on three artificial data sets

Up to here, the results of the proposed algorithm on different real and artificial data sets were seen. In the case of big data analysis, no testing has been done, although the highest improvement in wine data set with highest dimensionality. However, there are reasons which will prove the applicability of the proposed algorithm on this kind of data sets. First, the grouping genetic operators unlike traditional GAs works with the grouping part of chromosomes, so it will reduce the time complexity. Second, the grouping genetic operators do not work as classical GA operators, in fact their proper modification in the number of clusters and accuracy of clustering led to production of diverse and balanced population. Third, the application of local search helps in producing new and different chromosome, while the answers near to a local optimum and chromosome became similar to each other.

At the end there are two suggestions for application of the proposed algorithm on real big data sets. First, it's better to use a feature selection and feature extraction method before the application of the proposed algorithm to cope with the curse of high dimensionality. Second, another kind of clustering algorithms which are suitable for clustering of high dimensional data sets could be applied for production of the initial population.

6 Conclusions

Clustering is the unsupervised classification of patterns into groups. The clustering dilemma has been referred in many perspectives mostly by researchers in many restraints as it reveals its broad petition. It has benefited as one of the steps in the experimental data analysis. Conversely, clustering is a combinatorial difficult problem, particularly when lacking entire data recognition, number of clusters should be determined. In this chapter, an efficient grouping genetic algorithm is proposed for clustering along with these stages: application of various numbers of clusters in a data set in order to find the suitable number of clusters, optimization of the algorithm by means of effective crossover and mutation operators, quality enhancement by implementation of the local search method. Experimental and analytical results of testing the algorithm on three artificial and two real data sets and its evaluation compared with others led to the best performance of the proposed algorithm from two aspects: quality of clustered data sets and search, number of clusters.

References

1. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recognition Letters 31(8), 651–666 (2010)
2. Jain, M.N., Murty, A.K., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–322 (1999)
3. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
4. Everitt, B.S., Landau, S., Leese, M.: Cluster Analysis, 4th edn. Arnold, London (2001)
5. Gordon, A.D.: Classification, 2nd edn., Boca Raton FL (1999)

6. He, H., Tan, Y.: A two stage genetic algorithm for automatic clustering. Neurocomputing 81, 49–59 (2012)
7. Jiang, H., Yi, S., Yang, L.J.F., Hu, X.: Ant clustering algorithm with k harmonic means clustering. Expert Systems with Applications 37(12), 8679–8684 (2010)
8. Zhang, C., Ouyang, D., Ning, J.: An artificial bee colony approach for clustering. Expert Systems with Applications 37(7), 4761–4767 (2010)
9. Dong, H., Dong, Y., Zhou, C., Yin, W., Hou, G.: A fuzzy clustering algorithm based on evolutionary programming. Expert Systems with Applications 36, 11792–11800 (2009)
10. Liu, Y., Wu, X., Shen, Y.: Automatic clustering using genetic algorithms. Applied Mathematics and Computation 218(4), 1267–1279 (2011)
11. Davies, D., Bouldin, D.: A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1(2), 224–227 (1997)
12. Dongxia, C., Xianda, Z.: Dynamic niching genetic algorithm with data attraction for automatic clustering. Tsinhua Science and Techonology 14(6), 718–727 (2009)
13. Chang, D., Zhao, Y., Zheng, C., Zhang, X.: A genetic clustering algorithm using a messagebased similarity measure. Expert Systems with Applications 392(2), 2194–2202 (2012)
14. Kuo, Y.J., Syu, R.J.A., Chen, Z.Y., Tien, F.C.: Integration of particle swarm optimization and genetic algorithm for dynamic clustering. Information Sciences 195, 124–140 (2012)
15. Liu, R., Jiao, L., Zhang, X., Li, Y.: Gene transposon based clone selection algorithm for automatic clustering. Information Sciences 204, 1–22 (2012)
16. Falkenauer, E.: The grouping genetic algorithm: Widening the scope of the gas. Belgian Journal of Operations Research, Statistics and Computer Science 33, 79–102 (1992)
17. Falkenauer, E.: Genetic algorithms and grouping problems. John Wiley & Sons, Inc., Chichester (1998)
18. Brown, E.C., Vroblefski, M.: A grouping genetic algorithm for the microcell sectorization problem. Engineering Applications of Artificial Intelligence 17(6), 589–598 (2004)
19. Keeling, K.B., James, T.L., Brown, E.C.: A hybrid grouping genetic algorithm for the cell formation problem. Computers and Operations Research 34, 2059–2079 (2007)
20. Hung, C., Brown, E.C., Sumichrast, R.T.: CPGEA: a grouping genetic algorithm for material cutting plan generation. Computers and Industrial Engineering 44 (4), 651–672 (2003)
21. Agustin-Blas, L.E., Salcedo-Sanz, S., Vidales, P., Urueta, G., Portilla-Figueras, J.A.: Near optimal citywide wifi network deployment using a hybrid grouping genetic algorithm. Expert Systems with Applications 38(8), 9543–9556 (2011)
22. Chen, Y., Fan, Z.P., Ma, J., Zeng, S.: A hybrid grouping genetic algorithm for reviewer group construction problem. Expert Systems with Applications 38, 2401–2411 (2011)
23. Martinez-Bernabeu, L., Florez-Revuelta, F., Casado-Diaz, J.M.: Grouping genetic operators for the delineation of functional areas based on spatial interaction. Expert Systems with Applications 39, 6754–6766 (2012)
24. Agustin-Blas, L.E., Salcedo-Sanz, S., Jimenez-Fernandez, S., Carro-Calvo, L., Del Ser, J., Portilla-Figueras, J.A.: A new grouping genetic algorithm for clustering problems. Expert Systems with Applications 39, 9695–9703 (2012)
25. Bhattacharya, M., Islam, R., Abawajy, J.: Evolutionary optimization:a big data perspective. Journal of Network and Computer Applications (2014)
26. Yannibelli, V., Amandi, A.: A deterministic crowdig evolutionary algorithm to form learning teams in a collabrative learning context. Expert Systems with Applications 39(10), 8584–8592 (2012)
27. Mahfoud, S.: Crowding and preselection revisited. Technical Report, Illinois Genetic Algorithm Laboratoty (1992)

28. Tsutsui, S., Ghosh, A.: A search space division in gas using phenotypic squares estimates. Information Sciences 109, 119–133 (1998)
29. Wang, H., Wu, Z., Rahnamayan, S.: Enhanced opposition-based differentioal evolution for solving high-dimentional continuous optimization problems. Soft Computing 15(11), 2127–2140 (2011)
30. Tan, P.N., Steinback, M., Kumar, V.: Introduction to data mining. Addison Wesley, USA (2005)
31. Kaur, H., Wasan, S., Al-Hegami, A., Bhatnagar, V.: A unified approach for discovery of interesting association rules in medical databases. In: Perner, P. (ed.) ICDM 2006, vol. 4065, pp. 53–63. Springer, Heidelberg (2006)
32. Kaur, H., Wasan, S.: An integrated approach in medical decision making for eliciting knowledge. Web-based Applications in Health Care and Biomedicine 7, 215–227 (2010)
33. Kaur, H., Chauhan, R., Wasan, S.: A bayesian network model for probabilistic estimation. In: Mehdi Khosrow, P. (ed.) Encyclopedia of Research and Information Technology, 3rd edn., pp. 1551–1558. IGI Global, USA (2014)
34. Chauhan, R., Kaur, H.: Predictive analytics and data mining: A framework for optimizing decisions with R tool. In: Tripathy, B.K., Acharjya, D.P. (eds.) Advances in Secure Computing, Internet Services, and Applications, pp. 73–88. IGI Global, USA (2014), doi:10.4018/978-1-4666-4940-8.ch004
35. Chang, D., Zhang, X., Zheng, C., Zhang, D.: A robust dynamic niching genetic algorithm with niche migration for automatic clustering problem. Pattern Recognition 43, 1346–1360 (2010)
36. Srikanth, R., George, R., Warsi, N.: A variable-length genetic algorithm for clustering and classification. Pattern Recognition Letters 16, 789–800 (1995)
37. Ghozeil, A., Fogel, D.B.: Discovering patterns in spatial data using evolutionary programming. In: Koza, J.R., Goldberg, D.E., Fogel, D.B., Riolo, R.L. (eds.) Genetic Programming, pp. 521–527. MIT Press, Cambridge (1996)
38. Bandyopadhyay, S., Maulik, U.: Genetic clustering for automatic evolution of clusters and application in image classification. Pattern Recognition 35, 1197–1208 (2002)
39. Chiang, M.M.T., Mirkin, B.: Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. Journal of Classification 27(1), 3–40 (2009)
40. Saitta, S., Raphael, B., Smith, I.F.C.: A comprehensive validity index for clustering. Intelligent Data Analysis 12, 529–548 (2008)
41. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Society 66, 846–850 (1971)

Self Organizing Migrating Algorithm with Nelder Mead Crossover and Log-Logistic Mutation for Large Scale Optimization

Dipti Singh and Seema Agrawal

Abstract. This chapter presents a hybrid variant of self organizing migrating algorithm (NMSOMA-M) for large scale function optimization, which combines the features of Nelder Mead (NM) crossover operator and log-logistic mutation operator. Self organizing migrating algorithm (SOMA) is a population based stochastic search algorithm which is based on the social behavior of group of individuals. The main characteristics of SOMA are that it works with small population size and no new solutions are generated during the search, only the positions of the solutions are changed. Though it has good exploration and exploitation qualities but as the dimension of the problem increases it trap to local optimal solution and may suffer from premature convergence due to lack of diversity mechanism. This chapter combines NM crossover operator and log-logistic mutation operator with SOMA in order to maintain the diversity of population and to avoid the premature convergence. The proposed algorithm has been tested on a set of 15 large scale unconstrained test problems with problem size taken as up to 1000. In order to see its efficiency over other population based algorithms, the results are compared with SOMA and particle swarm optimization algorithm (PSO). The comparative analysis shows the efficiancy of the proposed algorithm to solve large scale function optimization with less function evaluations.

1 Introduction

Self Organizing Migrating Algorithm (SOMA) is a stochastic population based algorithm based on the social behavior of a group of individuals presented by Zelinka

Dipti Singh
Department of Applied Sciences, Gautam Buddha University, Greater Noida, India
e-mail: diptipma@rediffmail.com

Seema Agrawal
Department of Mathematics, S.S.V.P.G. College, Hapur, India
e-mail: Seemagrwl7@gmail.com

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_6

and Lampinen in 2000 [1]. This algorithm is inspired by the competitive cooperative behavior of intelligent creatures solving a common problem. Such a behavior can be observed anywhere in the world. A group of animals such as wolves or other predators may be a good example. If they are looking for food, they usually cooperate and compete so that if one member of this group is successful (it has found some food or shelter) then the other animals of the group change their trajectories towards the most successful member. If a member of this group is more successful than the previous best one then again all members change their trajectories towards the new successful member. It is repeated until all members meet around one food source. Like other evolutionary algorithm it also works with a population of solutions. The main feature of this algorithm which makes it distinguished as compared to other algorithms is that no new solutions are created during the search. Instead, only the positions of the solutions are changed during a generation, called a migration loop. SOMA converges very fast for small scale problems but as the problem size increases its convergence becomes very slow and it may trap to local optima. It is because of poor balancing between exploration and exploitation. Exploration and exploitation are considered as two major characteristics of population based algorithms for maintaining the diversity of the population and to speed up the convergence.

To overcome the difficulty of premature convergence and to avoid the loss of diversity of population in search space one simple way is to combine population based algorithms with the features of other population based algorithms or hybridized them with local search algorithms. In this regard, many attempts have been made in past. Domingo proposed a real coded crossover operator for evolutionary algorithms based on the statistical theory of population distributions [2]. Chelouah and Siarry proposed a hybrid method that combines the feature of continuous Tabu search and Nelder-Mead Simplex algorithm for the global optimization of multi-minima functions [3]. Deep and Dipti proposed a hybridized variant SOMGA for function optimization which combines the features of binary coded GA and real coded SOMA [4]. Fan et al. hybridized Nelder-Mead Simplex method with Genetic algorithm and particle swarm optimization to locate the global optima of non-linear continuous variable functions [5]. Premalatha and Nataranjan established a hybrid variant of PSO that proposes the modification strategies in PSO using GA to solve the optimization problems [6]. Khosravi et al. proposed a novel hybrid algorithm by combining the abilities of evolutionary and conventional algorithm simultaneously [7]. Ghatei et al. designed a new hybrid algorithm using PSO and GDA, in which global search character of PSO and local search factor of Great Deluge Algorithm are used based on series [8] . Ahmed et al. proposed a hybrid HPSOM algorithm, in which PSO is integrated with genetic algorithm mutation method [9]. Millie et al. presented a variant of quantum behaved particle swarm optimization (Q-QPSO) for solving global optimization problems which is based on the characteristics of QPSO, and uses interpolation based recombination operator for generating a new solution vector in the search space [10]. Deep and Bansal developed a variant of PSO with hybridization of quadratic approximation operator for economic dispatch problems with valve-point effects [11]. Deep and Thakur proposed a new mutation operator

for real coded genetic algorithm [12]. Xing et al. developed a novel mutation operator based on the immunity operation [13]. Deep et al. proposed a new mutation operator for real coded genetic algorithms and its performance is compared with real coded power mutation operator [14]. Mohan and Shankar developed random search technique for global optimization based on quadratic approximation [15]. Deep and Das proposed a quadratic approximation based hybrid genetic algorithm for function optimization, in this paper they hybridized four GAs (GA1-GA4) by incorporating the quadratic approximation operator in to them [16]. Deep and Bansal presented the hybridization of PSO with quadratic approximation operator (QA), the hybridization is performed by splitting the whole swarm into two subswarms [17]. To improve the performance of real coded genetic algorithm Deep and Das hybridized it with quadratic approximation [18]. Millie et al. presented a new variant of particle swarm optimization named QPSO for solving global optimization problems [19]. There has not been done much work on hybridization of SOMA with other approaches except [4]. Recently Singh Dipti et al. presented a variant (SOMAQI) of SOMA, in which SOMA is combined with quadratic interpolation crossover in order to improve its efficiency for finding the solution of global optimization problems of small scale [20].

In this chapter, again a variant NMSOMA-M of SOMA has been proposed, in which SOMA is hybridized with NM crossover operator and Log-logistic mutation operator. The proposed algorithm, not only remove the difficulty of premature convergence of large scale function optimization but also maintain the diversity of the population. In this approach a new linear NM crossover operator has been designed to create a new member in the search space and has been used along with Log-logistic mutation operator to maintain the diversity of the populations during the search. Its efficiency has been tested on 15 scalable unconstrained test problems with problem size vary up to 1000 and a comparative analysis has been made between PSO, SOMA and proposed algorithm. The information about PSO is given in [21].

The chapter is organized as follows: In section 2, SOMA is described. In section 3, the methodology of proposed Algorithm NMSOMA-M has been discussed in detail. Test problems used for testing of the proposed algorithm has been listed in section 4. In section 5, numerical results of the present study have been discussed. Finally, the chapter concludes with Section 6 drawing the conclusions of the present study.

2 Self Organizing Migrating Algorithm

Self organizing migrating algorithm is relatively a new stochastic evolutionary algorithm which is based on the social behavior of a group of individuals [22]. Inspired by the competitive cooperative behavior of intelligent creatures, the working of this algorithm is very simple. At each generation the individual with highest fitness value is known as leader and the worst is known as active is taken into consieration. Rather than competing with each other, the active individual proceeds in the direction of the

leader. This algorithm moves in migration loops and in each migration loop, active individual travels a certain distance towards the leader in n steps of defined length. This path is perturbed randomly by a parameter known as PRT parameter. It is defined in the range of $< 0, 1 >$. A PRT vector is created using this PRT parameter value before an individual proceeds towards leader, known as perturbation vector. This randomly generated binary perturbation vector controls the allowed dimensions for an individual of population. If an element of the perturbation vector is set to zero, then the individual is not allowed to change its position in the corresponding dimension. To create this perturbation vector, following expression is used:

if $rnd_j < PRT$ then

$PRTVector_j = 1$

else

$PRTVector_j = 0$

The movement of an individual during the migration is given as follows:

$$x_{i,j}^{MLnew} = x_{i,jstart}^{ML} + (x_{i,j}^{ML} - x_{i,jstart}^{ML}) t PRTVector_j \tag{1}$$

Where $t \in < 0, bystepto, pathlength >$, ML is actual migration loop, $x_{i,j}^{MLnew}$ is the new positions of an individual, $x_{i,jstart}^{ML}$ is the positions of active individual and $x_{i,j}^{ML}$ is the positions of leader. The computational steps of SOMA are given as follows. The flow chart of SOMA process is depicted in Figure 1:

Algorithm (SOMA)

1. Generate initial population
2. Evaluate all individuals in the population
3. Generate PRT vector for all individuals
4. Sort all of them
5. Select the best fitness individual as leader and worst as active
6. For active individual new positions are created using equation (1). Then the best position is selected and replaces the active individual by the new one if it is better than active individual
7. If termination criterion is satisfied stop else go to Step-2
8. Report the best individual as the optimal solution

3 Proposed NMSOMA-M Algorithm

In this section a new hybrid variant of SOMA, NMSOMA-M has been presented which uses NM crossover operator and log logistic mutation operator for creating the new solution member in the search space. As discussed earlier, in the working of SOMA, no new solutions are created during the search instead only the positions of the solutions are changed. Due to which there is loss of diversity in the population as we move on to solve large scale optimization problem. So, to avoid premature convergence and for maintaining the diversity of the population, new points are created in the search space using NM crossover operator and log logistic mutation operator.

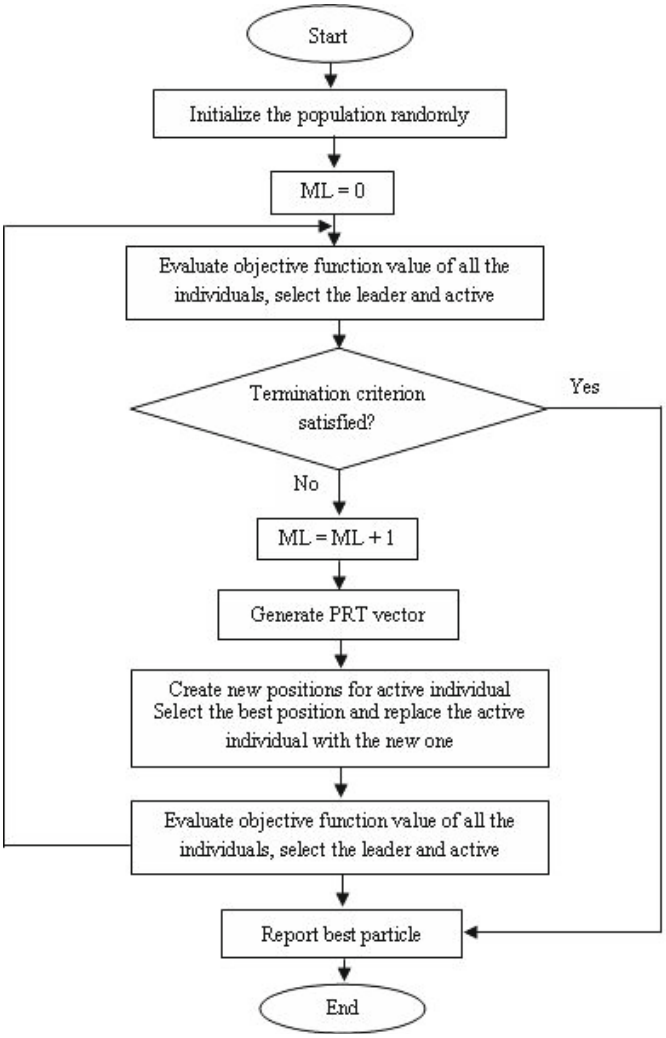


Fig. 1 Flow chart of SOMA process

3.1 Nelder Mead (NM) Crossover Operator

The Nelder Mead simplex (NM) method is a computational algorithm and is based upon the work of Spendley et al. [23]. It forms a simplex and uses this simplex to search for a local minimum. A simplex is defined as a geometrical figure which is formed by $(N+1)$ vertices, where N is the number of variables of a function. Through a sequence of elementary geometric transformation (reflection, contraction, expansion), the initial simplex moves, expands or contracts. After each transformation, the current worst vertex is replaced by a better one. In the proposed

work, Nelder Mead simplex search method has been used as a linear NM crossover operator which uses two out of three randomly chosen members from population of individuals to create a new member. The computational steps of NM crossover operator method are as follows:

Algorithm (NM Crossover Operator)

1. Choose parameters α, β and γ
2. Select an initial simplex with randomly chosen three vertices
3. Calculate function values at chosen vertices
4. Find x_h (the worst point), x_l (the best point), x_g (next to the worst point) and evaluate the value of objective functions f_h, f_l and f_g at these points
5. Compute x_c which is the centroid of x_l and x_g
6. The NM uses three operators; reflection, contraction and expansion to improve the worst point. Compute the reection point x_r by using the following expression and then compute the value of objective function f_r.

$$x_r = x_c + \alpha(x_c - x_h) \qquad \text{(Reflection)}$$

if $f_r < f_l$

$$x_{new} = (1+\gamma)x_c - \gamma x_h \qquad \text{(Expansion)}$$

else if $f_r \geq f_h$

$$x_{new} = (1-\beta)x_c + \beta x_h \qquad (Outside\ contraction) \tag{2}$$

else if $f_g < f_r < f_h$

$$x_{new} = (1+\beta)x_c - \beta x_h \qquad \text{(Inside contraction)}$$

Calculate f_{new} and replace x_h by x_{new} if $f_{new} < f_h$

7. The method continues until reaching some stopping criteria

3.2 Log Logistic Mutation Operator

The mutation operator [14] has been taken into consideration and adopted in this chapter. This randomly selects one solution x_{ij} and sets its value according to the following rule:

$$x_{ij}^{new} = \begin{cases} x_{ij} + \lambda(u - x_{ij}) & if \quad r \geq T \\ x_{ij} - \lambda(x_{ij} - l) & if \quad r < T \end{cases} \tag{3}$$

where $r \in (0,1)$ is uniformly distributed random number, u and l are the upper and lower bounds of the decision variable, $T = (x_{ij} - l)/(u - x_{ij})$ and λ is a random number following log logistic distribution and is given as equation 4.

$$\lambda = b\left(\frac{h}{1-h}\right)^{\frac{1}{\alpha}} \tag{4}$$

Where $h \in (0,1)$ is a uniformly distributed random number, $b > 0$ is a scale parameter and α is termed as mutation index as it controls the strength of mutation. More information on this operator can be found in [14].

3.3 Methodology of the Proposed Algorithm NMSOMA-M

First the individuals are generated randomly. At each generation the individual with highest fitness value is selected as leader and the worst one as active individual. Now the active individual moves towards leader in n steps of defined length. The movement of this individual is given in equation (1). Again the best and worst individual from the population is selected. Now a new point is created using Nelder Mead crossover operator using equation (2). This new point is accepted only if it is better than active individual and is replaced with active individual. Then again the best and worst individual from the population is selected. Now a new point is created using log logistic mutation operator using equation (3). This new point is accepted only if it is better than active individual and is replaced with active individual. The computational steps of NMSOMA-M are given below. The flowchart of the proposed algorithm NMSOMA-M process is depicted in Figure 2.

Algorithm(NMSOMA-M)

1. Generate initial population
2. Evaluate all individuals in the population
3. Generate PRT vector for all individuals
4. Sort all of them
5. Select the best fitness individual as leader and worst as active
6. For active individual new positions are created by using equation (1). The best position is selected and replaces the active individual by the new one
7. Create new point by crossover operator as defined in equation (2)
8. If new point is better than active, replace active with the new one
9. Create new point by mutation operator as defined in equation (3)
10. If new point is better than active, replace active with the new one
11. If termination criterion is satisfied then terminate the process; else go to step 2
12. Report the best individual as the optimal solution

4 Benchmark Functions

In this section the set of 15 scalable benchmark functions have been listed. These problems vary in nature and their complexity. The performance of proposed algorithm has been evaluated on the following functions which can be formulated as follow:

1. Ackley function

$$minf(x) = -20\exp\left(-0.02\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$$

for $x_i \in [-30,30], x^* = (0,0,\cdots,0), f(x^*) = 0$

2. Cosine Mixture

$$minf(x) = 0.1n + \sum_{i=1}^{n} x_i^2 - 0.1 \sum_{i=1}^{n} \cos(5\pi x_i)$$

for $x_i \in [-1,1]$, $x^* = (0,0,\cdots,0)$, $f(x^*) = 0$

3. Exponential

$$minf(x) = 1 - \exp\left(-0.5 \sum_{i=1}^{n} x_i^2\right)$$

for $x_i \in [-1,1]$, $x^* = (0,0,\cdots,0)$, $f(x^*) = 0$

4. Griewank

$$minf(x) = 1 + \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

for $x_i \in [-600,600]$, $x^* = (0,0,\cdots,0)$, $f(x^*) = 0$

5. Levy and Montalvo 1

$$minf(x) = \frac{\pi}{n}\left(10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\right);$$

$y_i = 1 + \frac{1}{4}(x_i + 1)$
for $x_i \in [-10,10]$, $x^* = (0,0,\cdots,0)$, $f(x^*) = 0$

6. Levy and Montalvo 2

$$minf(x) = 0.1\left(\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})]\right.$$

$$\left. + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\right)$$

for $x_i \in [-5,5]$, $x^* = (0,0,\cdots,0)$, $f(x^*) = 0$

7. Rastrigin

$$minf(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$$

for $x_i \in [-5.12,5.12]$, $x^* = (0,0,\cdots,0)$, $f(x^*) = 0$

8. Rosenbrock

$$minf(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

for $x_i \in [-30,30]$, $x^* = (0,0,\cdots,0)$, $f(x^*) = 0$

9. Schwefel 3

$$minf(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$$

for $x_i \in [-10,10]$, $x^* = (0,0,\cdots,0)$, $f(x^*) = 0$

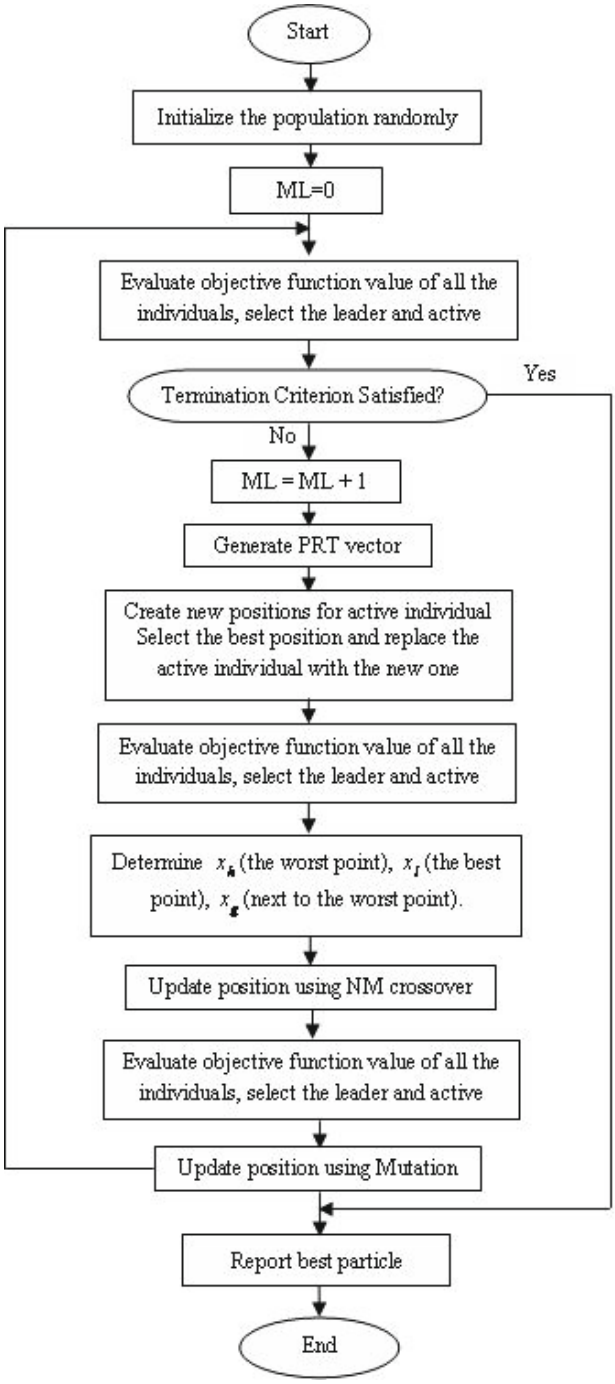


Fig. 2 Flowchart of NMSOMA-M process

10. De-Jongs function with noise

$$minf(x) = \sum_{i=0}^{n-1} (i+1)x_i^4 + rand(0,1)$$

for $x_i \in [-1.28, 1.28], x^* = (0,0,\cdots,0), f(x^*) = 0$

11. Step function

$$minf(x) = \sum_{i=1}^{n} \left(x_i + \frac{1}{2}\right)^2$$

for $x_i \in [-100, 100], x^* = (0.5, 0.5, \cdots, 0.5), f(x^*) = 0$

12. Sphere function

$$minf(x) = \sum_{i=1}^{n} x_i^2$$

for $x_i \in [-5.12, 5.12], x^* = (0,0,\cdots,0), f(x^*) = 0$

13. Axis parallel hyper ellipsoid

$$minf(x) = \sum_{i=1}^{n} ix_i^2$$

for $x_i \in [-5.12, 5.12], x^* = (0,0,\cdots,0), f(x^*) = 0$

14. Ellipsoidal

$$minf(x) = \sum_{i=1}^{n} (x_i - i)^2$$

for $x_i \in [-n, n], x^* = (1,2,\cdots,n), f(x^*) = 0$

15. Brown

$$minf(x) = \sum_{i=1}^{n-1} \left[\left(x_i^2\right)^{\left(x_{i+1}^2+1\right)} + \left(x_{i+1}^2\right)^{\left(x_i^2+1\right)} \right]$$

for $x_i \in [-1, 4], x^* = (0,0,\cdots,0), f(x^*) = 0$

5 Numerical Results on Benchmark Problems

In this section NMSOMA-M has been used to solve 15 benchmark problems in order to estimate its efficiency. For this purpose the dimension of the problems varies from 30 to 1000. In almost all problems except Griewank function the complexity of the problem increases as the dimension of the problem increases. So on the basis of the level of complexity, problems with dimension 30 is considered as small scale and 1000 as large scale. The proposed algorithm is coded in $C++$ and run on a Presario V2000 1.50 GHz computer. Since NMSOMA-M is probabilistic technique and relies heavily on the generation of random numbers, therefore 30 trials of each are carried out, each time using a different seed for the generation of random numbers.

A run is considered to be a success if the optimum solution obtained falls within 1% accuracy of the known global optimal solution. The stopping criterion is either a

run of success or a fixed number of migrations 10,000. Maximum number of function evaluations allowed are taken as 1,50,000. For fair comparison all parameters are kept same for all three algorithms. The comparative performance of these three algorithms is measured in terms of three criteria, namely accuracy, efficiency and reliability. They are described as follows:

1. Accuracy: It is based on average of mean objective function values of the successful runs
2. Efficiency: It is based on average number of function evaluations and
3. Reliability: This is based on the success rate of the algorithms

In the beginning, trials for the 15 problems are performed for dimension $n =$ 30, 50 and 100. The value of parameters after fine tuning related to NMSOMA-M, namely population size, PRT, step size, path length and total number of migrations allowed for one run are shown in Table 1.

Table 1 Parameters of NMSOMA-M

Parameters	Values
Dimension	30, 50, 100
Population size	10
PRT	0.1, 0.3, 1
Step, Path length	0.31, 3
Total number of migrations allowed	10,000
α, β and γ	1.8, 0.8 and 1

The number of successful runs (out of a total of 30 runs) taken by NMSOMA-M, PSO and SOMA for dimensions 30, 50 and 100 has been presented in Tables 2, 3 and 4 respectively. From table 2 it is clear that the performance of NMSOMA-M is better than SOMA and PSO. PSO shows worst performance out of three. The main reason behind the failure of PSO in many problems can be considered as population size. PSO requires large population size to work in comparison with SOMA. But, in the proposed method for solving 30, 50, 100, and 1000 dimension problem, only 10 population size is required. Similar kind of behavior can also be seen in table 5 and 8 respectively. The performance of SOMA and PSO start deteriorating as the complexity of the problem increases with rise in problem size. PSO almost fail to solve problems with dimension 100. On the basis of this analysis, these three algorithms can be ranked as PSO $<$ SOMA $<$ NMSOMA-M. Hence NMSOMA-M is most reliable.

In Tables 5, 6 and 7, the average number of function evaluations taken by NMSOMA-M, PSO and SOMA for dimensions 30, 50 and 100 respectively has been presented. From these three tables it is clear that NMSOMA-M attained desirable success in much lesser function evaluations as compared to SOMA and PSO. Since NMSOMA-M works with small population size, the function evaluations required is also very less. The algorithms can be ranked as PSO $<$ SOMA $<$ NMSOMA-M. Hence NMSOMA-M is most efficient.

Table 2 Percentage of success for dimension 30

Problem number	Number of successful runs out of 30		
	PSO	SOMA	NMSOMA-M
1	11	24	**30**
2	16	29	**30**
3	30	30	**30**
4	12	09	**30**
5	05	30	**30**
6	23	**29**	19
7	0	0	**30**
8	0	0	0
9	01	30	**30**
10	01	23	**30**
11	17	29	**30**
12	30	30	**30**
13	22	30	**30**
14	0	30	**30**
15	22	30	**30**

Table 3 Percentage of success for dimension 50

Problem number	Number of successful runs out of 50		
	PSO	SOMA	NMSOMA-M
1	0	06	**30**
2	0	23	**30**
3	30	30	**30**
4	03	08	**30**
5	0	30	**30**
6	16	**24**	09
7	0	0	**29**
8	0	0	0
9	0	30	**30**
10	0	0	**30**
11	06	30	**30**
12	30	30	**30**
13	18	30	**30**
14	0	30	**30**
15	11	30	**30**

Table 4 Percentage of success for dimension 100

Problem number	Number of successful runs out of 100		
	PSO	SOMA	NMSOMA-M
1	0	0	**30**
2	0	0	**30**
3	0	30	**30**
4	0	06	**30**
5	0	23	**30**
6	0	**07**	03
7	0	0	**28**
8	0	0	0
9	0	0	**30**
10	0	0	**30**
11	0	22	**30**
12	14	30	**30**
13	0	30	**30**
14	0	28	**30**
15	0	30	**30**

Table 5 Average number of function evaluations for dimension 30

Problem number	Average number of function evaluations		
	PSO	SOMA	NMSOMA-M
1	106423	32915	**422**
2	105392	15594	**339**
3	97650	6993	**382**
4	107052	22829	**595**
5	82958	10530	**5543**
6	92192	**13075**	20479
7	150000	150000	**1202**
8	150000	150000	150000
9	96440	21623	**700**
10	131170	94878	**5417**
11	107075	17341	**683**
12	93328	14159	**341**
13	96291	15695	**540**
14	150000	**16125**	37049
15	96395	15612	**848**

Table 6 Average number of function evaluations for dimension 50

Problem number	Average number of function evaluations		
	PSO	SOMA	NMSOMA-M
1	150000	50638	**603**
2	150000	23735	**645**
3	150000	12823	**445**
4	126573	42164	**568**
5	150000	18808	**9724**
6	111745	**20708**	73269
7	150000	150000	**1304**
8	150000	150000	150000
9	150000	39881	**699**
10	150000	150000	**2584**
11	123406	45490	**606**
12	116734	27072	**372**
13	117857	33949	**514**
14	150000	**35493**	60161
15	120549	24748	**758**

Table 7 Average number of function evaluations for dimension 100

Problem number	Average number of function evaluations		
	PSO	SOMA	NMSOMA-M
1	150000	150000	**708**
2	150000	150000	**564**
3	150000	44055	**445**
4	150000	114810	**665**
5	150000	58759	**20874**
6	150000	**72029**	113873
7	150000	150000	**1197**
8	150000	150000	150000
9	150000	150000	**784**
10	150000	150000	**1780**
11	150000	104382	**600**
12	150000	66735	**774**
13	150000	81890	**482**
14	150000	**87777**	135741
15	150000	67296	**925**

Tables 8, 9 and 10, present the mean objective function value corresponding to NMSOMA-M, PSO and SOMA for dimensions 30, 50 and 100 respectively. NMSOMA-M is not only achieving good success rate with lesser function evaluations but also attained objective function value with good accuracy. Results show

Table 8 Mean objective function value for dimension 30

Problem number	Mean objective function value		
	PSO	SOMA	NMSOMA-M
1	0.00971	0.000905	**0.000816**
2	0.00938	0.000817	**0.000572**
3	0.00966	0.000965	**0.000854**
4	0.00936	0.000760	**0.000736**
5	0.00941	0.00708	**0.000968**
6	0.00712	**0.000697**	0.00702
7	36.089	7.169	**0.000670**
8	55.38	327.498	**25.719**
9	0.00983	0.00841	**0.000659**
10	0.00894	0.00861	**0.000562**
11	0.00956	0.00872	**0.000690**
12	0.00943	0.000839	**0.000428**
13	0.00954	0.000807	**0.000779**
14	2618.8	0.00857	**0.000736**
15	0.00954	0.00742	**0.000570**

Table 9 Mean objective function value for dimension 50

Problem number	Mean objective function value		
	PSO	SOMA	NMSOMA-M
1	16.3998	0.000971	**0.000660**
2	0.72846	0.000772	**0.000441**
3	0.00967	0.00926	**0.000445**
4	0.00990	0.000924	**0.000630**
5	3.50605	0.000897	**0.000813**
6	0.09137	**0.000785**	0.00936
7	95.1538	14.249	**0.000737**
8	135.189	193.262	**45.619**
9	59.0000	0.000858	**0.000596**
10	7.27434	0.03268	**0.000574**
11	0.00914	0.00848	**0.000497**
12	0.00968	0.000941	**0.000648**
13	0.00944	0.000785	**0.000510**
14	1904.80	0.00846	**0.000803**
15	0.00963	0.00789	**0.000730**

that the ranking of all the algorithms is PSO $<$ SOMA $<$ NMSOMA-M. Hence NMSOMA-M is most accurate.

Table 11 has presented the results taken by only NMSOMA-M for dimension 1000. Since the performance of SOMA and PSO has not been found satisfactory

Table 10 Mean objective function value for dimension 100

Problem number	Mean objective function value		
	PSO	SOMA	NMSOMA-M
1	19.9251	3.03138	**0.000859**
2	36.1656	0.78662	**0.000270**
3	0.99999	0.00883	**0.000762**
4	669.999	0.00412	**0.000755**
5	5.69121	0.00230	**0.000980**
6	0.00958	**0.000405**	0.00944
7	329.632	55.263	**0.000685**
8	195.04	617.418	**95.352**
9	171.254	0.89232	**0.000711**
10	1912.25	0.14871	**0.00769**
11	*	0.00692	**0.000240**
12	0.00980	0.000816	**0.000742**
13	1534.60	0.000904	**0.000134**
14	*	0.00807	**0.00101**
15	*	0.00873	**0.000730**

rather disappointing as the dimension rises to 1000, results are not taken by these algorithms for dimension 1000. Although NMSOMA-M has already proved its robustness by solving 100 dimensional problems using 10 population size, the main purpose of using NMSOMA-M for solving 1000 dimensional problems is to show its efficacy to solve large scale problems. In Table 11, success rate, average function evaluations and mean objective function value of NMSOMA-M for dimension 1000 has been presented. Success rate obtained by NMSOMA-M is very good. Function evaluations taken by this algorithms is also very less with desirable accuracy. Therefore, NMSOMA-M can be considered as a good approach for solving large scale function optimization problems.

The problems which could not be solved by the particular algorithm is given the sym-bol $(*)$ at the corresponding entries. After analyzing the performance of all three algorithms in terms of three criterions, a compact view of results is reported in Table 12. NMSOMA-M outperforms PSO and SOMA in all the factors considered.

In order to reconfirm our results and to show the results graphically, the relative performance of all the algorithms has been analyzed by using a Performance Index (PI) The relative performance of an algorithm using this PI is calculated by using the following equation (5).

$$PI = \frac{1}{N_p} \sum_{i=1}^{N_p} \left(k_1 \alpha_1^i + k_2 \alpha_2^i + k_3 \alpha_3^i \right) \tag{5}$$

Where

$$\alpha_1^i = \frac{Sr^i}{Tr^i}$$

Table 11 Performance of NMSOMA-M for dimension 1000

Problem number	Success rate	Average number of function calls	Mean objective function value
1	30	758	0.000645
2	30	581	0.000658
3	30	522	0.000479
4	30	718	0.000537
5	30	54373	0.000945
6	*	*	*
7	25	2123	0.000316
8	30	150000	994.667
9	30	1066	0.000760
10	30	3932	0.000763
11	30	749	0.000651
12	30	660	0.000449
13	30	698	0.000597
14	*	*	*
15	30	1751	0.000570

$$\alpha_2^i = \begin{cases} \frac{Mo^i}{Ao^i} & if \quad Sr^i > 0 \\ 0 & if \quad Sr^i = 0 \end{cases}$$

$$\alpha_3^i = \begin{cases} \frac{Mf^i}{Af^i} & if \quad Sr^i > 0 \\ 0 & if \quad Sr^i = 0 \end{cases}$$

Sr^i= Number of successful runs of i^{th} problem
Tr^i= Total number of runs of i^{th} problem
Ao^i= Mean objective function value obtained by an algorithm of i^{th} problem
Mo^i= Minimum of mean objective function value obtained by all algorithms of i^{th} problem
Af^i= Average number of function evaluations of successful runs used by an algorithm in obtaining the solution of i^{th} problem
Mf^i= Minimum of average number of function evaluations of successful runs used by all algorithms in obtaining the solution of i^{th} problem
N_p= Total number of problems analyzed

The variables k_1, k_2 and k_3; $k_1 + k_2 + k_3 = 1$ and $0 \leq k_1, k_2, k_3 \leq 1$ are the weights assigned to percentage of success, mean objective function value and average number of function evaluations of successful runs, respectively. From the above definition it is clear that modified PI is a function of k_1, k_2 and k_3 since $k_1 + k_2 + k_3 = 1$, one of $k_i, i = 1, 2, 3$ could be eliminated to reduce the number of variables from the expression of PI. But it is still difficult to analyze the behavior of this PI, because the surface of PI for all the algorithms are overlapping and it is difficult to visualize

Table 12 Comparison of NMSOMA-M, PSO, SOMA

Dimension	Factors	Performance of NMSOMA-M	NMSOMA-M Vs PSO	NMSOMA-M Vs SOMA	Overall performance of NMSOMA-M, PSO and SOMA
30	Success rate	Better	11	06	PSO: 02
		Equal	03	08	SOMA: 09
		Worse	01	01	NMSOMA-M: 13
	Average function calls	Better	15	12	PSO: 0
		Equal	00	01	SOMA: 02
		Worse	00	02	NMSOMA-M: 12
	Mean function value	Better	15	14	PSO: 0
		Equal	00	00	MOMA 01
		Worse	15	14	NMSOMA-M: 14
50	Success rate	Better	13	05	PSO: 01
		Equal	01	09	SOMA: 10
		Worse	01	01	NMSOMA-M: 13
	Average function calls	Better	15	12	PSO: 0
		Equal	00	01	SOMA: 02
		Worse	00	02	NMSOMA-M: 12
	Mean function value	Better	14	14	PSO: 0
		Equal	01	00	MOMA 01
		Worse	00	01	NMSOMA-M: 14
100	Success rate	Better	14	09	PSO: 01
		Equal	01	05	SOMA: 05
		Worse	00	01	NMSOMA-M: 13
	Average function calls	Better	15	12	PSO: 0
		Equal	00	01	SOMA: 02
		Worse	00	02	NMSOMA-M: 12
	Mean function value	Better	15	14	PSO: 0
		Equal	00	00	MOMA 01
		Worse	00	01	NMSOMA-M: 14

them. Hence equal weights are assigned to two terms at a time in the PI expression. This way PI becomes a function of one variable. The resultant cases are as follows:

(i) $k_1 = w, k_2 = k_3 = \frac{1-w}{2}, 0 \leq w \leq 1$

(ii) $k_2 = w, k_1 = k_3 = \frac{1-w}{2}, 0 \leq w \leq 1$

(iii) $k_3 = w, k_1 = k_2 = \frac{1-w}{2}, 0 \leq w \leq 1$

The graph corresponding to each of case (i), (ii) and (iii) for dimension 30 is shown in Figure 3, where the horizontal axis represents the weight *w* and the vertical axis represents the performance index PI. The graph corresponding to each of case (i), (ii) and (iii) for dimension 50 is shown in Figure 4 whereas Figure 5 dipcts the graph corresponding to each of case (i), (ii) and (iii) for dimension 100.

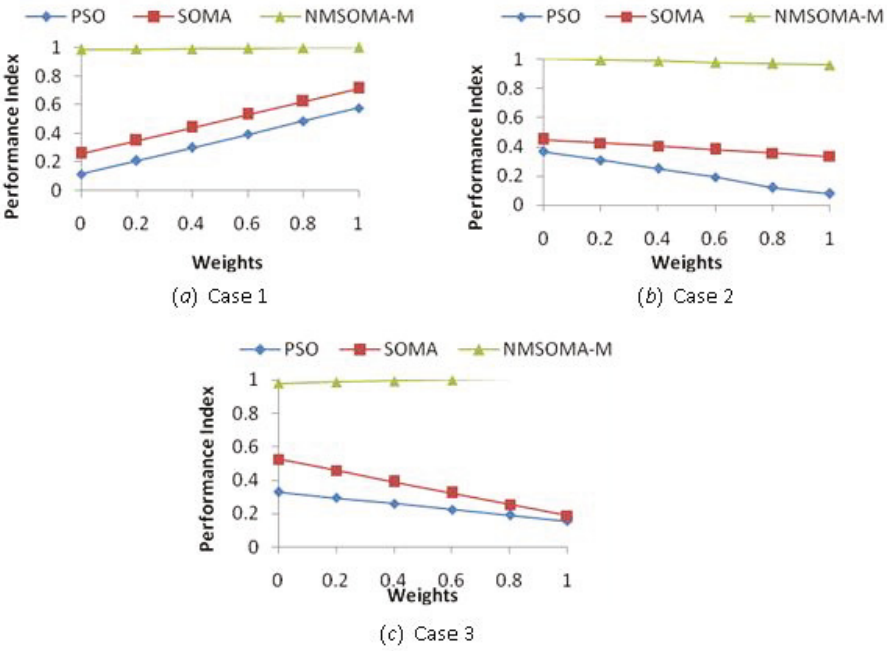


Fig. 3 Performance index of PSO, SOMA and NMSOMA-M for dimension 30

In case (i), the mean objective function value and average number of function evaluations of successful runs are given equal weights. Performance index's of NMSOMA-M, PSO and SOMA are superimposed in the Figures 3(i), 4(i) and 5(i). It is observed that the value of performance index for NMSOMA-M is more than PSO and SOMA. In case (ii), equal weights are assigned to the numbers of successful runs and mean objective function value of successful runs. Performance indexs of NMSOMA-M, PSO and SOMA are superimposed in the Figures 3(ii), 4(ii) and 5(ii). It is observed that the value of PI for NMSOMA-M is more as compared to PSO and SOMA. Similar case also observed in case (iii). This can be viewed from the Figures 3(iii), 4(iii) and 5(iii).

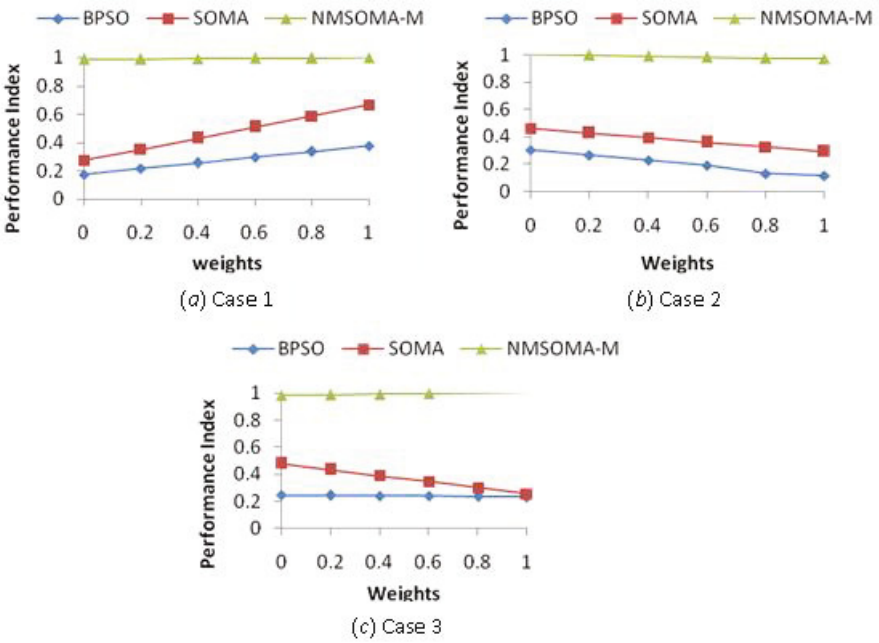


Fig. 4 Performance index of PSO, SOMA and NMSOMA-M for dimension 50

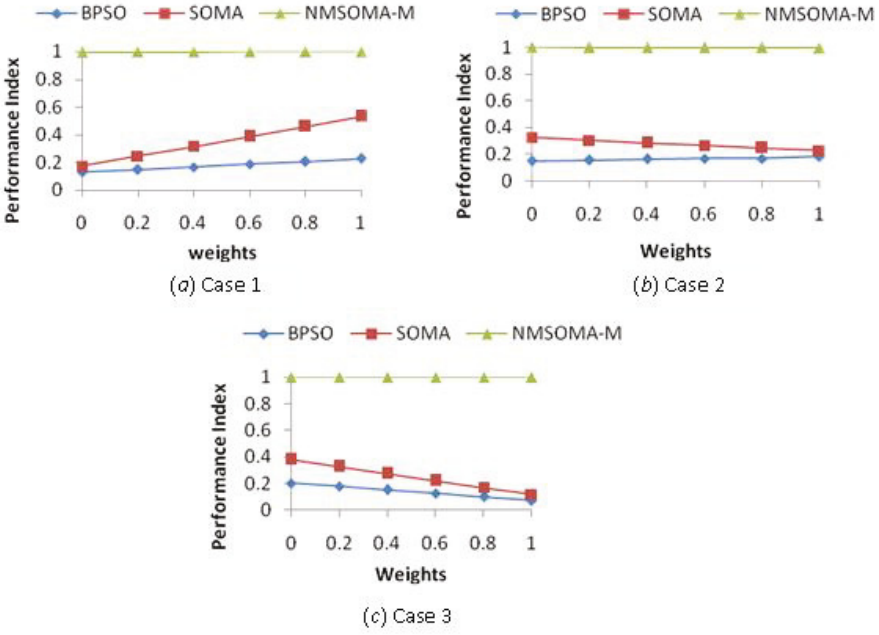


Fig. 5 Performance index of PSO, SOMA and NMSOMA-M for dimension 100

6 Conclusion

In this chapter a hybridized variant NMSOMA-M of SOMA with Nelder Mead crossover operator and log logistic mutation has been presented. This algorithm has been designed to improve the efficiency of SOMA and to overcome the difficulty of premature convergence due to trapping in local optimal solution. Though SOMA works well for solving small scale problems but its performance becomes poor due to loss of diversity as the dimension of the solution space becomes large. In the working of presented algorithm NMSOMA-M, NM crossover and log logistic mutation operator are used to maintain the diversity in the solution space by creating new points. NMSOMA-M has been tested on a set of 15 scalable test problems and results are taken for dimension 30, 50, 100 and 1000 respectively. Since the performance of PSO and SOMA was not found satisfactory, the results obtained by NMSOMA-M have been compared with results obtained by SOMA and PSO only for dimension 30, 50 and 100. NMSOMA-M not only attained good success rate, in less function evaluations with desirable accuracy but also use very small population size to work with. At last on the basis of the results presented it can be concluded that the presented algorithm NMSOMA-M is an efficient, reliable and accurate to solve large scale real life optimization problems.

References

1. Zelinka, I., Lampinen, J.: SOMA - Self organizing migrating algorithm. In: Proceedings of the 6th International Mendel Conference on Soft Computing, Brno, Czech, Republic, pp. 177–187 (2000)
2. Domingo, O.B., Cessae, H.M., Nicolas, G.P.: CIXL2: A crossover operator for evolutionary algorithms based on research. Journal of Artificial Intelligence Research 24, 1–48 (2005)
3. Chelouah, R., Siarry, P.: A hybrid method combining continuous tabu search and nelder – Mead simplex algorithm for global optimization of multiminima functions. European Journal of Operational Research 161, 636–654 (2005)
4. Deep, K., Dipti, S.: A new hybrid self organizing migrating genetic algorithm for function optimization. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 2796–2803 (2007)
5. Fan, K.-S., Liang, Y.C., Zahara, E.: A genetic algorithm and a particle swarm optimizer hybridized with nelder-mead simplex search. Computers and Industrial Engineering 50, 401–425 (2006)
6. Premalatha, K., Nataranjan, A.M.: Hybrid PSO and GA for global optimization. International Journal of Open Problems and Computational Mathematics 2(4), 597–608 (2009)
7. Khosravi, A., Lari, A., Addeh, J.: A new hybrid of evolutionary and conventional optimization algorithm. Applied Mathematical Sciences 6, 815–825 (2012)
8. Ghatei, S., Panahi, F.T., Hosseinzadeh, M., Rouhi, M., Rezazadeh, I., Naebi, A., Gahtei, Z., Khajei, R.P.: A new hybrid algorithm for optimization using PSO and GDA. Journal of Basic and Applied Scientific Research 2, 2336–2341 (2012)

9. Esmin, A.A.A., Matwin, S.: A hybrid particle swarm optimization algorithm with genetic mutation. International Journal of Innovative Computing, Information and Control 9, 1919–1934 (2013)
10. Pant, M., Thangaraj, R., Abraham, A.: A new PSO algorithm with crossover operator for global optimization problems. In: Corchado, E., Corchado, J.M., Abraham, A. (eds.) Innovations in Hybrid Intelligent Systems, vol. 44, pp. 215–222. Springer, Heidelberg (2007)
11. Bansal, J.C., Deep, K.: Quadratic approximation PSO for economic dispatch problems with valve-point effects. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Dash, S.S. (eds.) SEMCCO 2010. LNCS, vol. 6466, pp. 460–467. Springer, Heidelberg (2010)
12. Deep, K., Thakur, M.: A new mutation operator for real coded genetic algrithms. Applied Mathematics and computation 193, 229–247 (2007)
13. Xing, L.N., Chen, Y.W., Yang, K.W.: A novel mutation operator base on immunity operation. European Journal of Operational Research 197, 830–833 (2009)
14. Deep, K., Shashi, Katiyar, V.K.: A new real coded genetic algorithm operator: Log logistic mutation. In: Deep, K., Nagar, A., Pant, M., Bansal, J.C. (eds.) Proceedings of the International Conference on SocProS 2011. AISC, vol. 130, pp. 193–200. Springer, Heidelberg (2012)
15. Mohan, C., Shankar, K.: Random search technique for global optimization. Asia Pacific Journal of Operations Research 11, 93–101 (1994)
16. Deep, K., Das, K.N.: Quadratic approximation based hybrid genetic algorithm function optimization. Applied Mathematics and Computations 203, 86–98 (2008)
17. Deep, K., Bansal, J.C.: Hybridization of particle swarm optimization with quadratic approximation. Opsearch 46, 3–24 (2009)
18. Deep, K., Das, K.N.: Performance improvement of real coded genetic algorithm with quadratic approximation based hybridization. International Journal of Intelligent Defence Support Systems 2, 319–334 (2010)
19. Pant, M., Thangaraj, R., Abraham, A.: A new quantum behaved particle swarm optimization. In: Keijzer, M. (ed.) Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO, Netherlands, pp. 87–94 (2008)
20. Singh, D., Agrawal, S., Singh, N.: A novel variant of self organizing migrating algorithm for function optimization. In: Pant, M., Deep, K., Nagar, A., Bansal, J.C. (eds.) Proceedings of the Third International Conference on Soft Computing for Problem Solving. AISC, vol. 258, pp. 225–234. Springer, Heidelberg (2014)
21. Eberhart, R.C., Kennedy, J.: Partcle Swarm Optimization. In: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948. IEEE Service Center, Piscataway (1995)
22. Zelinka, I.: SOMA - Self organizing migrating algorithm. In: Onwubolu, G.C., Babu, B.V. (eds.) New Optimization Techniques in Engineering. STUDFUZZ, vol. 141, pp. 167–217. Springer, Heidelberg (2004)
23. Spendley, W., Hext, G.R., Himsworth, F.R.: Sequential application of simplex designs in optimization and evolutionary operation. Technometrics 4, 441–461 (1962)

A Spectrum of Big Data Applications for Data Analytics

Ritu Chauhan and Harleen Kaur

Abstract. As technology is gaining its insights, vast amount of data is getting collected from various resources. Foremost complex nature of data is providing challenging task among the researchers to store, process and analyze big data. At present, big data analytics tends to be an emerging domain which potentially has limitless opportunities for possible future outcomes. However, big data mining provides application capabilities to extract hidden information from large volumes of data for knowledge discovery process. In fact big data mining is demonstration varied challenges and vast opportunity among researchers and scientist for another upcoming decade. This chapter provides broad view of big data in medical application domain. In addition, a framework which can handle big data by using several preprocessing and data mining technique to discover hidden knowledge from large scale databases is designed and implemented. The proposed chapter also discuss the challenges in big data to gain insight knowledge for future outcomes.

1 Introduction

In past era there is a tremendous increase in data collected from various resources which includes sensor technology, geographic information systems, health informatics, pattern recognition, marketing survey data, imaging and other resources. This data type has outraged human capabilities to process information for knowledge discovery process. However, these resources of datasets are termed as "Big Data" which refers to voluminous nature of data with complex structure. In order to process such big data, traditional database technology proves insufficient to store,

Ritu Chauhan
Amity University, Sector 125, Noida, India
e-mail: rituchauha@gmail.com

Harleen Kaur
Jamia Hamdard University, Delhi, India
e-mail: harleen_k1@rediffmail.com

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_7

extract and analyze data. However, researchers are putting there paramount capabilities to discover knowledge from big data. The concept of big data was raised in 1998 for first time in a silicon graphics (SGI) slide deck by John Mashey with the title of "Big Data and the Next Wave of InfraStress" [1]. The perception of analyzing big data was introduced by Weiss and Indrukya in which they discuss the context of big data with respective data mining techniques [2]. The retrospective studied were suggested due to fact that huge amount of data is gathered from various resources. Conversely, big data can be stated as '3V's' model proposed by Gartner, where he discussed the concept of big data with three relevant attributes: high volume of data as expandable nature of data with high rate and insufficient amount of tools which can actually analyze data, high velocity of data flow where the context of data is high speed and retrieving relevant information in real time environment, and high variety of data types where he discussed data as structured and unstructured with different formats which includes text, video, imaging, sensor and other types [33]. Eventually with prospective of time the big data has become an ambiguous term whose actual meaning is constantly varying with identical research studies. In recent study Demchenko et al. [5], dened big data with respect to five relevant attributes which includes volume, velocity, variety, veracity, and value [33]. Volume is represented as large amount of data gathered from various resources, velocity refers to speed of data in respective of their creation, variety defines the complex nature of data, veracity refers to fact the significance and authenticity of big data, and finally value refers to quality of outcomes and results formed which were previously hidden.

The notion of big data can be studied in several application areas which include:

1. Health Informatics: To discover independent data of patients for monitoring and retrieving hidden facts which can improve futuristic decision making is known as Health Informatics. However the traits of Demchenko et al. [5] if studied on health care can be termed as: volume where it can be discussed as large data of patient record, as well as independent attribute in case of microarray datasets. Velocity can refer to rate at which datasets are retrieved. Variety refers to different types of data available with complex nature of data. Veracity refers to fact that real world clinical data may contain noisy, incomplete, or erroneous data which must be handled cautiously. Hence, veracity handles potentially clinical data from erroneous problems.
2. Business: The application of big data in entrepreneur domain can discover new trends and behavior of customers for bridging new challenges and access marketing trends for future customer oriented business. However the vast nature of big data can offer challenges for entrepreneur to discuss advantages and disadvantages of current policies and their future outcomes.
3. Future Investment: The data analyzed in different domains which include financial transactions can be potentially applied for sustainable economic development where the focus would be on improving quality of life with respective resources available for mankind.

These application areas will eventually enhance different sectors and develop futuristic decision making for different domains. Hence, the research conducted

in this chapter focuses on application area of health informatics where data was collected among the independent and varied features [3].

However, its concern that health data supposed to meet all the requirements of big data but still it requires several computation techniques to process its data. As we know the large volume of data collected from online database system may require storage as well as fast computation technology to handle complexity of data and requires algorithms which can interfere in developing future decision making policy [7, 9, 11, 13, 20, 25]. For this chapter, several features of clinical big data is reviewed and determine different methods used for manipulation and analysis of these datasets. This chapter is focused on clinical datasets and analysis through various data mining techniques which can handle big data. The selected studies is examined to extract information on research interests, goals, achievements, and the implemented methodologies for future medical decision making.

The rest of chapter is discussed as follows: Section 2 discuss the brief concept of big data in clinical domain, Section 3 discuss the framework for retrieval of information from big data. Results and implementation is discussed in Section 4. Last Section 5 discusses about conclusion and future works.

2 Big Data in Clinical Domain

As we know that health care data is expected to grow exponentially large in years ahead. In addition health care policies are varying according to preference of patient's needs and healthcare environments. However purpose is to acquire generative tools which can handle big data, benefit healthcare practioners and patients for future medical diagnosis. The goal among researchers and health care providers is to integrate, digitize the value of big data in different health care sectors which includes hospital, small nursing homes, independent physician offices and other healthcare organizations to realize significant nature of big data analysis for future outcomes [18,19, 21].

The actual outcomes include detecting a disease at earlier stages, for example detecting the cancer at earlier stage; can be cured more effortlessly rather than later outcomes. Certainly other domains can also utilize big data analytics such as fraud detection in healthcare for detecting maximum fraud cases in insurance sector. However several questions can be answered, while using big data analytics in clinical domain which includes patients who will survive surgery, patients more prone to breast cancer, patient's disease progression, patients who will not benefit from surgery, patients who can acquire certain disease during hospital stay and other conditions. A survey was conducted by McKinsey who estimates that big data analytics can approach more than $300 billion in savings per year in U.S. healthcare. He briefed that two thirds can be achieved by reductions of approximately 8% in national healthcare expenditures. McKinsey converse that big data can help

in reducing waste and inefficiency at different application areas which includes clinical research to discuss more relative clinical factors and cost effective ways to diagnose and treat patients with best possible clinical decision making [6]. The other areas which includes research and development are new improved statistical tools and algorithms to handle better treatment process for individual patient care, thus growing new trends for diagnosis in market. Discover adverse effect of certain drug before reaching market or studying different trends for drug designing and clinical trials by analyzing patient records. The process of targeting more accurately tested vaccine for specific disease or certain communicable disease. The data should be formed into knowledge for identifying needs during crises, so best utilized during prevention outcomes. Big data can contribute in other healthcare domains which include studying the micro array data of patients and determine the gene effects for future generations and predicts the future interventions. The other areas take account of where patient have maximum chances of getting a specific disease and may have preventive measure before acquiring the disease.

Recently a study conducted by Duhigg (2012) revealed that a girl pregnancy was figured out by analyzing the consumer behavior of the girl [8]. The clinical big data can also be utilized to determine causality, effect, or connection between the risk factors and the occupancy of deriving the disease. Ursum et al. [10] discuss the contextual relationships among the sero conversion and relative age of patients with auto antibodies, which acts as inflammatory effects in 18,658 rheumatoid arthritis patients and controls, hence discuss the concept of citrullinated proteins and peptides tends to be more reliable markers. Consequently, draws the conclusion that citrullinated proteins and peptides tends be reliable to rheumatoid arthritis than was immunoglobulin M rheumatoid factor.

Ajdacic-Gross et al [12] analyze the data on 11,905 Swiss conscripts from 2003 for stuttering and discovering. No single overwhelming risk factor for stuttering was discovered, however several factors such as premature birth and parental alcohol abuse appeared influential.

3 Framework for Big Data Analytics

The raw big data can be highly unstructured with divisive nature, hence to discover knowledge from such data can lead to deceptive results. Eventually, to deal with complexity of big data researchers and scientists are laying down numerous efforts to ascertain new techniques for retrieval of hidden information. Although analysis of big data lay challenges and offers knowledge which can benefit several application domains which include marketing, political elections, clinical data, sports, media, micro array , imaging data, geographic information data, sensor data, social media, remote sensing and online transactional system [12, 14, 15, 17, 19, 23, 25, 26]. The benefits of big data are abundant and far exceeding, so appropriate data analytics technology to discover hidden facts and knowledge from large volumes of data is required [4]. The current framework presented in this chapter is an integrated

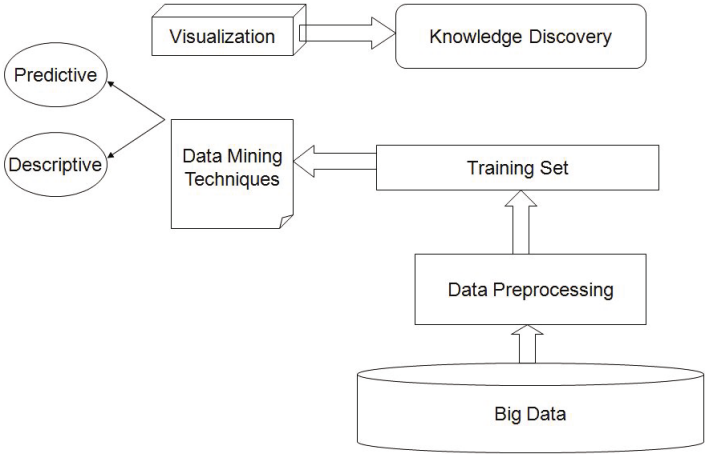


Fig. 1 Framework for big data analytics

environment which can appropriately handle large complex data and discover patterns for future knowledge discovery. The Figure 1 represents a framework for data analytics and knowledge discovery.

3.1 Big Data

The bottom layer of framework comprise of big data which can be represented as $B < a_1, a_2, \cdots, a_n >$ where a represents the attributes corresponding to $r < r_1, r_2, \cdots, r_n >$, r row values in time series data $t < t_1, t_2, \cdots, t_n >$. Eventually, process of extraction in big data analytics is exploratory search of unknown and hidden facts where the user doesn't have enough knowledge about the data. A study performed by Jimmy Lin discusses the concept of big data with data mining technology however the study was based on twitter data [35]. He discuss about infrastructure capabilities of big data and data mining techniques in broad spectrum. However, proposed approach is based on healthcare big data to discover hidden patterns which can benefit healthcare practioners for future medical diagnosis. This chapter examines different data mining techniques to extract information on research goals and implement them for future diagnosis.

3.2 Data Preprocessing

The raw big data is highly susceptible to inconsistent data records, noise and missing values. The inconsistent nature of data can generate results which can provide null benefit to end users. In order to improve the quality of results and dis-cover knowledge from big data, preprocessing technique should be involved for am-putation of inconsistent values from datasets [22, 27, 28, 29, 30, 31]. However preprocessing

tends to be critical technique among the data mining process which involves data cleaning, data integration, data transformation and data reduction.

3.3 Training Set

Raw big data contains features which are at times no use to end user. So instead of handling entire database, a large set of data is often extracted from raw data for further data analytics. However if the same terminology is applied in clinical domain for heart patients then instead of keeping entire data, doctors can use secondary data where heart related features can extract future prognosis of disease. Further the foremost features can be kept for analysis instead of utilizing entire data sets which occupies space and involves cost for effective and efficient analysis of data. As end result in clinical medicine the researchers are only interested in analyzing secondary data which consists of features of their own interest. Ultimately the secondary data is enormous in size and hence require several computation techniques for its analysis.

3.4 Data Mining Techniques

Data mining is an interdisciplinary field emerged from various techniques such as artificial intelligence, soft computing, multi dimensional databases, statistical and mathematical discipline with machine learning community. The process relies to discover hidden information and knowledge from large databases for futuristic problem solving. To solve real datasets tribulations, data mining techniques are generally diversified as predictive and descriptive. The descriptive data mining techniques generalize the different characteristics of databases while predictive data mining tasks perform the presumption of data in order to make futuristic decision making [19, 20, 21]. The descriptive data mining focuses on determine the patterns which are interpreted by humans after considering the whole dataset and constructs the model interpretation of data. The predictive data mining evolve around some variables to generated unknown or future models based on variables of interest by the given dataset.

However the goals of predictive and descriptive data mining tasks have common characteristics except that predictive techniques requires the special variable while the goal of descriptive data is to combine data in specific way. Conversely, the goal of descriptive data mining is to generate patterns within the data and discover the relationships among the attributes whereas the prediction techniques predict the future outcomes of data. The main predictive and descriptive data mining tasks can be classified as following [21]:

Classification: Classification technique utilizes the predictive learning technique that creates models to predict the targets from the set of predefined variables or sets. It can iteratively be defined as a process of finding function that classifies the data into number of classes. It is a two step process to build for future prediction. Firstly process begins with describing predefined set of classes or concept hierarchies and then the model is further evaluated for classification. However classification

technique is used for prediction, whereas the values are already known by the user. For example, classification techniques can be utilized by healthcare practioners to answer prediction tasks such as if patients are suffering from specific symptoms, then after certain time they can have acquire disease with relation to symptoms.

Cluster analysis: Cluster analysis is descriptive data mining technique to group the similarity among the associated object while utilizing different statistical techniques to determine the clusters of variants shapes and size. However there exists no predefined classes in clustering as it determines the similarity while using different similarity measures. Data clustering usually generates the clusters with overall areas related to sparse and crowded objects and measure similarity among the clusters [34]. The approach of new database oriented clustering methods has already been indicated to cluster data in space. These well known clustering algorithms from partitioning based clustering methods such as k-means are highly inefficient and scalable for high dimensional databases [16]. The assumption made by algorithms available in literature is all objects which shows similarity are clustered and they reside in memory at same time.

Association Rules: Association rules are descriptive data mining techniques that associates the relationship among the variables in big datasets. The association rules are represented as (A, B) as set of attributes, represented by $A \rightarrow B$, where the left side of arrow represents antecedent (that is A) and right side of arrow is consequent (that is B).

However, rules are generated in the form of support and confidence among the datasets. Whereas, support is defined as the number of transactions are fixed in an event then occurrence of object in particular event is defined as the support of that event. It can be represented in transaction as $supp(A \rightarrow B) = supp(A)$. The confidence in rules can be defined as estimation of probability in transaction, it can be represented as $conf(A \rightarrow B) = \frac{supp(A \rightarrow B)}{supp(A)}$.

Segmentation: Segmentation techniques are usually opted for different segments of population with consideration into demographic variables which may include age, race, or gender. The segmentation techniques are used as analysis technique to break the population in form of groups and finally perform a given tasks for the same. For example, segmentation technique can be applied in between children with 0-2 years for vaccination of HIV. As well as there are numerous questions which can be answered using segmentation techniques such as 'number of patients those are suffering from lung cancer for specific age group'. The race that is effected most and several other related queries which can also be answered.

3.5 Description and Visualization

The data mining techniques are widely used to discuss and visualize the large datasets for discovery of hidden rule and patterns. Although when datasets becomes large in number the explanation of data becomes really typical, but data mining visualization techniques are extremely practical in different domains to extract visual patterns for

users. The visualization can be termed as strong descriptive data mining techniques for determination of hidden patters from large scale databases.

4 Results and Implementation

The dataset represented in this chapter consists of diabetic patients admitted to hospital for diagnosis. The data was collected from 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. The data was summarized for (18 hospitals), which includes Northeast (58), South (28), and West (16). The data consists of over 50 features representing patient and hospital outcomes. The features were extracted from the unique visit of patients during hospital stay. Hence, data is integrated from both inpatient and outpatient, which includes emergency department and several other resources [32]. The data contains several attributes such as patient number, race, gender, age, admission type, time in hospital, medical specialty of admitting physician, number of lab test performed, HbA1c test result, diagnosis, number of medication, diabetic medications, number of outpatient, inpatient, and emergency visits in the year before the hospitalization etc.

The data was filtered and discretized with *R*-first last, where generative study was conducted among the population data and hence was complied among 5 categories of race which includes Caucasian where the generated cases were 47148 among the highest in study, African American cases were 14693, Hispanic includes 1236 cases, Asian were 321 among the lowest cases studies and all other race include as 804. The categorization of above data was made in gender category where females were 35448 larger in number as compare to male patients as 30086 and there was 1 invalid data present for the analysis.

Further the analysis of data was conducted among the age group of patients; the data was summarized from 1-90 year and above. The study suggested that maximum patients admitted was among the age group 70-80 years which includes 17047 patients however the second highest was among the age group of 60-70 where 14304 patients were diagnosed. Further study suggested that maximum cases of male patients admitted in hospital were among 70-80 year old however the female cases admitted maximum was among the age group 50-60 year. The minimum patients diagnosed were less than 10 years old. However, the admission status of all patients were defined among the category as emergency, urgent, elective, newborn, not available, null, trauma center, and Not maped. The 45327 patient cases were admitted in hospital as emergency and urgent grounds. Further the patients discharge status was studied in regard to patient's admission and cases were diagnosed. It is found that maximum patients admitted in hospital were discharged to home (38514), however the other discharge status of patients can be found in Table 1.

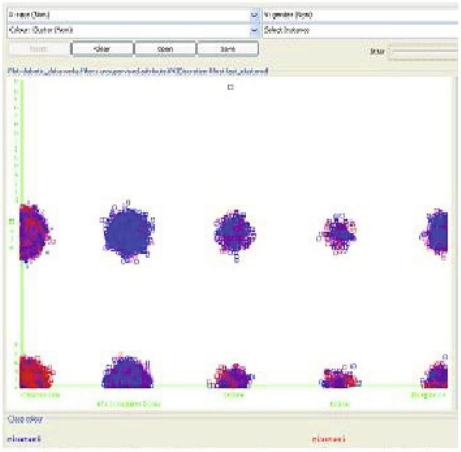


Fig. 2 Clusters in accordance with race and gender

Further analytics study was conducted using training data set to form clusters using k means clustering technique for analysis of data. The results shown in Figure 2 suggest that two clusters were formed were cluster-0 has acquired 41037 (63%) cases whereas cluster-1 has 24498 (37%) cases.

Cluster-0 represents Caucasian race where male cases were 30433 and female cases were 18048. Cluster 1 represents the African American race where male cases studied were 9204 as compared to female cases which were 5489. Further the experimentation was conducted in retrospective of discharge status with associated race. It is found that Caucasian population among the female cases was more prominent when discharged from hospital. Figure 3 proves the linked results for the same.

The study also suggested the male cases were maximum as discharged from hospital among the race African American. In Figure 4 a similar study was conducted in respective of age, race and gender of patients admitted to hospital.

The study show that clusters were found maximum among the age group from 50-80 yrs, however male cases were higher in Caucasians race as compared to African American. Whereas, female cases were maximum in number for African American patients admitted to hospital care.

In Figure 5 the clusters were determined in comparison with diabetes medicine and attributes which include age and race. It founds two clusters such as cluster-0, represents total of 35157 cases, and cluster-1, has 30378 cases. The maximum male cases were encountered with age group of 70-80 yrs and 2716 cases were given no medication whereas 2441 were given diabetic medicine.

The cluster-1 represents the female population among the age group of 50-60 year where the diabetes medicine was provided to 16901 patients and there was no diabetes medicine given to 13477 patients. In Figure 6 retrospective studies was conducted for discharge status, age, gender and diabetes medicine.

Table 1 Discharge status of patient cases

Discharge status	Cases	Discharge status	Cases
Discharged to home	38514	Still patient or expected to return for outpatient services	187
Discharged/transferred to another short term hospital	1350	Hospice / home	186
Discharged/transferred to SNF	8050	Hospice / medical facility	54
Discharged/transferred to ICF	664	Discharged /transferred within this institution to Medicare approved swing bed	
Discharged/transferred to another type of inpatient care institution	1011	Discharged/transferred/referred another institution for outpatient services	25
Discharged/transferred to home with home health service	7636	Discharged/transferred/referred to this institution for outpatient services	
Left AMA	368	Null	
Discharged/transferred to home under care of Home IV provider	105	Expired at home. Medicaid only, hospice	
Admitted as an inpatient to this hospital	14	Expired in a medical facility, Medicaid only, hospice	
Neonate discharged to another hospital for neonatal aftercare	1138	Expired, place unknown. Medicaid only, hospice	3671
Expired	3	Discharged/transferred to another rehab fac including rehab units of a hospital	6
Unknown/Invalid	977	Discharged/transferred to a long term care hospital	1164
Discharged/transferred to another type of health care institution not defined elsewhere	4	Discharged/transferred to a nursing facility certified under medicaid but not certified under medicare	308
Discharged/transferred to a federal health care facility	90	Not mapped	10
Discharged/transferred/referred to a psychiatric hospital of psychiatric distinct part unit of a hospital	90	Discharged/transferred to a critical access hospital (CAH)	90

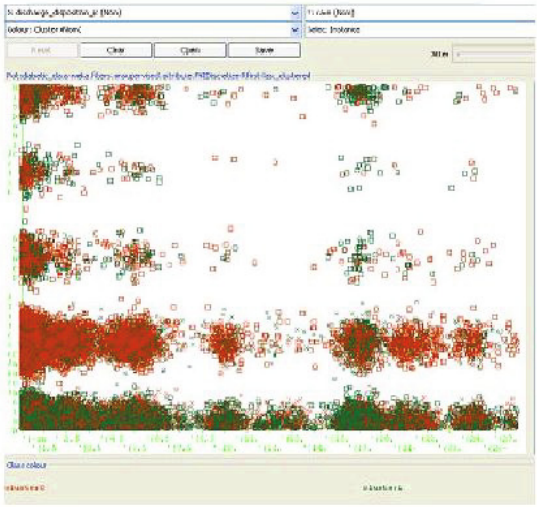


Fig. 3 Clusters in accordance with discharge status

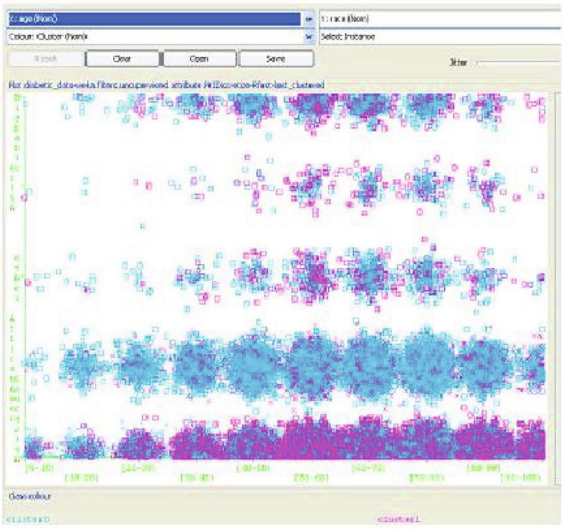


Fig. 4 Clusters in accordance with race and age

The study proves that we formed two clusters such as cluster-0 and cluster-1. Cluster 0 proves that maximum male patients with all age groups and race Caucasian, who were admitted to hospital on emergency and urgent encounters were given diabetes medicine, got discharged from hospital early as compared to other patients with no diabetes medicine provided. Whereas, cluster-1 represents the

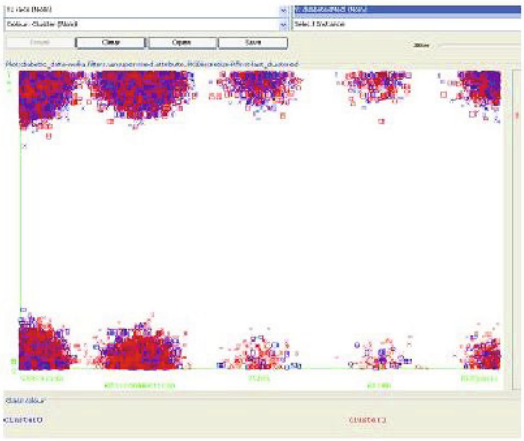


Fig. 5 Clusters in accordance with diabetes medicine

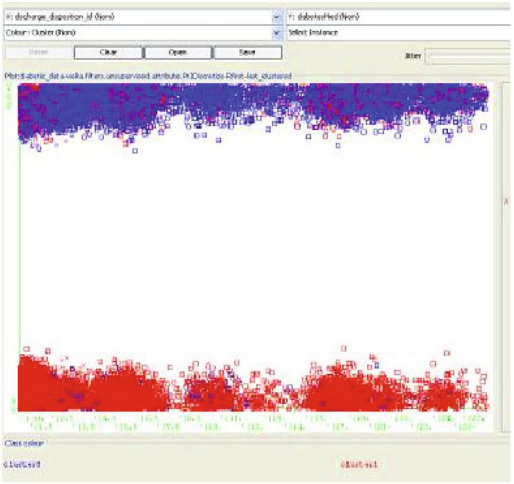


Fig. 6 Clusters in accordance with discharge and diabetes

Caucasian race where no diabetes medicine was provided to patients admitted to hospital.

5 Conclusion

We are in the epoch where big data is playing an imperative role for future outcomes. Through the improved analysis of big data will prove fruitful for scientific discoveries as well as future interventions. However, there exist several technical challenges among researchers to handle complexity, heterogeneity of big data. These

challenges are visible in all application domains, hence transformative technology should be synthesized where benefits from big data can be generalized. In the proposed chapter, a framework to deal with challenges of big data by addressing different data mining techniques for big data analytics is analysed and implemented. The focus of study includes big data with application domain of healthcare databases, where analytics study conducted proves, big data mining is an effective and efficient technique to discover hidden knowledge for better future outcomes. It is understood the patterns and discuss cause of patients admitted in hospital due to diabetes where the eccentric features such as age, sex, race and inpatient and outpatient data were related. The other aspects were also studied, which medication was effective in accordance to race and age of patents admitted in hospital.

The proposed chapter utilizes preprocessing technique to remove inconsistent and missing values, further the data is discretized for utilizing big data mining technique such as clustering to discover hidden patterns from training dataset for paramount future outcomes. The technique involves several benefits to end users where less time is involved and comparative results are found.

The future outcomes of research will be focused on designing and implementation of algorithm for big data. The research will be intervened with application and user domain knowledge process, where the focus will be role of user and application domain for discovering hidden and knowledgeable pattern.

References

1. Mashey, J.: Big Data and the next wave of Infrastress. In: Usenix Technical Conference (1999), http://www.Usemix.org/publications/library/proceedings/usemix99/invited.talks/mashey.pdf
2. Weiss, S.H., Indurkhya, N.: Predictive Data Mining: A Practical Guide. Morgan Kaufmann Publishers, San Francisco (1998)
3. Xindong, W., Gong, Q.W., Wei, D.: Data mining with big data. IEEE Transactions on Knoweledge and Data Engineering 26 (1), 97–107 (2014)
4. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning-Data Mining, Inference, and Prediction. Springer, Heidelberg (2009)
5. Demchenko, Y., Zhao, Z., Grosso, P., Wibisono, A., De Laat, C.: Addressing big data challenges for scientific data infrastructure. In: Proceedings of IEEE 4th International Conference on Cloud Computing Technology and Science, pp. 614–617. IEEE Xplore (2012), doi:10.1109/CloudCom.2012.6427494
6. Chauhan, R., Kaur, H., Alam, A.: Data clustering method for discovering clusters in spatial cancer databases. International Journal of Computer Application 10(6), 9–14 (2010)
7. Manyika, J., Chui, M., Brown, B., Buhin, J., Dobbs, R., Roxburgh, C., Byers, A.: Big data: The next frontier for innovation, competition, and productivity, pp. 1–36. McKinsey Global Institute, USA (2011)
8. Duhigg, C.: The power of habit. The Random House Publishing Group, New York (2012)
9. Hellerstein, J.: Parallel programming in the age of big data. Gigaom Blog (2008), http://gigaom.com/2008/11/09/mapreduce-leads-the-way-for-parallel-programming

10. Ursum, J., Bos, W.H., Van De Stadt, R.J., Dijkmans, B.A., Van, S.D.: Different properties of ACPA and IgM-RF derived from a large dataset: further evidence of two distinct autoantibody systems. Arthritis Research Therapy 11(3), 1439–1443 (2009)
11. Jacobs, A.: The pathologies of big data. ACM Queue 7(6), 1–12 (2009)
12. Ajdacic, G.V., Vetter, S., Müller, M., Kawohl, W., Frey, F., Lupi, G., Blechschmidt, A., Born, C., Latal, B., Rossler, W.: Risk factors for stuttering: A secondary analysis of a large data base. European Archives of Psychiatry and Clinical Neuroscience 260(4), 279–286 (2010)
13. Chen, H., Chiang, R.H.L., Storey, V.C.: Business intelligence and analytics: From big data to big impact. Management Information System Quarterly 36(4), 1165–1188 (2012)
14. Rigaux, P., Scholl, M.O., Voisard, A.: Spatial Databases with Application to GIS. Morgan Kaufmann Publishers, San Francisco (2002)
15. Talia, D.: Parallelism in knowledge discovery techniques. In: Fagerholm, J., Haataja, J., Järvinen, J., Lyly, M., Råback, P., Savolainen, V. (eds.) PARA 2002. LNCS, vol. 2367, pp. 127–136. Springer, Heidelberg (2002)
16. Ng, R.T., Han, J.: Efficient and effective clustering methods for spatial data mining. In: Santiago, C., Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proceedings of 20th International Conference on Very Large Databases, pp. 144–155. Morgan Kaufmann Publishers, USA (1994)
17. Gahegan, M.: Is inductive machine learning just another wild goose (or might it lay the golden egg). International Journal of Geographical Information Science 17(1), 69–92 (2003)
18. Sarndal, C.E., Swensson, B., Wretman, J.: Model Assisted Survey Sampling Series. Springer Series in Statistics, vol. XV. Springer, Heidelberg (1992)
19. Adhikary, J., Han, J., Koperski, K.: Knowledge discovery in spatial databases: Progress and challenges. In: Proceedings of the SIGMOD 1996 Workshop on Research Issues on Data Mining and Knowledge Discovery, Montreal, Canada, pp. 55–70 (1996)
20. Chauhan, R., Kaur, H.: Predictive Analytics and Data Mining: A Framework for Optimizing Decisions with R Tool. In: Tripathy, B.K., Acharjya, D.P. (eds.) Advances in Secure Computing, Internet Services, and Applications, pp. 73–88. IGI Global, USA (2014), http://www.igi-global.com/chapter/predictive-analytics-and-data-mining/99451, doi:10.4018/978-1-4666-4940-8.ch004
21. Fayyad, U.M., Haussler, D., Stolorz, P.: KDD for science data analysis: Issues and examples. In: Simoudis, E., Han, J., Fayyad, U.M. (eds.) Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, pp. 50–56. AAAI Press, Menlo Park (1996)
22. Koperski, K., Han, J., Stefanovic, N.: An efficient two step method for classification of spatial data. In: Poiker, T.K., Chrisman, N. (eds.) Proceedings of the 8th Symposium on Spatial Data Handling, pp. 45–54. International Geographic Union, Simon Fraser University, Canada (1998)
23. Kolatch, E.: Clustering algorithms for spatial databases: A survey. University of Maryland (2001), http://citeseer.nj.nec.com/436843.html
24. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: An introduction to cluster analysis. John Wiley & Sons, Inc., New Jersey (1990)

25. Kaur, H., Chauhan, R., Alam, M. A., Aljunid, S., Salleh, M.: SPAGRID: A spatial grid framework for high dimensional medical databases. In: Corchado, E., Snášel, V., Abraham, A., Woźniak, M., Graña, M., Cho, S.-B. (eds.) HAIS 2012, Part III. LNCS, vol. 7208, pp. 690–704. Springer, Heidelberg (2012)
26. Kaur, H., Chauhan, R., Alam, M.A.: An optimal categorization of feature selection methods for knowledge discovery. In: Zhang, Q., Segall, R.S., Cao, M. (eds.) Visual Analytics and Interactive Technologies: Data, Text and Web Mining Applications, pp. 94–108. IGI Publishers, USA (2010)
27. Kaur, H., Wasan, S.K.: An integrated approach in medical decision making for eliciting knowledge, web-based applications in healthcare & biomedicine. In: Lazakidou, A. (ed.) Annals of Information Systems, vol. 7, pp. 215–227. Springer, Heidelberg (2009)
28. Kaur, H., Chauhan, R., Aljunid, S.: Data mining cluster analysis on the influence of health factors in casemix data. BMC Health Services Research 12(suppl. 1), 03 (2012)
29. Kadhim, M.A., Alam, M.A., Kaur, H.: A multi intelligent agent architecture for knowledge extraction: Novel approaches for automatic production rules extraction. International Journal of Multimedia and Ubiquitous Engineering 9(2), 95–114 (2014)
30. Dash, M., Liu, H.: Feature selection methods for classifications. Intelligent Data Analysis 1, 131–156 (1997)
31. Kohavi, R., John, G.: Wrappers for feature subset election. Artificial Intelligence 12, 273–324 (1997)
32. Boyd, D., Crawford, K.: Critical questions for big data. Information, Communication and Society 15(5), 662–679 (2012)
33. Beyer, M.A., Laney, D.: The Importance of 'Big Data': A Definition. Gartner's definition of big data (2012), http://www.gartner.com/technology/research/big-data
34. Chen, M.S., Han, J., Yu, P.S.: Data mining: An overview from a database perspective. IEEE Transcation on Knowledge Data Engineering 8(6), 866–883 (1996)
35. Lin, J., Dmitriy, V.R.: Scaling big data mining infrastructure: the twitter experience. SIGKDD Explorations 14(2), 6–19 (2012)

Fundamentals of Brain Signals and Its Medical Application Using Data Analysis Techniques

P. Geethanjali

Abstract. In this chapter, the various data analysis techniques devoted to the development of brain signals controlled interface devices for the purpose of rehabilitation in a multi-disciplinary engineering is presented. The knowledge of electroencephalogram (EEG) is essential for the neophytes in the development of algorithms using EEG. Most literatures, demonstrates the application of EEG signals and no much definite study describes the various components that are censorious for development of interface devices using prevalent algorithms in real-time data analysis. Therefore, this chapter covers the EEG generation, various components of EEG used in development of interface devices and algorithms used for identification of information from EEG.

1 Introduction

Recently, electroencephalogram (EEG) is gaining attention and becoming attractive to researchers in various fields of engineering such as computer science, biomedical, electrical and electronics engineering, etc. The electrical activity recorded from the surface of the head is referred as electroencephalogram. The electrical signal from brain activity, enable us to understand the disorderliness of human for the purpose of treatment as well as to develop communication and or control assistive devices to the disabled people. Developing stimuli for generation of electrical signal from the brain to communicate and or control assistive devices and understanding of acquired signals using digital signal processing techniques is crucial in the area of biomedical for computer science researchers [16].

The decoding of brain signals for diagnosis, eliciting and decoding brain signals for communication and control are the thrust areas of research. This chapter is

P. Geethanjai
VIT University, School of Electrical Engineering, Vellore-632014, Tamilnadu, India
e-mail: pganjali78@hotmail.com, pgeethanjali@vit.ac.in

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_8

intended as a preliminary guide for computer science engineers and others who wish to use brain signals in medical applications. In order to understand and decode the signals, the next section of the chapter discusses briefly the different types of brain signals used in medical applications. This chapter briefs about the various stimulus techniques and their issues in eliciting the brain activity. This chapter also discusses interpretation of event related potential (ERP) due to visual stimuli from the EEG for developing communication and or control devices. Also, this chapter discusses on various data analysis in brain signal processing in diagnosis as well as in brain computer interface (BCI). Another growing modality of data mining on brain signal is gaming application. However, this topic is not in the scope of this chapter.

2 Brain Waves

The brain waves are used to diagnose brain damage and various disorders, such as epilepsy, schizophrenia, depression, etc. Therefore, it is necessary to have knowledge on different types of brain waves. The purpose of this section is to explore the different types of brain waves and its characteristics to investigate its potential applications in data analysis.

Hans Berger, a German neuropsychiatrist, who recorded the electrical activity of the brain without opening the skull called an electroencephalograph. EEG measures the electrical signals of the brain using a pair of electrodes on the scalp. EEG reflects mostly the current flow associated with the excitatory and inhibitory postsynaptic potentials of pyramidal neurons in the cerebral cortex [30]. The current penetrates to the scalp through the brain, dura, skull and skin. The electric voltage field generated from individual neurons are very small and do not contribute to detectable EEG. The individual neuron can be examined using microelctrodes. A recordable EEG on the scalp is the superposition of the voltage fields due to a large number of neurons within a volume of tissue generate. This EEG oscillation may be spontaneous or it may include oscillations due to evoked or induced.

2.1 Spontaneous EEG Waves

The basic pattern of spontaneous EEG signal variation due to eye close and open has a common sinusoidal pattern. The amplitude of the normal EEG signal can varies from -100μV and 100μV or 0.5 to 100μV peak-to-peak. The frequency of the signal is ranging from 0 Hz to 40 Hz or more. The electrical frequencies of spontaneous brain signals are divided in the frequency band delta(δ), theta(θ), alpha(α), beta(β), gamma(γ). Each of the frequency bands and their relation to different mental states are given below [32].

Delta Waves are in frequency ranging from 0.5 to 3.5 Hz, are the slowest waves and occurs during sleep state. Presence of delta wave in the awake state, indicates defects in the brain. Delta waves may also occur due to artifacts [15].

Theta Waves are those have frequencies ranging from 3.5 to 7.5 Hz, are not as slow as delta but not very fast. These waves are related to day dreaming and inefficiency. The lowest level of wave indicates a state between sleep and awake, which is useful for hypnotherapy to study the memory process. Theta also arises due to emotions, sensations and intuition. The high level of theta in the awake state of adults is considered abnormal, but for children up to 13 years are considered normal [23].

Alpha Waves are the next wave in order of frequency ranging from 7.5 to 12 Hz are associated with a relaxed state in awake condition. A high amplitude and slower frequency, alpha waves will occur with eyes closed and tranquil state of a person. These waves are strongest at occipital cortex as well as frontal cortex. The amplitude is lower during thinking and calculating state of the brain [28].

Beta Waves ranging from 12 to 30Hz occur when the subject is strongly engaged such as problem solving, judgement, decision making. These waves are faster. The beta frequency band is divided as low beta, ranging from 12 -15Hz, midrange beta ranging from 16 to 20Hz and high beta ranging from 21 to 30 Hz. The low beta band occur when the person is relaxed but focused and integrated. The mid range of beta occurs during thinking and awareness of self and our surroundings. The occurrence reflects alertness, but not with agitation. High beta waves occur with alertness as well as agitation. The mental activity of beta, includes mental activity such as problem solving, planning. These beta waves are seen on both sides of the brain and more significantly in frontal and occipital lobes [24, 20].

Gamma Waves are final brain wave and have frequencies ranging from 31 to 100Hz. These waves can be found in every part of the brain. These waves are associated with memory matching and various types of learning. Gamma waves disappear during deep sleep induced by anesthesia, but appear with the transition back to a awake state [20].

2.2 Event-Related Potential (ERP)

EEG is a conglomeration of different neural sources of activity and it is difficult to segregate the individual neuro-cognitive processes. Subsequently, it becomes difficult to identify specific neural processes. However, it is possible to extract voltage fluctuations in different frequency bands of EEG, when a subject is exposed to external stimuli, such as musical tone, viewing a picture and internal stimuli causing subjects movements. The voltage changes may be associated with the brains response to a stimulus that occurs due to processing of activity at a definite latency time after the specific stimuli. This is called Event Related Potentials (ERP) to

indicate that the potentials are generated due to specific events. These are originally referred as Evoked Potential (EP), since the potentials are evoked by stimuli rather than spontaneous EEG. But ERP is the proposed term to refer potentials that have stable time relationship with a reference event [31]. The changes in voltage within a particular time period are on the order of μV and are too small to be detected in comparison to the EEG. The most common way of extracting the ERP signal involves the averaging samples from the record, a number of EEG epochs, each time-locked into repetitions of the particular event. EEG activity will disappear to zero in averaging procedure, since EEG that vary randomly across epochs and not time-locked to the event. This ERPs are time and phase locked [3, 29].

ERP can be elicited using a wide variety of external visual, auditory and somatosensory stimuli. These sensory stimuli are elicited either by a sequence of relatively high-frequency stimuli or by transient stimuli. The ERP responses due to high frequency periodic stimuli overlap in time and the waveforms are periodic in nature. These ERP responses are termed as steady state evoked potential. But, the responses due to transient stimuli are separated in time and are termed as transient potentials. Further, ERP can be elicited when a subject is preparing for movement, generating movement or as a reaction to errors. But for the purpose cognitive studies from brain signals, researchers use auditory or visual stimuli compared to olfactory, gustatory stimulus [2, 17]. There are three widely used event related responses based on measure of response to visual, auditory and sensation in neuro-cognitive processes.

Visual Evoked Response (VER): The subjects generates response caused by visual stimulus such as alternating checkerboard pattern on a computer screen, flashing of lights. VER are used to detect the abnormality in vision namely demyelination. This is used in brain computer interface (BCI) to develop communication and control devices.

Brainstem Auditory Evoked Response (BAER): The patient generates response hearing a series of auditory clicks in each ear via earphone. BAER are used to detect the abnormality in auditory nerves. These are small evoked potentials and elicited within the initial 10ms of auditory stimulus onset.

Somatosensory Evoked Response (SSER): Short electrical impulses are administered to peripheral sensory nerves such as median or the tibial nerve. SSER are used to detect the abnormality in sensory nerves. The earliest form of potential due to visual, auditory and somatosensory is termed as visual evoked potential (VEP), auditory evoked potential (AEP), and sensory evoke potential (SEP) respectively [1].

There are two classes of ERP components that can be elicited due to sensory, motor or cognitive like retrieval processes. The initial components of ERP occur roughly within the first 100 milliseconds after exposing to stimulus, are referred as sensory or exogenous, which depend on the physical parameters of the eliciting stimulus like luminance. In contrast, later parts of occurrence of the ERPs are referred as cognitive or endogenous, is associated with the mental imagination like mental arithmetic or motor imagery (MI). In addition to exogenous-endogenous

classification, there is another class called mesogenous, that are intermediate between exogenous-endogenous [2, 12].

There are various types of ERP components used in cognitive experiments. The first ERP component reported in the literature is contingent negative variation (CNV) which reflect the preparation of subject for the upcoming target. Subsequently, researchers invented many ERP components such as mismatch negativity (MMN), error-related negativity (ERN), P300/P3, N400/N4, C1, P1/P100, N1/N100, P2/P200, N170, vertex positive potential, N2 (and its sub-component namely N2a, N2b, N2pc), N4/N400, slow cortical potential, etc.

Most of the ERP waveform is described on the basis of their polarity of the peak, latency and scalp distribution. The peak of the components provides a measure of neural activation. The latency provides the timing with reference to the onset of the stimulus. Finally, scalp distribution entails on the overall pattern of activity that may be the summation from different neuronal sources in the brain areas. The peaks of ERP components are labeled Px (negative) or Nx (positive) components according to the amplitude and latency on a different time scale x. The universal convention is to plot positive voltage downward and negative voltage upward. The letter x denotes an ordinal position or the latency position of the peak within the waveform after the onset of the stimulus. The position of the peak is denoted like for an instance P1 which indicate the first positive peak in the waveform or P100 is a positive peak at 100 ms. The labeling with the latency is often 100 times the ordinal position of the peak in the waveform. Also, these labels are not associated with the nature of the stimulus for instance, the P100 and N100 components due to auditory stimuli bear no relationship with the P100 and N100 of visual stimuli [27].

Further, spontaneous brain activity elicits frequency specific oscillations in EEG called sensorimotor rhythm (SMR) to actual movement and motor imagery (MI) movements. The frequency specific changes in the ongoing EEG in human sensorimotor cortex referred as event related desynchronization (ERDs) or power decrease and event related synchronization (ERS) or power increase in a given frequency band. These are time-locked, but not phase locked. These ERD/ERS can be considered as transient post-synaptic response to stimulus [29]. SMR are oscillations in the EEG recorded over the sensorimotor areas at frequency bands 8-12 Hz (alpha/mu) and 18-26 Hz (beta) bands [11]. Subjects can learn to modulate SMR amplitude with an imaginary movement. (e.g. MI of a hand/foot movement) to control output devices [8, 13].

Positive or negative polarizations of the EEG that last from 300ms to several seconds are termed as slow cortical potentials (SCP) and are generated due to movement. They originate in depolarizations of the apical dendritic tree in the upper cortical layers that are caused by synchronous firing. Movement related cortical potential (MRCP) belongs to SCP family consists of pre-movement potential called Bereitschaftspotential (BP)/readiness potential (RP), reafferent potential, pre-motion positivity and motor potential denote potential during the movement and preparation for movement.

In general, SCP, SMR and spontaneous EEG signals are generated due to spontaneous brain activity. The BCI due to, spontaneous brain activity is called as Asynchronous interface and event-related potentials are termed as synchronous interfaces. The synchronous interface requires limited training, but the asynchronous interface requires lengthy training [5, 18].

2.3 Components of EEG Based Systems

The applications of EEG include, monitoring of brain diseases such as epilepsy-seizure and localization of the focus of epileptic seizures, brain tumors, encephalopathies infections, cerbrovasucular and sleep disorders, testing of epilepsy drug effects, identifying damaged area due to stroke, head injury, etc., monitoring alertness, brain death and coma; testing of afferent pathways, controlling anesthesia depth, etc. Other promising applications of brain signal is the control of computer/device and robots [4]. Sample EEG waveforms for a healthy subject, subject with epileptic and motor imagery of the healthy subject is shown in Fig. 1. The sample data which is publicly available is used [6, 7]. Since morphology of EEG is intrusive and it is laborious to diagnose disorders, further, require experts who may prone to subjective judgement in treatment of disorders. It is necessary that the patterns in the EEG waveform are to be identified and classified accurately. Depending on the applications, suitable brain signals are to be utilized. A typical sleep disorder diagnosis, requires the acquisition of multichannel EEG signals, extracting the features and the classification of features. But for control of computer/device can use a wide variety of evoked brain signals elicited due to external stimuli. The basic components of an EEG based application system are:

1. EEG signal acquisition system
2. Feature extraction
3. Feature reduction/translation
4. Classification of wave for the purpose of diagnosis or control

However, for development of control using EP requires an additional component called stimulator. The field of Brain Computer Interface (BCI) employs various paradigms such as Motor Imagery, Steady State Visual Evoked Potentials and P300 for control of computer software based on the real-time analysis of EEG signals. Since BCI is based on visual stimuli, the next section focuses generation of visual stimuli and issues related to generation of stimuli. Subsequent section discusses feature extraction techniques such as Fourier, Wavelet analysis, feature translational techniques like principal component analysis, classification techniques, namely linear discriminant analysis, neural network, which researchers are interested in understanding to extract the information from the source of EEG data within the brain.

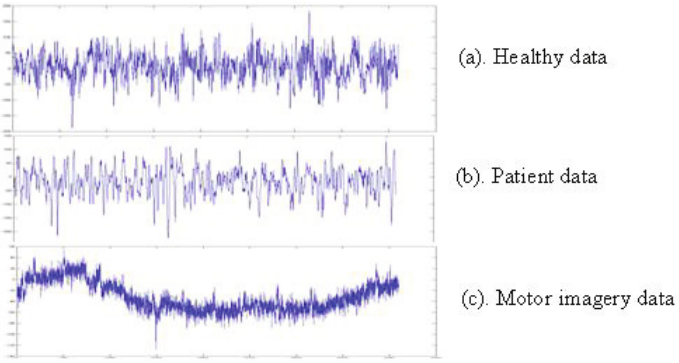


Fig. 1 Example of EEG signals taken from different subjects

3 Generation of Visual Stimuli

Researchers have employed different types of electrophysiological sources such as sensorimotor activity, P300, VEP, SCP, activity of neural cell, response to mental tasks and multiple neuromechanicsms to generate control signals in BCI systems. In visual BCI systems,visual target will flicker at different frequencies. In addition to BCI, visual stimuli are essential to identify the vision disorders. This is referred as photic driving. The visual stimuli can be presented to the subject using external light sources such as LEDs, fluorescent lights, Xe-lights or computer display. However, a variation of the stimulation parameters such as size, position, colour on a computer display is more flexible than external light.

The widely used evoked potential in BCI as well as in cognition is steady state visual evoke potential (SSVEP). One of the types of visual stimuli are flickers of boxes of different colour/characters depending on the application such as wheelchair control, speller BCI. A typical visual stimulus based BCI system is shown in Fig. 2. The BCI system can recognize the intention of the users by decoding elicited SSVEP due to flickers. The flickering stimuli of frequency less than 2Hz elicit transient VEP. But, flickering stimuli of constant intensity evoke strongest SSVEP response in 5-12 Hz, the medium SSVEP response in 12-25Hz and weak SSVEP response in 25-50Hz. The useful frequency range to evoke SSVEP is 6-24 Hz with minimum inter-stimulus gap of 0.2Hz. However, stimuli at low frequency will be uncomfortable for the subject. Therefore, a generation of stimuli with precision and identification of intention from the SSVEP are two challenges of BCI. Generally, the computer has the disadvantages that it may not present the stimuli on display for the programmed duration [10, 19, 20].

In such systems, visual stimuli can be generated using high-resolution timers like Windows Multimedia Timer or using the frame-based method. The accuracy of timer based visual stimuli generation is limited by the other active Windows process.

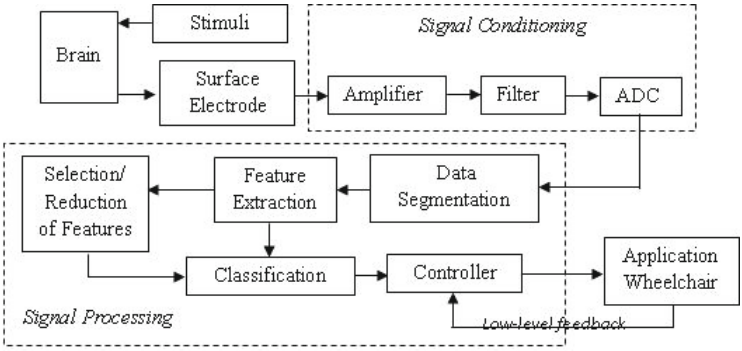


Fig. 2 Block Diagram of BCI system

In frame based approach, the number of frequencies is limited by the refresh rate of the monitor used in order to ensure the stability in flickering frequency. For example, a monitor with a refresh rate of 60Hz will generate only seven commands to elicit SSVEP with 3-9 frames per period. Applications requiring a large number of commands may be difficult in frame based approach with limited refresh rate. The refresh rate of the LCD monitor is mostly 60Hz and has limited visual stimuli. The CRT monitors are widely used due to its higher refresh rate, maximum upto 240Hz. The application of LED monitor is reported less in the literature and evoke SSVEP in 1-90Hz. Therefore, for SSVEP based BCI, the critical challenge is programming number of targets under limited frequencies [10].

Researchers are attempting to different approaches to generate multiple target. Also attempts have been made to use three stimuli to realize 15 targets using frequency and phase mixed information. Similarly, research groups are attempting multiple frequencies sequential coding, modulation of two different frequencies etc. [20].

The generation of visual stimulus at constant intensity with utmost accuracy is critical in order to elicit the response from the brain. This is due to the fact that the complexity of computer displays and timing mechanisms due to operating systems (OS) has increased. A computer with general purpose operating systems have developed scheduling policies, the capacity to manage multitasking concurrency of executing service application, but are affected by unpredictable task scheduling process. However, real-time operating systems are not faster, but predictable. Therefore, development of software like E-Prime, PsychoPy with an optimization algorithm for a general purpose OS may alleviate the timing problem. Further, researchers are also developing a special driver to synchronize stimulus activation and display [14].

4 Processing of Brain Signals

The raw EEG data has been translated using various signal processing method to, identify disorders, generate useful communication and or control commands as discussed in sub-section 2.3. The brain signals are acquired non-invasively using electrodes on the scalp of the subject. 10/20 international system is the most widely used methods for positioning of electrodes. The number of electrodes and position of electrodes depends on the type of application. The signals so acquired are amplified for electronic processing and filtered to remove electrical noise such as power line interference, radio-frequency interference, motion artifacts, etc. The recorded ongoing EEG may be preprocessed to remove artifacts due to ocular, muscular and cardiac activities.

4.1 Preprocessing

In preprocessing stage, artifacts are removed from the amplified brain signal using techniques such as independent component analysis (ICA), principal component analysis, singular value decomposition (SVD), factor analysis, nonnegative matrix factorization (NMF), sparse component analysis (SCA) to improve the signal to noise ratio of the signal. In this subsection widely used ICA and PCA are briefly discussed.

Independent Component Analysis (ICA): Transform the data into a number of independent components. In brain signals, each electrode record signals from number of neural sources. ICA assumes the recorded signals are linear combinations of independent sources.

$$y_i(t) = a_{i1}s_1(t) + a_{i2}s_2(t) + \cdots + a_{in}s_n(t); \quad i = 1, 2, \cdots n \tag{1}$$

Where

s_i is the i^{th} source

a_{in} is the constant

Equation (1) can be expressed in matrix form for the recorded signals from different places as shown below

$$y = As \tag{2}$$

Where

$A$is the mixing matrix

The sources can be found if the mixing matrix is calculated based on the assumption that the source variables s are independent and non-Gaussian. Researchers used ICA for preprocessing of brain signals [9, 26].

Principal Component Analysis (PCA): PCA transforms correlated variables into an uncorrelated variables, which reduces data without much loss of information. The uncorrelated variables are termed as principal components. PCA has been used

in preprocessing, feature extraction, feature reduction of brain signals [21, 22, 25]. A given source can be written as an equation (3)

$$y = P'DP \tag{3}$$

Where

D is the diagonal matrix that contains the square root of the eigenvalues

P is the principal component matrix

The principal component matrix is calculated through singular value decomposition and elements of the matrix is arranged in descending order of the maximum variance. However, the reduction of the dataset is obtained through truncation of the P matrix.

4.2 Feature Extraction

Most of the researchers, have not used data preprocessing and extracted the features directly from the recorded data. Direct application of the measured data for diagnosis/classification usually not preferred because of large dimension. This necessitates the reduction of dimension of data through extraction of the features, that will possibly help in revealing the characteristics, to discriminate the recorded data. A wide variety of techniques were utilised to extract the features of the biomedical data. The extracted features influence the performance of the classification. Features contain information both in time and frequency domain. In each domain of feature, a number of selection criteria exist that strongly influence its efficacy with respect to classification. Time based features are not considered widely and frequency based features have been used. Researchers used several methods like non-parametric methods (Fast Fourier transform), model-based/parametric methods (auto regression, moving average, auto regression-moving average) and time-frequency methods (short time Fourier Transform, wavelet transform, Wigner-Ville distribution) and eigenvector methods (minimum norm) to extract the features from EEG time series.The following subsection discusses briefly the various feature extraction methods used in analysis of recorded EEG data.

Fast Fourier Transform (FFT): The extraction of power from sampled time series data is based on the N-point discrete Fourier transform (DFT) and the equation is given below.

$$y_k = \sum_{n=0}^{N-1} y_n e^{-j2\pi n/N}; \quad k = 0, 1, \cdots N-1 \tag{4}$$

Short Time Fourier Transform (STFT): EEG signals are non-stationary signals and Fourier Transform (FT) is not suitable for analysis. Applying Fourier Transform with a running window function w with time shift δ is referred as STFT. Mathematical definition of STFT for a signal $y(t)$ is given below.

$$STFT(\delta, f) = \int_{-\infty}^{\infty} y(t) w(t-\delta) e^{-i\omega t} dt \tag{5}$$

STFT of a signal represents the power spectrum of the signal estimated at the point (δ,f). The disadvantage of STFT is that there is a trade off between time accuracy and frequency precision. By making the window w narrow, offers improved time resolution, but at the cost of low frequency resolution and vice-versa. Further, the wider window analysis is limited due to the assumption of stationary.

Wavelet Transform(WT): In order to alleviate the above mentioned problem, wavelet based feature extraction has been widely used. The WT decomposes a signal onto a set of basis called wavelet including low frequency to high frequency components. There are two types of WT (1). Continuous wavelet transform (CWT) (2). Discrete wavelet transform (DWT). WT is a convolution of signal $y(t)$ and translated (δ) and scaled (a) mother wavelet function ψ. Mathematical definition of WT for a signal $y(t)$ is given below.

$$WT(\delta, a) = \frac{1}{a} \int_{-\infty}^{\infty} y(t) \psi \left(\frac{(t-\delta)}{a} \right) dt \tag{6}$$

Further, if scale parameter is larger ($\gg 1$), the mother wavelet is expanded along the time axis and corresponds to a low frequency component, similarly low value of the scale parameter (<1) contract the wavelet along the time-axis, which corresponds to high frequencies. The selection of mother wavelet is crucial in WT. The advantage of wavelet is that the high frequency components can be analysed with a higher time accuracy than the lower frequency components of the signal.

The CWT produces a larger number of coefficients due to continuous variation of scaling and shifting. But, discrete wavelet transform (DWT) produces a minimum number of coefficients to recover the original signal by restricting the translation and scaling parameters to power of 2. In addition to selection of mother wavelet, the selection of number of levels of decomposition also crucial in DWT. The selection of level depends on the dominant frequency content of the signal.

Auto Regression (AR): Parametric modeling represents the amplitude of a time series data in a mathematical model. A model that depends only on the previous outputs of the system is called Auto regressive (AR) model. Moving average (MA) model depends only on present values of the system. Further, autoregressive-moving-average model (ARMA) depends on both inputs and outputs of the system. AR model is a common technique used in biomedical applications, where the coefficients of AR model are considered as features. This AR can be adaptive, if the parameters are updated with new data and non-adaptive if the parameters are chosen to a sequence of data. But the determination of AR model that fits a set of sequence is a crucial which depends on the intricacy and the application of the data.The AR model involves determination of coefficients and is expressed by equation (7).

$$y_k = -\sum_{i=1}^{Q} \alpha_i y_{k-i} + \delta_k \tag{7}$$

Where

y_k is the estimated data.

α_i are the AR-coefficients

δ_k is the estimation error

Q is the order of the model (Number of coefficients)

In non-adaptive AR model, Yule-Walker equation, the Burg algorithm may be used to estimate the coefficients.

Eigenvector Methods: These methods are used for estimating the power spectral density (PSD), sharp peak at the expected frequency of noise-corrupted signals. PSD can be estimated using Pisarenko Method, MUSIC method, minimum-norm method based on the eigen-decomposition of the correlation matrix.

In Pisarenkomethod, eigenvector corresponding to minimum eigenvalue is used to calculate the spectrum. The polynomial containing zeros on the unit circle is estimated using the following equation.

$$y(f) = \sum_{n=0}^{m} a_n e^{-j2\pi fk} \tag{8}$$

Where

$y(f)$ is the desire polynomial.

m is the order of eigen filter $y(f)$

a_n is coefficients

In minimum-norm method PSD, is calculated for K dimension noise-subspace as given below.

$$P(f,K) = \frac{1}{|y(f)|^2} \tag{9}$$

In multiple signal classification (MUSIC), method PSD, is calculated for K dimension noise-subspace as given below.

$$P(f,K) = \frac{1}{\frac{1}{K}\sum_{n=0}^{K-1} |y_n(f)|^2} \tag{10}$$

In addition to the above mentioned methods, there are many other feature extraction techniques such as higher order spectral features, phase space plot, correlation dimension, fractional dimension, Lyapunov exponent.

4.3 Feature Selection and Reduction

Generally, the high dimensional extracted features ,i.e. product of number of channels and number of features per channel are overburden to the classifier. Thus, it requires a reduction of the extracted features without much loss of classification accuracy. The two most common methods in the reduction in feature dimension can be achieved in two schemes: feature/channel subset selection and feature projection. Former approach searches all existing features/channel to choose an optimal subset feature/channel which are best for classification. The latter approach creates a subset

of new features using a linear projection or non-linear projection of extracted features. In the field of datamining, feature selection (FS) is categorized into filter based FS and wrapper based (FS). Filter based FS method, select subset of features without classification and utilizing only intrinsic property of the data. The wrapper method selects a feature subset using a criterion function with classification algorithm. Filter methods are faster, but not effective as that of a wrapper.

Dimension reduction of feature vector is more apparent in an EEG signal analysis, as the number of channels are more. The most common method of reduction of feature dimensionality is statistics over the extracted features in the EEG signal segment. The widely used statistical features are mean of the absolute values, average power, standard deviation, the ratio of the absolute mean, maximum, minimum, maximum power level, minimum power level etc. Further, researchers used PCA, not only for denoising and also for feature reduction. Unlike PCA, genetic algorithms (GA) are used for selection of features for reduction of feature vector dimension.

Genetic Algorithm (GA): GA is a heuristic optimization technique based on survival of the fittest. The problem is initialized with random population and select the individual on the basis of fitness with respect to the objective function. The selected individual produces new population by crossover and mutation. This selection of individuals is iterated many times to obtain individual with good fitness.

4.4 Classification

The final stage in the EEG data analysis system is a classifier to identify the feature vector of different classes of EEG signals. Researchers widely investigated the classification performance of various linear, nonlinear classifier, for the classification of extracted features.

In addition to the nature of EEG signals, external factors such as noise cause reasonable variation in the value of a particular feature extracted from the signals over time. Therefore, the classifier should be adequately robust and intelligent to the influence of physiological and physical conditions. A classifier should be able to adapt itself to changes during long-term operation, by exploiting offline and or online training as well as adequately fast to meet real-time constraints. The following subsection discusses few widely used classifiers in EEG data analysis.

Linear Discriminant Analysis (LDA): LDA is a very simple technique that separates the feature vector of different classes using hyperplanes. The hyperplane is obtained by one versus rest strategy, which divides each class from all other classes. The equation of each class, discriminating variable (c_i) is a weighted combination of feature vectors and is given by the following equation.

$$c_i = \sum_j \gamma_j x_j \tag{11}$$

Where

γ_j is the weight for the j_{th} feature variable x_j

Support Vector Machine (SVM): Like LDA, SVM classifier also uses discriminant hyperplane to separate classes. But, the hyperplane separation is not obtained linear and obtained by mapping the feature vectors that lie closest to the boundary to a linearly separable high dimensional space. Hence the name support vector machine. SVM classifiers use a kernel function to find the decision plane at the best location. The selection of parameters to the kernel function is one of the crucial problems in SVM. The commonly used kernel function in BCI is given below.

$$K = e^{\left(\frac{||x-y||^2}{2\sigma^2}\right)} \tag{12}$$

An SVM classifier is insensitive to overtraining and to curse-of-dimensionality. Further, the parameters can be found using quadratic programming software. But it fails if the sample size is more than 1000 and need special-purpose optimization in addition to the lower speed of execution.

Neural Network (NN): Artificial intelligence techniques are another widely used classification technique. Artificial intelligence consists of several techniques, in that NN is widely used in classification applications due to its capability of solving non-linear problems. NN is an interconnection of artificial neurons to obtain non-linear decision hyperplane. NN are application specific and can learn during non-linear mapping between input and output during training. The most commonly used NN is feed-forward networks consist of an input layer, hidden layer and output layer. This multilayer NN is widely trained using a backpropagtion algorithm to form a nonlinear mapping of feature vectors and desired output pattern.

Besides multilayer NN with backpropagation algorithm, other NNs like learning vector quantization neural network, radial basis function NN, etc. are also used in EEG analysis.

Nearest Neighbour: k-Nearest Neighbour (*k*NN), Mahalanobis distance based classifier are some of the distance based classifier. In this *k*NN, Euclidean distance is measured between the feature vectors with all stored prototype vectors. Each class is characterised by prototypes obtained from the training feature vectors. In these conditions, the class of test feature vector is obtained by majority voting amongst *k* nearest neighbours. Mahalanobis distance classifier assigns the feature vector corresponding to the nearest prototype.

The final stage in BCI is the controller, to obtain desired output control commands based on signal patterns and control schemes. Researchers worked with microprocessors, microcontrollers and DSP controllers to actuate the control device.

5 Conclusion

During the past decades, brain computer interface is particularly appealing in different disciplines of engineering in the development of rehabilitation aid for people suffering from disruptive communication between body and brain. A major goal of BCI is to achieve an alternative mode of communication, mobility that leads to

improved quality of life of an individual suffering from spinal cord injury, brain injury etc., which disrupts the normal life. However, in development of BCI, it is vital to understand the signals, methods of generation of stimuli, the data analysis techniques, and development of control mechanisms in case of applications which performs actuation such as wheelchair, teleoperative devices etc. Therefore, a lot of research needs to be performed in various disciplines of engineering for a development of a robust BCI for the boon of the society.

6 Future Work

In literature, researchers used various techniques in diagnosis and in the brain control interface. One of the challenges is finding the relationship between EEG and electrocortocography (ECoG). Even after the strong research, still BCI is unable to use in few individuals. Now researchers are in lookout to identify a new paradigm to make it feasible. Since the brain signals are non-stationary, it is necessary to identify an adaptable feature classifier with the signals.

References

1. Hinrichs, H.: Evoked Potentials. In: Moore Jr., J.E., Maitland, D.J. (eds.) Biomedical Technology and Devices. CRC Press (2013)
2. Fabiani, M., Gratton, G., Coles, M.: Event-related brain potentials. In: Cacioppo, J., Tassinary, L., Bernston, G. (eds.) Handbook of Psychophysilogy. Cambridge University Press, Cambridge (2000)
3. Picton, T.W.: Electrophysiology of Mind: Event-Related Brain Potentials and Cognition. In: Rugg, M.D., Coles, M.G.H. (eds.) Psychophysilogy. Oxford University Press, Oxford (1995)
4. Bickford, R.D.: Electroencephalography. In: Adelman, G. (ed.) Encyclopedia of Neuroscience. Birkhäuser, Cambridge (1987)
5. Roman-Gonzalez, A.: EEG Signal Processing for BCI Applications. In: Hippe, Z.S., Kulikowski, J.L., Mroczek, T. (eds.) Human – Computer Systems Interaction: Backgrounds and Applications 2. AISC, vol. 98, pp. 571–591. Springer, Heidelberg (2012)
6. `http://www.sccn.ucsd.edu/~arno/fam2data/publicly_available_EEG_data.html` (cited April 5, 2013)
7. `http://www.meb.uni-bonn.de/epileptologie/science/physik/eegdata.html` (cited February 16, 2013)
8. Blankertz, B., Sannelli, C., Halder, S., Hammer, E.M., Kubler, A., Muller, K.R., Curio, G., Dickhaus, T.: Neurophysiological predictor of SMR-based BCI performance. Journal of Neuroimage (2010), doi:10.1016/j.neuroimage.2010.03.022
9. Cao, L.J., Chua, K.S., Chong, W.K., Lee, H.P., Gu, Q.M.: A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. Neurocomputing (2003), doi:10.1016/S0925-2312(03)00433-8
10. Cheng, M., Gao, X., Gao, S., Xu, D.: Design and Implementaion of a Brain-Computer Interface with High Transfer Rates. IEEE Trans. Biomedical Engineering (2002), doi:10.1109/TBME.2002.803536

11. Cheyne, D.W.: MEG Studies of Sensorimotor Rhythms: A review. J. Experimental Neurology (2012), doi:10.1016/j.expneurol.2012.08.030
12. Cichocki, A., Washizawa, Y., Rutkowski, T., Bakardjian, H., Phan, A.: Noninvasive BCIs: Multiway Signal-Processing array decompositions. IEEE Computer Society (2008), doi:10.1109/MC.2008.431
13. Cincotti, F., Mattia, D., Aloise, F., Bufalari, S., Schalk, G., Oriolo, G., Cherubini, A., Marciani, M.G., Babiloni, F.: Non-invasive brain-computer interface system: towards its application as assistive technology. J. Brain Res. Bull. (2008), doi:10.1016/j.brainresbull.2008.01.007
14. Garaizar, P., Vadillo, M.A., Lopez-de-lpina, D., Matute, H.: Measuring software timing errors in the presentation of visual stimuli in cognitive neuroscience experiments. Plos One (2014), doi:10.1371/journal.pone.0085108
15. Hammond, D.C.: What is Neurofeedback? Journal of Neurotheraphy (2011), doi:10.1080/10874208.2011.623090
16. Krusienski, D.J., Grosse-Wentrup, M., Galan, F., Coyle, D., Miller, K.J., Forney, E., Anderson, C.W.: Critical issues in State-of-the-art brain-computer interface signal processing. J. Neural Engineering (2011), doi:10.1088/1741-2560/8/2/025002
17. Nijholt, A., Tan, D.: Brain-computer interfacing for intelligent systems. IEEE Computer Society (2008), doi:10.1109/MIS.2008.41
18. Pichiorri, F., De. VicoFallani, F., Cincotti, F., Babiloni, F., Molinari, M., Kleih, S.C., Neuper, C., Kubler, A., Mattia, D.: Sensorimotor rhythm-based brain-computer interface training: the impact on motor cortical responsiveness. J. Neural Eng. (2011), doi:10.1088/1741-2560/8/2/025020
19. Shyu, K., Chiu, Y., Lee, P., Liang, J., Peng, S.: Adaptive SSVEP-Based BCI System With Frequencyand Pulse Duty-Cycle Stimuli Tuning Design. IEEE Trans. Neural Systems and Rehabilitation Engineering (2013), doi:10.1109/TNSRE.2013.2265308
20. Zhang, Y., Xu, P., Liu, T., Hu, J., Zhang, R., Yao, D.: Multiple Frequencies Sequential Coding for SSVEP-Based Brain-Computer Interface. Plos One (2014), doi:10.1371/journal.pone.0029519
21. Guan, C., Thulasidas, M., Wu, J.: High performance P300 speller for brain-computer interface. In: Proc. IEEE Int. Workshop on Biomedical Circuits and System (2004), doi:10.1109/BIOCAS.2004.1454155
22. Hu, J., Si, J., Olson, B.P., He, J.: Principle component feature detector for motor cortical control. In: Proc. 26th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (2004), doi:10.1109/IEMBS.2004.1404123
23. Heinrich, H., Gevensleven, H., Strehl, U.: Annotation: Neurofeedback-train your brain to train behavior. J. of Child Psychology and Psychiatry 48, 3–16 (2007)
24. Rangaswamy, M., Porjesz, B., Chorlian, D., Wang, B.K., Jones, K.A., Bauer, L.O.: Beta power in the EEG of alcoholics. J. Biol. Psychiatry 52, 831–842 (2002)
25. Anderson, C.W., Devulapalli, S.V., Stolz, E.A.: Signal classification with different signal representations. In: Proc. IEEE Workshop on Neural Networks for Signal Processing (1995), doi:10.1109/NNSP.1995.514922
26. Bayliss, J.D., Ballard, H.D.: Single trial P300 recognition in a virtual environment. In: Proc. Int. ICSC Symp. on Soft Computing in Biomedicine, Genova, Italy (1998)
27. Friedman, D., Johnson Jr., R.: Event-Related Potential (ERP) Studies of Memory Encoding and Retrieval: A Selective Review. Microscopy Research and Techniques 51, 6–28 (2000)

28. Lukas, S.E., Mendelson, J.H., Benedikt, R.: Electroencephalographic correlates of marihuana-induced euphoria. Drug and Alcohol Dependence 37, 131–140 (1995)
29. Pfurtscheller, G., Lopes da Silva, F.H.: Event-related EEG/MEG synchronization and desynchronization: Basic principles. Clinical Neurophysiology 110, 1842–1857 (1999)
30. Teplan, M.: Fundamentals of EEG measurement. Measurement Science Review 2, 1–11 (2002)
31. Luck, S.J.: An Introduction to the Event-Related Potential Technique, 2nd edn. MIT Press, Cambridge (2005)
32. Niedermeyer, E., Lopes da Silva, F.H.: Electroencephalography: Basic principles, clinical applications and related fields, 3rd edn. Williams & Wilkins, Philadelphia (1993)

Part III
Big Data Analysis and Cloud Computing

BigData: Processing of Data Intensive Applications on Cloud

D.H. Manjaiah, B. Santhosh, and Jeevan L.J. Pinto

Abstract. Cloud computing, rapidly emerging as a new computation paradigm, provides agile and scalable resource access in a utility-like fashion, especially for the processing of big data. The need to store, process, and analyze large amounts of data makes enterprise customers to adopt cloud computing at scale. Understanding processing of data intensive applications on cloud is key to designing next generation cloud services. Here we aimed to discuss a close-up view about Cloud Computing, Big Data and processing of big data on cloud as well as the state-of-the-art techniques and technologies we currently adopt to deal with the Big Data problems on cloud.

1 Introduction

Big Data has been one of the current and future research frontiers. Gartner defined big data as "Big data is high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making". He listed big data in "Top 10 Strategic Technology Trends For 2013" [1] and "Top 10 Critical Tech Trends For The Next Five Years" [2].Big Data is a collection of very huge data sets with a great diversity of types made it difficult to process by using state-of-the-art data processing

D.H. Manjaiah
Department of Computer Science, Mangalore University, Mangalore
e-mail: manju@mangaloreuniversity.ac.in

B. Santhosh
Department of Computer Science, AIMIT St Aloysius College, Mangalore
e-mail: santhosh.aloysius@gmail.com

Jeevan L.J. Pinto
Department of Computer Science, Srinivas Institute of Management Studies, Mangalore
e-mail: jeevanpinto@rediffmail.com

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_9

approaches or traditional data processing platforms. In 2012, Gartner gave a more detailed definition as: "Big Data are high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization". More generally, a data set can be called Big Data if it is formidable to perform capture, curation, analysis and visualization on it at the current technologies. With diversified data provisions, such as sensor networks, telescopes, scientific experiments, and high throughput instruments, the datasets increase at exponential rate [3, 4]. The off-the-shelf techniques and technologies that we used to store and analyse data cannot work efficiently and satisfactorily. The challenges arise from data capture and data curation to data analysis and data visualization. In many instances, science is lagging behind the real world in the capabilities of discovering the valuable knowledge from massive volume of data. Based on precious knowledge, we need to develop and create new techniques and technologies to excavate Big Data and benefit our specified purposes. Big Data has changed the way that we adopt in doing businesses, managements and researches. Data-intensive science especially in data-intensive computing is coming into the world that aims to provide the tools that we need to handle the Big Data problems. Data-intensive science [5] is emerging as the fourth scientific paradigm in terms of the previous three, namely empirical science, theoretical science and computational science. Thousand years ago, scientists describing the natural phenomenon was only based on human empirical evidences, so we call the science at that time as empirical science. It is also the beginning of science and classified as the first paradigm. Then, theoretical science emerged hundreds years ago as the second paradigm, such as Newtons Motion Laws and Keplers Laws. However, in terms of many complex phenomenon and problems, scientists have to turn to scientific simulations, since theoretical analysis is highly complicated and sometimes unavailable and infeasible. Afterwards, the third science paradigm was born as computational branch. Simulations in large of fields generate a huge volume of data from the experimental science, at the same time, more and more large data sets are generated in many pipelines. There is no doubt that the world of science has changed just because of the increasing data-intensive applications. The techniques and technologies for this kind of data-intensive science are totally distinct with the previous three. Therefore, data-intensive science is viewed as a new and fourth science paradigm for scientific discoveries [6].

2 Cloud Computing and Big Data

The development of virtualization technologies have made supercomputing more accessible and affordable. Powerful computing infrastructures hidden in virtualization software make systems to be like a true physical computer, but with the flexible specification of details such as number of processors, memory and disk size, and operating system. The use of these virtual computers is known as cloud computing [7], which has been one of the most robust Big Data techniques [8, 9].

The name of cloud computing comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. It entrusts remote services with a users data, software and computation. The combination of virtual machines and large numbers of affordable processors has made it possible for internet-based companies to invest in large-scale computational clusters and advanced data-storage systems. Cloud computing not only delivers applications and services over the Internet, it also has been extended to infrastructure as a service, for example, Amazon EC2, and platform as a service, such as Google AppEngine and Microsoft Azure. Infrastructure vendors provide hardware and a software stack including operating system, database, middleware and perhaps single instance of a conventional application. Therefore, it shows out illusion of infinite resources without up-front cost and fine-grained billing. It leads to the utility computing, i.e., pay-as-you-go computing. Surprisingly, the cloud computing options available today are already well matched to the major themes of need, though some of us might not see it. Big Data forms a framework for discussing cloud computing options. Depending on special need, users can go into the marketplace and buy infrastructure services from providers like Google and Amazon, Software as a Service (SaaS) from a whole crew of companies starting at Salesforce and proceeding through NetSuite, Cloud9, Jobscience and Zuora-a list that is almost never ending. Another bonus brought by cloud environments is cloud storage which provides a possible tool for storing Big Data. Cloud storage have good extensibility and scalability in storing information. Cloud computing is a highly feasible technology and attract a large number of researchers to develop it and try to apply to Big Data problems. With respect to consumer and producer perspective- For consumers, big data is about using large datasets from new or diverse sources to provide meaningful and actionable information about how the world works. For producers, big data is the technology necessary to handle the large, diverse datasets. Producers characterize big data in terms of volume, variety and velocity. How much data is there, of what types, and how quickly can you derive value from it etc [20].

2.1 Benefits for Big Data on Cloud Adoption [21]

Big data applications can benefit the most from the clouds advantages, and building a business case around business agility or high ROI is easier than just making an argument for cost savings.

Cost reduction: Cloud computing offers a cost-effective way to support big data technologies and the advanced analytics applications that can drive business value. Enterprises are looking to unlock datas hidden potential and deliver competitive advantage. Big data environments require clusters of servers to support the tools those process the large volumes, high velocity and varied formats of big data. IT organizations should look to cloud computing as the structure to save costs with the clouds pay-per-use model.

Reduce overhead: Various components and integration are required for any big data solution implementation. With cloud computing, these components can be automated, reducing complexity and improving the IT teams productivity.

Rapid provisioning / time to market: Provisioning servers in the cloud is as easy as buying something on the Internet. Big data environments can be scaled up or down easily based on the processing requirements. Faster provisioning is important for big data applications because the value of data reduces quickly as time goes by.

Flexibility/scalability: Big data analysis, especially in the life sciences industry, requires huge compute power for a brief amount of time. For this type of analysis, servers need to be provisioned in minutes. This kind of scalability and flexibility can be achieved in the cloud, replacing huge investments on super computers with simply paying for the computing on an hourly basis.

3 Big Data Processing Challenges in Cloud Computing

Big data in cloud computing environment consists of large number of applications that produce, manipulate, or analyze data in the range of hundreds of megabytes (MB) to petabytes (PB) and beyond. According to Lee et al [10], the field of data intensive computing constitute "the technologies, the middleware services, and the architectures that are used to build useful high-speed, wide area distributed systems".

Processing of data intensive applications on cloud has different challenges in each sub process. Typically, the analysis process is shown In Fig. 1. The following section gives a brief information about challenges for each sub process.

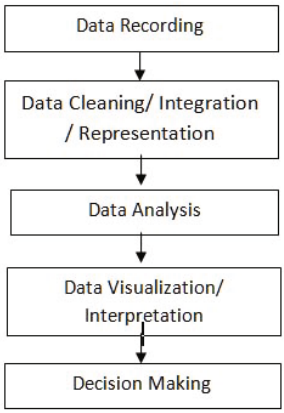


Fig. 1 Knowledge discovery process [19]

3.1 Data Capture and Storage

The worlds technological capacity to store information has roughly doubled about every three years since the 1980s. In many fields, like medical and financial data often be deleted just because there is no enough space to store these data. These valuable data are generated and captured at high cost, but ignored finally. Big Data has changed the way we capture and store data, including data storage device, data storage architecture, data access mechanism. The accessibility of Big Data is on the top priority of the knowledge discovery process.

To store Big Data on cloud there are four main types of cloud storage: Personal Cloud Storage, public cloud storage, private cloud storage and hybrid cloud storage. Personal cloud storage is also known as mobile cloud storage and personal cloud storage is a subset of public cloud storage that applies to storing an individual's data in the cloud and providing the individual with access to the data from anywhere. It also provides data syncing and sharing capabilities across multiple devices. Apple's iCloud is an example of personal cloud storage.

In Public Cloud Storage where the enterprise and storage service provider are separate and there aren't any cloud resources stored in the enterprise's data center. The cloud storage provider fully manages the enterprise's public cloud storage. Private Cloud Storage is a form of cloud storage where the enterprise and cloud storage provider are integrated in the enterprise's data center. In private cloud storage, the storage provider has infrastructure in the enterprise's data center that is typically managed by the storage provider. Private cloud storage helps resolve the potential for security and performance concerns while still offering the advantages of cloud storage. Hybrid Cloud Storage is a combination of public and private cloud storage where some critical data resides in the enterprise's private cloud while other data is stored and accessible from a public cloud storage provider.

Cloud storage may be broadly categorized into two major classes of storage: unmanaged storage and managed storage. In unmanaged storage, the storage service provider makes storage capacity available to users, but defines the nature of the storage, how it may be used, and by what applications. The options a user has to manage this category of storage are severely limited. However, unmanaged storage is reliable, relatively cheap to use, and particularly easy to work with. Most of the user-based applications that work with cloud storage are of this type. Unmanaged cloud storage providers include 4Shared, Adrive, Badongo, Box.net etc.Managed cloud storage is mainly meant for developers and to support applications built using Web services. Managed cloud storage is provisioned and provided as a raw disk. It is up to the user to partition and format the disk, attach or mount the disk, and make the storage assets available to applications and other users. Managed cloud storage providers include AWS Amazon - Simple Storage Service(S3), EMC2 Atmos, Google Storage for Developers IBM Smart Business Storage Cloud, Nirvanix, Rackspace Cloud.

Big Data in cloud storage should be accessed easily and rapidly for further analysis, fully or partially break the restraint: CPU-heavy but I/O-poor. In the past decades, the persistent data were stored by using hard disk drives (HDDs) [10].

In general, HDDs had much slower random I/O performance than sequential I/O performance, and data processing engines formatted their data and designed their query processing methods to work around this limitation. But, HDDs are increasingly being replaced by SSDs today, and other technologies such as PCM are also around the corner. These current storage technologies cannot possess the same high performance for both the sequential and random I/O simultaneously, which requires us to rethink how to design storage subsystems for Big Data processing systems. Direct-attached storage (DAS), network-attached storage (NAS), and storage area network (SAN) are the enterprise storage architectures that were commonly used. However, all these existing storage architectures have severe drawbacks and limitations when it comes to large-scale distributed systems. Aggressive concurrency and per server throughput are the essential requirements for the applications on highly scalable computing clusters, and todays storage systems lack the both. Optimizing data access is a popular way to improve the performance of data-intensive computing; these techniques include data replication, migration, distribution, and access parallelism. Data-access platforms, such as CASTOR, dCache, GPFS and Scalla/Xrootd, are employed to demonstrate the large scale validation and performance measurement [11]. Data storage and search schemes also lead to high overhead and latency, distributed data-centric storage is a good approach in large-scale wireless sensor networks (WSNs). Shen, Zhao and Li proposed a distributed spatialtemporal similarity data storage scheme to provide efficient spatialtemporal and similarity data searching service in WSNs. The collective behaviour of individuals that cooperate in a swarm provide approach to achieve self-organization in distributed systems [12, 13].

3.2 Data Transmission

Cloud data storage is popularly used as the development of cloud technologies. The network bandwidth capacity is the bottleneck in cloud and distributed systems, especially when the volume of communication is large. The secure data transmission can be achieved by protocols like IPSec, SSL over web but certain issues like Throughput and complexity in coding needs to be addressed.

3.3 Data Curation

Data curation is aimed at data discovery and retrieval, data quality assurance, value addition, reuse and preservation over time. This field specifically involves a number of sub-fields including authentication, archiving, management, preservation, retrieval, and representation. The existing database management tools are unable to process Big Data that grow so large and complex. This situation will continue as the benefits of exploiting Big Data allowing researchers to analyse business

trends, prevent diseases and combat crime. Though the size of Big Data keeps increasing exponentially, current capability to work with is only in the relatively lower levels of petabytes, exabytes and zettabytes of data. The classical approach of managing structured data includes two parts, one is a schema to storage the data set, and another is a relational database for data retrieval. For managing large-scale datasets in a structured way, data warehouses and data marts are two popular approaches. A data warehouse is a relational database system that is used to store and analyze data, also report the results to users. The data mart is based on a data warehouse and facilitate the access and analysis of the data warehouse. A data warehouse is mainly responsible to store data that is sourced from the operational systems. The preprocessing of the data is necessary before it is stored, such as data cleaning, transformation and cataloguing. After these preprocessing, the data is available for higher level online data mining functions. The data warehouse and marts are Standard Query Language (SQL) based databases systems. NoSQL database, also called Not Only SQL, is a current approach for large and distributed data management and database design. Its name easily leads to misunderstanding that NoSQL means not SQL. On the contrary, NoSQL does not avoid SQL. While it is true that some NoSQL systems are entirely non-relational, others simply avoid selected relational functionality such as fixed table schemas and join operations. The mainstream Big Data platforms adopt NoSQL to break and transcend the rigidity of normalized RDBMS schemas. For instance, Hbase is one of the most famous used NoSQL databases as shown in Fig. 2.

However, many Big Data analytic platforms, like SQLstream and Cloudera Impala, series still use SQL in its database systems, because SQL is more reliable and simpler query language with high performance in Big Data real-time analytics.

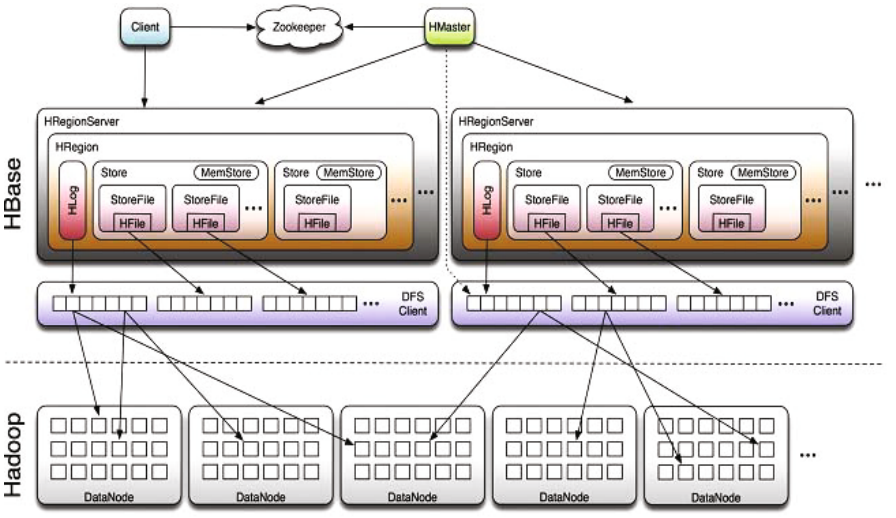


Fig. 2 Hbase NoSQL database system architecture. Source: Apache Hadoop.

To store and manage unstructured data or non-relational data, NoSQL employs a number of specific approaches. Firstly, data storage and management are separated into two independent parts. This is contrary to relational databases which try to meet the concerns in the two sides simultaneously. This design gives NoSQL databases systems a lot of advantages. In the storage part which is also called key-value storage, NoSQL focuses on the scalability of data storage with high-performance.

In the management part, NoSQL provides low-level access mechanism in which data management tasks can be implemented in the application layer rather than having data management logic spread across in SQL or DB-specific stored procedure languages. Therefore, NoSQL systems are very flexible for data modelling, and easy to update application developments and deployments .Most NoSQL databases have an important property. Namely, they are commonly schema-free. Indeed, the biggest advantage of schema-free databases is that it enables applications to quickly modify the structure of data and does not need to rewrite tables. Additionally, it possesses greater flexibility when the structured data is heterogeneously stored. In the data management layer, the data is enforced to be integrated and valid. The most popular NoSQL database is Apache Cassandra. Cassandra, which was once Facebook proprietary database, was released as open source in 2008. Other NoSQL implementations include SimpleDB, Google BigTable, Apache Hadoop, MapReduce, MemcacheDB, and Voldemort. Companies that use NoSQL include Twitter, LinkedIn and NetFlix.

3.4 Data Analysis

The first impression of Big Data is its volume, so the biggest and most important challenge is scalability when we deal with the Big Data analysis tasks. In the last few decades, researchers paid more attentions to accelerate analysis algorithms to cope with increasing volumes of data and speed up processors following the Moores Law. For the former, it is necessary to develop sampling, on-line, and multire solution analysis methods . In the aspect of Big Data analytical techniques, increment algorithms have good scalability property, not for all machine learning algorithms. Some researchers devote into this area . As the data size is scaling much faster than CPU speeds, there is a natural dramatic shift in processor technologyalthough the clock cycle frequency of processors is doubling following Moores Law, the clock speeds still highly lag behind. Alternatively, processors are being embedded with increasing numbers of cores. This shift in processors leads to the development of parallel computing. For those real-time Big Data applications, like navigation, social networks, finance, biomedicine, astronomy, intelligent transport systems, and internet of thing, timeliness is at the top priority. Assigning the timeliness of response when the volume of data to be processed is very large is a big challenge for stream processing involved by Big Data. It is right to say that Big Data not only have produced many challenge and changed the directions of the development of the hardware, but also in software architectures. That is the swerve to cloud computing,

which aggregates multiple disparate workloads into a large cluster of processors. In this direction, recently the distributed computing is being developed at high speed.

Data security surfaces with great attentions. Significant security problems include data security protection, intellectual property protection, personal privacy protection, commercial secrets and financial information protection [14]. Most developed and developing countries have already made related data protection laws to enhance the security. Research groups and individuals need to carefully consider the legislation of where they store and process data to make sure that they are in compliance with the regulations. For Big Data related applications, data security problems are more awkward for several reasons. Firstly, the size of Big Data is extremely large, channelling the protection approaches. Secondly, it also leads to much heavier workload of the security. Otherwise, most Big Data are stored in a distributed way, and the threats from networks also can aggravate the problems.

3.5 Data Visualization

The main objective of data visualization [15, 16] is to represent knowledge more intuitively and effectively by using different graphs. To convey information easily by providing knowledge hidden in the complex and large-scale data sets, both aesthetic form and functionality are necessary. Information that has been abstracted in some schematic forms, including attributes or variables for the units of information, is also valuable for data analysis. This way is much more intuitive than sophisticated approaches. Online marketplace eBay, have hundreds of million active users and billions of goods sold each month, and they generate a lot of data. To make all that data understandable, eBay turned to Big Data visualization tool: Tableau, which has capability to transform large, complex data sets into intuitive pictures. The results are also interactive. Based on them, eBay employees can visualize search relevance and quality to monitor the latest customer feedback and conduct sentiment analysis. For Big Data applications, it is particularly difficult to conduct data visualization because of the large size and high dimension of Big Data. However, current Big Data visualization tools mostly have poor performances in functionalities, scalability and response time. What we need to do is rethinking the way we visualize Big Data, not like the way we adopt before. For example, the history mechanisms for information visualization also are data-intensive and need more efficient approaches. Uncertainty can lead to a great challenge to effective uncertainty-aware visualization and arise in any stage of a visual analytics process. New framework for modelling uncertainty and characterizing the evolution of the uncertainty information are highly necessary through analytical processes. The shortage of talent will be a significant constraint to capture values from Big Data. In the United States, Big Data is expected to rapidly become a key determinant of competition across many sectors. However, this area demands for deep analytical positions on Big Data could exceed the supply being produced on current trends by 140,000 to 190,000 [17] positions. Furthermore, this kind of human resource

is more difficult to educate. It usually takes many years to train Big Data analysts which have the intrinsic mathematical abilities and related professional knowledge. It is foreseeable that there will be another hot competition about human resources in Big Data developments. After reviewing a number of challenges, the optimists take a broad view of challenges and hidden benefits. They have enough confidence that we have the capabilities to overcome all the obstacles as new techniques and technologies are developed. There are many critiques and negative opinions from the pessimists. Some researchers think Big Data will lead to the end of theory, and doubt whether it can help us to make better decisions. Whatever, the mainstream perspectives are most positive, so a large number of Big Data techniques and technologies have been developed or under developing.

4 Big Data Cloud Tools: Techniques and Technologies

To capture the value from Big Data, we need to develop new techniques and technologies for analyzing it. Scientists have developed a wide variety of techniques and technologies to capture, curate, analyze and visualize Big Data. Even so, they are far away from meeting variety of needs. These techniques and technologies cross a number of discipline, including computer science, economics, mathematics, statistics and other expertises. Multidisciplinary methods are needed to discover the valuable information from Big Data. We will discuss current techniques and technologies for exploiting data intensive applications.

We need tools (platforms) to make sense of Big Data. Current tools concentrate on three classes, namely, batch processing tools, stream processing tools, and interactive analysis tools. Most batch processing tools are based on the Apache Hadoop infrastructure, such as Mahout and Dryad. The latter is necessary for real-time analytic for stream data applications. Storm and S4 are good examples for large scale streaming data analytic platforms. The interactive analysis, processes the data in an interactive environment, allowing users to undertake their own analysis of information. The user is directly connected to the computer and hence can interact with it in real time. The data can be reviewed, compared and analyzed in tabular or graphic format or both at the same time. Googles Dremel and Apache Drill are Big Data platforms based on interactive analysis.

4.1 Processing Big Data with MapReduce

MapReduce is a software framework for easily writing applications which processes vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. MapReduce has become a dominant parallel computing paradigm for Big Data, i.e., colossal datasets at the scale of tera-bytes or higher. Ideally, a MapReduce system

should achieve a high degree of load balancing among the participating machines, and minimize the space usage, CPU and I/O time, and network transfer at each machine.

The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for scheduling the jobs component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the DFS (Distributed File System). DFS is fault tolerant and designed to be deployed on low cost hardware.

Job Tracker is a job configuration which specifies the map (M1, M2, M3 etc), combine and reduce function, as well as the input and output path of the data. JobConf is the primary interface for a user to describe a MapReduce job for execution. The framework tries to faithfully execute the job as described by JobConf. The JobTracker will first determine the number of splits (each split is configurable, 16-64MB) from the input path, and select some TaskTracker based on their network proximity to the data sources, then the JobTracker send the task requests to those selected TaskTrackers. Task Tracker will initiate the assigned mapper by extracting the input data from split, as a child process in a separate java virtual machine.The architecture of Haloop is as shown in Fig. 3.

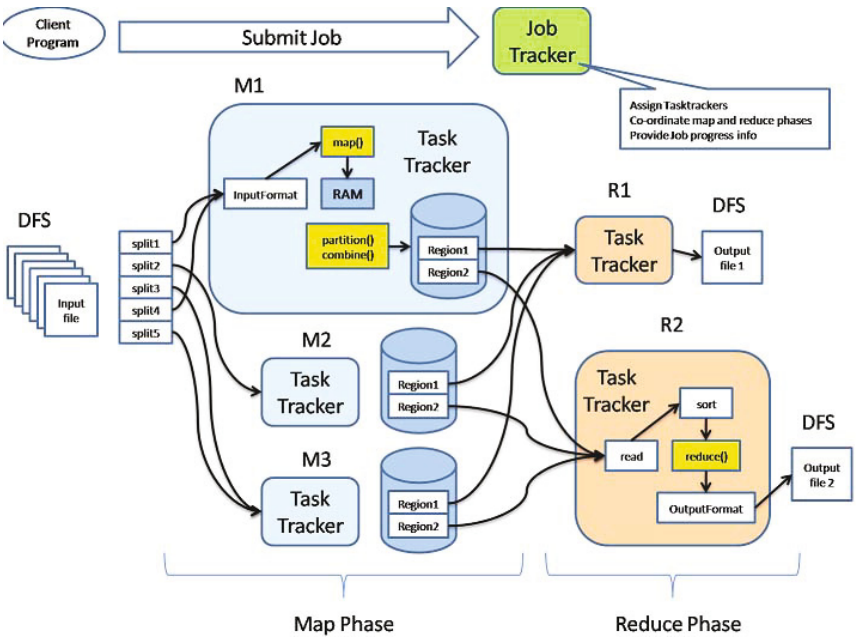


Fig. 3 MapReduce phases

4.2 Processing Big Data with Haloop

Bu, Howe, and Ernst proposed a modified variant of Hadoop that can process data iteratively on large clusters. It extends MapReduce and also provides various capabilities to it including caching mechanisms, loop aware task scheduler and other features In Hadoop the main programming model is known as MapReduce which is suitable for processing big data. Hadoop is a distributed file system that supports processing huge amount of data in terabytes or more in distributed environments such as cloud computing.Haloop is improved MapReduce framework. The architecture of Haloop includes loop aware task scheduler, and caching mechanisms besides other common requirements as there in Hadoop. The architecture of Haloop is as shown in Fig. 4. Bu, Howe, and Ernst proposed a modified variant of Hadoop that can process data iteratively on large clusters. It extends MapReduce and also provides various capabilities to it including caching mechanisms, loop aware task scheduler and other features In Hadoop the main programming model is known as MapReduce which is suitable for processing big data. Hadoop is a distributed file system that supports processing huge amount of data in terabytes or more in distributed environments such as cloud computing.Haloop is improved MapReduce framework. The architecture of Haloop includes loop aware task scheduler, and caching mechanisms besides other common requirements as there in Hadoop. The architecture of Haloop is as shown in Fig. 4.

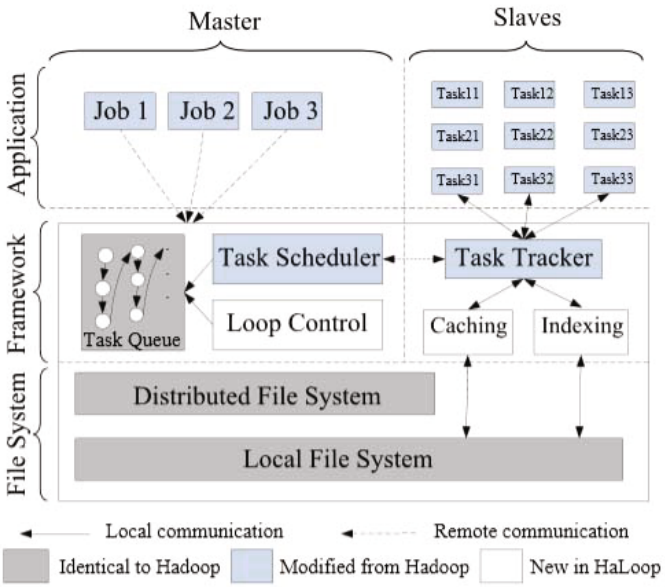


Fig. 4 The HaLoop framework, a variant of Hadoop MapReduce framework

As shown in the Fig. 4 it has three layers, file system, framework and application. There are two file systems in the file system layer. They are local file system and distributed file system. The local file system takes care of local storage and the distributed file system takes care of storage in multiple machines in order to manage big data processing.

In the framework layer, task tracker and task scheduler are the two important components. The task tracker is able to communicate with local file system that makes use of indexing and caching features in order to improve the processing performance. The task scheduler is different from Hadoop here as it is supported by a loop control. It does mean that the task scheduler is loop aware for high performance. Task queue is used to maintain queue of tasks for processing efficiently [18]. Caching is very important in the framework which reduces number of hits to the file system. Caching and indexing are the two important features used by task tracker in order to show high performance of processing big data. The task scheduler is in master node while the task tracker is in slave node. The master node takes jobs and gives to slave nodes. The slave nodes process the data and give result back to master node. This way data is processed in parallel to support big data. In the application layer, the master node manages jobs while the slaves manage tasks. Actually the jobs are divided into tasks and the tasks are performed by slave nodes. The master nodes only delegate the jobs to slave nodes in different ways. For instance the master can invoke slaves in either sequential or parallel fashion or it may use combination of both based on the workload. The master node communicates with the framework in order to get jobs done. With respect to iterative work there is a fundamental difference between Hadoop and Haloop that is Haloop is loop aware while the Hadoop is not. This fact is visualized in Fig. 5.

As can be seen in Fig. 5, it is evident that the Hadoop is not loop aware. It continues iterative work until jobs are completed while the Haloop is loop aware that knows how many times the loop has to be iterated. Other main difference is that the Haloop has caching mechanisms that help in improving speed of big data processing further. It focuses on reducer input cache, and reducer output cache for both caching and indexing. Haloop also has mapped input cache in order to map input data and tasks. PageRank is used by both frameworks [18].

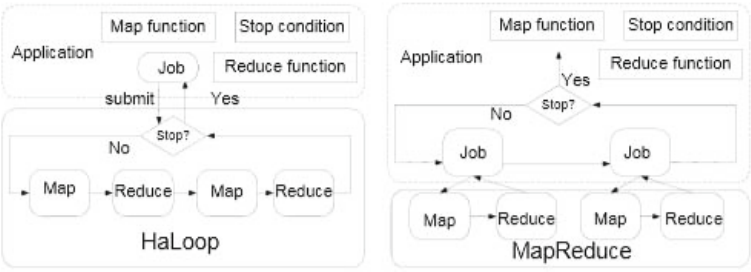


Fig. 5 Difference between Hadoop and Haloop in iterative processing

4.3 Cloudant

Cloudant's distributed database as a service (DBaaS) allows developers of fast-growing web and mobile apps to focus on building and improving their products, instead of worrying about scaling and managing databases on their own. Highly available, durable and feature-rich, the data store is built for scaling, optimized for concurrent reads & writes, and handles a wide variety of data types including JSON, full-text and geospatial.

4.4 Xeround

Xeround (pronounced zeh-round) is a management tool for deploying easily scalable MySQL databases across a variety of cloud providers and platforms. Its software allows for high availability and scalability and it works across a variety of cloud providers including AWS, Rackspace, Joyent and HP, as well as on OpenStack and Citrix platforms.

4.5 StormDB

Unlike other databases in the cloud, StormDB runs its fully distributed, relational database on bare-metal servers, meaning there is no virtualization of machines. StormDB officials claim this leads to better performance and easier management because users do not have to choose the size of virtual machine instances their database runs on. Despite running on bare metal, customers do share clusters of servers, although StormDB promises there is isolation among customer databases. StormDB also automatically shards databases in its cloud.

4.6 SAP

Enterprise software giant SAP is now playing in the cloud with HANA, a platform built on in-memory technology. Its cloud database from HANA complements the company's other on-premise database tools, including Sybase, and is available in Amazon Web Services' cloud. HANA includes other non-database apps too, including business management tools and application development.

4.7 Rackspace

Rackspace's database comes in either a cloud or managed hosted offering via Cloud Databases, which is the name of its product. Rackspace emphasizes the container-based virtualization of its Cloud Databases, which it says allow for higher performance of the database service compared to if it was run entirely on virtualized infrastructure. Cloud Databases also incorporates a SAN storage network and it's

based on an OpenStack platform. Rackspace has also a NoSQL database in its cloud from provider Cloudant.

4.8 MongoLab

In the NoSQL world, there are a variety of database platforms to choose from, including MongoDB. MongoLab gives users access to MongoDB on a variety of major cloud providers, including AWS, Azure and Joyent. Like other gateway-type services, MongoLab also integrates with various platform as a service (PaaS) tools at the application tier. MongoLab run on either shared or dedicated environments, with the latter being slightly more expensive.

4.9 Microsoft Azure

Microsoft uses its SQL Server technology to provide a relational database, allowing customers to either access a SQL database on its cloud, or hosted SQL server instances on virtual machines. Microsoft also emphasizes hybrid databases that combine data both on a customer's premise and with the Azure cloud through SQL Data Sync. Microsoft has a cloud-hosted NoSQL database service named Tables as well, while Blobs (binary large object storage), are optimized for media files such as audio and video.

4.10 Google Cloud SQL

Google's cloud database service is centered on two major products: Google Cloud SQL, which Google describes as a MySQL-like fully relational database infrastructure, and Google BigQuery, an analysis tool for running queries on large data sets stored in its cloud.

4.11 Garantia Data

Garantia offers a gateway service for users to run open source Redis and Memcached in-memory NoSQL databases services in AWS's public cloud. Using Garantia's software allows for automatic configuration of these open source data platforms by helping developers scale nodes, create clusters and architect for fault tolerance.

4.12 EnterpriseDB

EnterpriseDB focuses on the open source PostgreSQL databases, but its real claim to fame is its ability to work with Oracle database applications. With EnterpriseDB's Postgres Plus Advanced Server, organizations can use applications written for

on-premise Oracle databases through EnterpriseDB, which runs in clouds from Amazon Web Services and HP. It has binary replication and scheduled backups as well.

4.13 Amazon Web Services

Amazon Web Services has a variety of cloud-based database services, including both relational and NoSQL databases. Amazon Relational Database Service (RDS) runs either MySQL, Oracle or SQL Server instances, while Amazon SimpleDB is a schema-less database meant for smaller workloads. On the NoSQL side, Amazon DynamoDB is its solid-state drive (SSD)-backed database that automatically replicates workloads across at least three availability zones. AWS CTO Werner Vogels says DynamoDB is AWS's fastest growing service in AWS history. Amazon also offers a variety of auxiliary data management services, such as its newly announced data warehouse named Redshift, as well as Data Pipeline, which helps users integrate data from multiple sources for easier management.

5 Conclusion

This is an era of Big Data which is the frontier for innovation now, competition and productivity, a new wave of scientific revolution is just begin. This chapter gives a brief overview on Big Data, how to process data intensive applications, current techniques and technologies. One way of looking at Big Data is that it represents the large and rapidly growing volume of information that is mostly untapped by existing analytical applications and data warehousing systems. The actual technologies used will depend on the volume of data, the variety of data, the complexity of the analytical processing workloads involved, and the responsiveness required by the business. It will also depend on the capabilities provided by vendors for managing, administering, and governing the enhanced environment. These capabilities are important selection criteria for product evaluation. Human resources, capital investments and creative ideas are fundamental components of development of Big Data.

References

1. Savitz, E.: Gartner: top 10 strategic technology trends for 2013. White Paper, Information Technology Services, Queen's University (2012)
2. Savitz, E.: Gartner: 10 critical tech trends for the next five years. White Paper, Information Technology Services, Queen's University (2012)
3. Szalay, A., Gray, J.: Science in an exponential world. Nature Publication 440(7083), 413–414 (2006)
4. Lynch, C.: Big data: how do your data grow? Nature Publication 455(7209), 28–29 (2008)
5. Bell, G., Hey, T., Szalay, A.: Beyond the data deluge. Science Magzine 323(5919), 1297–1298 (2009)

6. Hey, T., Tansley, S., Tolle, K.: The fourth paradigm: data-intensive scientific discovery. Microsoft Research. 1–250 (2009)
7. Furht, B.: Cloud Computing Fundamentals. In: Escalante, A. (ed.) Handbook of Cloud Computing, pp. 1–20. Springer, USA (2011)
8. Sakr, S., Liu, A., Batista, D., Alomari, M.: A survey of large scale data management approaches in cloud environments. IEEE Communications Surveys and Tutorials 13(3), 311–336 (2011)
9. Schadt, E.E., Linderman, M.D., Sorenson, J., Lee, L., Nolan, G.P.: Computational solutions to large scale data management and analysis. Nature Reviews Genetics 11(9), 647–657 (2010)
10. Kasavajhala, V.: Solid state drive vs. hard disk drive price and performance study. White Paper, Dell PowerVault Storage System, pp. 1–13 (2012)
11. Muhleisen, H., Dentler, K.: Large scale storage and reasoning for semantic data using swarms. IEEE Computational Intelligence Magzine 7(2), 32–44 (2012)
12. Vettiger, P., Cross, G., Despont, M., Drechsler, U., Durig, U., Gotsmann, B., Haberle, W., Lantz, M.A., Rothuizen, H.E., Stutz, R., Binnig, G.K.: The millipede nanotechnology entering data storage. IEEE Transactions on Nano Technology 1(1), 39–55 (2002)
13. Han, J., Haihong, E., Le, G., Du, J.: Survey on nosql database. In: Proceedings of 6th IEEE International Conference on Pervasive Computing and Applications, pp. 363–366. IEEE Xplore (2011)
14. Smith, M., Szongott, C., Henne, B., Voigt, G.: Big data privacy issues in public social media. In: Proceedings of 6th IEEE International Conference on Digital Eco Systems Technologies, pp. 1–16. IEEE Xplore (2012)
15. Simoff, S.J., Böhlen, M.H., Mazeika, A. (eds.): Visual data mining: theory, techniques and tools for visual analytics. LNCS, vol. 4404. Springer, Heidelberg (2008)
16. Keim, D.A., Panse, C., Sips, M., North, S.C.: Visual data mining in large geospatial point sets. IEEE Computer Graphics and Applications 24(5), 36–44 (2004)
17. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Rox-burgh, C., Byers, A.H.: Big data: The next frontier for innovation, competition, and productivity. White Paper, McKinsey Global Institute, pp. 1–140 (2012)
18. McGowan, K.: Big data: the next frontier for innovation, competition, and productivity. SAS Solutions on Demond, pp. 1–16 (2013)
19. Fayyad, U.M., Shapiro, G.P., Smyth, P.: From data mining to knowledge discovery: An overview. In: Fayyad, U.M., Shapiro, G.P., Smyth, P., Utaurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 1–34. AAAI Press, UK (1996)
20. Eli, C.: Intersection of the cloud and big data. IEEE Cloud Computing 1(1), 84–85 (2014)
21. Pasalapudi, S.K.: Trends in cloud computing: big data's new home, White paper, Oracle profit (2014)

Framework for Supporting Heterogenous Clouds Using Model Driven Approach

Aparna Vijaya, V. Neelanarayanan, and V. Vijayakumar

Abstract. Cloud computing has gained the popularity of todays IT sector because of the low cost involved in setup, ease of resource configuration and maintenance. The increase in the number of cloud providers in the market has led to availability of a wide range of cloud solutions offered to the consumers. These solutions are based on different cloud architectures and usually are incompatible with each other. It is very hard to find a single provider which offers all services the end users need. Cloud providers offer proprietary solutions that force cloud customers to decide the design and deployment models as well as the technology at the early stages of software development. One of the major issues of this paradigm is; the applications and services hosted with a specific cloud provider are locked to their specific implementation technique and operational methods. Hence, moving these applications and services to another provider is a tedious task. This situation is often termed as vendor lock-in. Hence a way to provide portability of applications across multiple clouds is a major concern. According to the literature very few efforts have been made in order to propose a unified standard for cloud computing. DSkyL provides a way for reducing the cloud migration efforts. This chapter aims to sketch the architecture of DSkyL and the major steps involved in the migration process.

1 Introduction

Cloud computing is gaining in popularity, both in the academic world and in software industry. One of the greatest advantages of cloud computing is on-demand self service, providing dynamic scalability (elasticity) [1]. Cloud computing offers a vast amount of resources, available for end users on a pay-as-you-go basis.

Aparna Vijaya · V. Neelanarayanan · V. Vijayakumar
School of Computing Science and Engineering VIT University, Chennai Campus
e-mail: {aparna.v,neelanarayanan.v,vijayakumar.v}@vit.ac.in

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_10

These advantages can be exploited to prevent web-applications from breaking during peak loads, and supporting an unlimited amount of end users. The opportunity to choose between several cloud providers is dimmed by complexity of cloud solution heterogeneity. The main issue is technological inconsistencies for provisioning between providers, e.g., AWS offer command line interface (CLI) tools, while Rackspace offers web-based Application Programming Interface (APIs). Although most providers offer APIs; these are different as well, they are inconsistent in layout and entities. The result of this is vendor lock-in, the means used to provision an application to a given cloud must be reconsidered if it is to be re-provisioned on another provider. Because of incompatibilities, users that develop applications on a specific platform may encounter significant problems when trying to deploy their application in a different environment. But from the developers perspective, the fullest advantage of using cloud services lies when there is flexibility to switch between cloud service providers, to change platforms and when they provide interoperability.

This chapter aims to introduce the first version of DSkyL which is designed to harmonize the inconsistencies between providers, with a model-based approach.

2 Background

During the past few years there has been a tremendous increase in the number of cloud providers in the market due to the cloud hype. Deployment of an application in cloud is often influenced by the technologies and interfaces provided by each of this cloud vendor. To achieve portability between these different cloud providers, it takes a lot of effort and time. This section gives an overview on the terminologies associated to our work in providing portability which has been used in the later section of this chapter.

2.1 Cloud Computing

Cloud computing is a terminology associated to the delivery of on-demand computing resources ranging from applications to data centers over the internet on a pay as you use basis. Cloud Computing is about providing computation and resources as services, such as virtual instances and file storage, rather than products. The characteristics of cloud computing are as follows.

- On-demand self-service: With on-demand self-service, consumers can achieve provisioning without any human interaction. On-demand means dynamic scalability and elasticity of resource allocation, self-service so that users do not need to manually do these allocations themselves. Considering an online portal for viewing university results, for most of the year it will have low usage demands, but before and during the result announcement days it will have to serve a

humongous amount of requests. With on demand self-service this system could automatically be given more resources such as memory, computation power or even increase the number of instances to handle peak loads. The previous example has planned (or known) peak intervals, so even though automatic handling is appealing it could be solved by good planning. But sometimes predicting peak loads can be difficult, such as when a product suddenly becomes more popular than first anticipated. Twitter is a good example of a service that can have difficulties in estimating the amount of user demand and total amount of incoming requests. On a normal basis the service does not have to cope with huge amount of requests, but on rare and unpredictable scenarios cloud computing can help to tackle this with its characteristics such as on-demand self-service. With on-demand self-service allocation will automatically scale upwards as popularity increases and downwards as resources become superfluous.

- Resource pooling: Physical and virtual resources are pooled so they can be dynamically assigned and reassigned according to consumer demand. Users do not need to be troubled with scalability as this is handled automatically. This is a provider side characteristic which directly influence on-demand self-service. There is also a sense of location independence; users can choose geographical locations on higher abstracted levels such as country or state.
- Rapid elasticity: Already allocated resources can expand vertically to meet new demands, so instead of provisioning more instances (horizontal scaling) existing instances are given more resources such as Random-access Memory (RAM) and Central Processing Unit (CPU). On unexpected peak loads, the pressure will be instantly handled by scaling upwards.
- Measured service: Resources allocated in the cloud can be monitored by cloud providers, accommodating end users with monitoring data on resources they rent. It can be used for statistics for users, for instance to do analytical research on product popularity or determine user groups based on geographical data or browser usage. The providers themselves use this information to handle on demand services, if they notice that an instance has a peak in load or has a noticeable increase in requests they can automatically allocate more resources or capabilities to leave pressure. Measuring can also help providers with billing, if they for instance charge by resource load and not only amount of resources allocated.

2.1.1 Service Models

Service models define the different layers in cloud computing. There are three main architectural service models namely:

- Infrastructure-as-a-Service (IaaS) is similar to more standard solutions such as Virtual Private Servers (VPS), and is closest to standard hosting solutions.

- Platform-as-a-Service (PaaS) is built to guide and assist developers through abstractions. It helps developers by detaching them from configuration of operating system.
- Software-as- a-Service (SaaS) provides complete applications as services and end products. Google products such as Gmail, Google Apps and Google Calendar are examples of SaaS applications. By utilizing the characteristics of cloud computing, SaaS applications achieve cloud computing advantages.

2.1.2 Deployment Models

Deployment models define where and how applications are deployed in a cloud environment, such as publicly with a global provider or private in local data centers. There are four main deployment models.

- Public cloud is a deployment model infrastructure which is open to the public, so that companies can rent services from cloud providers. Cloud providers own the hardware and rent out IaaS and PaaS solutions to users. Examples of such providers are Amazon with AWS and Google with GAE. The benefit of this model is that companies can save costs as they do not need to purchase physical hardware or manpower to build and maintain such hardware. It also means that a company can scale their infrastructure without having to physically expand their data center.
- Private cloud is similar to classical infrastructures where hardware and operation is owned and controlled by organizations themselves. This deployment model has arisen because of security issues regarding storage of data in public clouds. With private cloud organization can provide data security in forms such as geographical location and existing domain specific firewalls, and help comply requirements set by the government or other offices.
- Community cloud is similar to private clouds but run as a combination between several organizations. Several organizations share the same aspects of a private cloud (such as security requirements, policies, and compliance considerations), and therefore share infrastructure.
- Hybrid cloud benefits by storing sensitive information in a private cloud while computing in a public cloud. For instance a government can establish by law how and where some types of information must be stored, such as privacy law. To sustain such laws a company could store data on their own private cloud while doing computation on a public cloud. In some cases such laws relates only to geographical location of stored data, making it possible to take advantage of public clouds that can guarantee geographical deployment within a given country.

2.2 Model Driven Engineering

Model-Driven Architecture (MDA) is a way of designing software with modeling provided by the Object Management Group (OMG). When working with MDA it is common to first create a Computation Independent Model (CIM), then a Platform-Independent Model (PIM) and lastly a Platform-Specific Model (PSM). The five different steps involved in the transformation of a model is as follows [1]:

- The developer first familiarizes himself with the business organization and the requirements of the domain to create a CIM. This should be done without any specific technology. The physical appearance of CIM models can be compared to use case diagrams in UML, where developers can model actors and actions (use cases) based on a specific domain.
- The next step aims at developing a PIM using descriptions and requirements from the CIM with specific technologies. Example of such Platform Independent Models can be class diagrams in UML used to describe a domain on a technical level.
- Convert the PIM into PSM. The next step is to convert the models into something more concrete and specific to a platform. Examples of such models can be to add language specific details to PIM class diagram such as types (String, Integer) for variables, access levels (private, public), method return types and argument types. From this it is possible to determine that a PSM is more specific to a platform than PIM, such as programming language or environment.
- Generate code form PSM. A PSM should be specific enough that code can be generated from the models. For instance can class diagrams be generated into entities, and additional code for managing the entities can be added as well.
- Deploy. The final step is based on deploying the PSM to a running product.

2.3 Necessity for Using Multiple Clouds

A large number of small and medium businesses are now moving to cloud to reduce their infrastructure and operational cost and also to increase the scalability. It takes a longer time for them to implement their application from the scratch and it even requires lot of extra effort for supporting the necessary changes that needs to be done to support different cloud vendors. Unfortunately every cloud platform is different from another i.e. they are heterogeneous. Because of the incompatibilities, the applications developed on a specific platform may encounter significant problems when deployed in a different environment. This gives rise to the familiar problem of vendor lock-in. This lock-in can be present on several layers of the cloud computing stack as described by NIST. On the SaaS layer during data migration, on the PaaS layer applications and services should have the possibility of migration (would often involve recompilation and generation of code), and on the IaaS complete virtual machine instances should be able to be moved from one provider to another. Assume

that there is a SaaS Enterprise Resource Planning (ERP) system. This system holds important data about accounts, invoices and clients. For years they have been satisfied with a vendor, but now the vendor have increased their prices or the quality of their service has decreased or they have changed their terms and conditions in ways that are unacceptable to business. The data can be downloaded from the old vendor's system. Having the ability to download your data only provides partial fulfillment of migration; what if the data is in a particular format which no other system can read? Then you have to rewrite the ERP to support the existing format or have to pay the next vendor to extend their ERP so it can deal with the existing format. This becomes a tedious task. Consumers should be able to easily shift from one cloud providers to another and should be free to choose the one that provides better services they need in terms of quality and/or cost. The ability of consumers to switch from one cloud platform provider to another can be critical for their business, especially when a cloud providers operation is unexpectedly terminated. The ability to run applications targeting multiple private, public, or hybrid clouds allows exploiting the peculiarities of each cloud solution and hence optimizing performance and availability.

2.4 Challenges for Migration

Applications developed using traditional software design methods and frameworks will face certain challenges while migration onto cloud. All the applications as well as the supporting network infrastructure may not be suitable for migration onto the cloud. Typical challenges while migrating applications to cloud are depicted below:

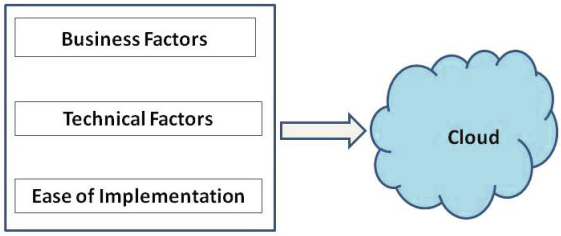


Fig. 1 Factors affecting migration of application to cloud

The key business factors to be considered while moving to the cloud are:

- Cost: The cost model for a software development project is termed as a combination of capital expenditure and operational expenditure. Organizations generally budget for peak loads incurring higher capital expenditure. However, these costs while being high are budgeted and predictable. Moving to an operational cost model through the adoption of the cloud would mean paying for resources as per

usage. This model implies unpredictable operational costs especially for those applications with varying demand for e.g. public facing websites. Therefore, it is important for organizations to estimate application usage and operational costs before moving to the cloud. Further, migration costs need to be understood and factored in before making the decision to move into the cloud. Failure to do this could negate the cost savings that are sought to be derived from the adoption of the cloud.

- Data security: Security of data is a key concern while migrating applications to the cloud. Applications that have very sensitive and confidential information would be better off being behind the corporate firewall. Data with greater security tolerance however could be ported onto the cloud. Technical mechanisms for data security in the cloud are still evolving and security of data is still the top most inhibiter of cloud adoption [3].
- Regulations: Geopolitical issues especially for governments and financial institutions should be carefully evaluated before making the transition to the cloud. In the Indian context this is especially relevant as most cloud data centres are not located within the country. It is also important to ensure that local regulations relevant to each organization should be adhered to before deciding to move to the cloud [4].
- Provisioning: One of the key benefits of the cloud is the quick provisioning of resources. Applications that need to be quickly available and scaled up rapidly based on demand are ideal candidates for the cloud. Most organizations have business requirements that need to be supported by quick provisioning of IT data, e.g. an organization running a limited period online marketing campaign. Several applications are seasonal in nature as well for example HR and payroll applications, which need resources to be processed only during certain periods. These sorts of applications can make use of the ability of the cloud to quickly provision resources.

Some of the key technical aspects to be considered are:

Existing infrastructure: The traditional IT architecture is optimized to cater to the current demand in the sector. Moving to the cloud would necessitate a change in the IT architecture. With applications moving into the cloud, the way IT is delivered to end users would undergo a radical change. Some end user support would be dependent on the cloud service provider response. Hence, organizations would have to concentrate on building vendor management competencies [5].

Security architecture: Application security and controls would need to change to adapt to the cloud ecosystem. New types of mechanisms would be required to secure data in transit and at rest. Identity and access management mechanisms would need to be adapted to cloud deployments. Further, data encryption mechanisms and key management for the cloud are yet to mature [6].

Complexity: Simple applications can be easily migrated to the cloud and the amount of effort required to move such applications may not be not too significant. These applications can be directly migrated to Software as a Service [7] (SaaS) applications already available from various vendors. E.g. e-Mail applications can be directly ported onto cloud offerings like Office365, Google Apps or Lotus Live.

Similarly, moving a simple web server to an Infrastructure as Service [7] (IaaS) platform may not require as much effort. Migration of complex applications however, needs elaborate planning and testing prior to implementation. Legacy applications and existing enterprise applications could require code changes to work on the cloud.

Service Level Agreements (SLAs): Another key aspect to consider before migrating to the cloud is whether cloud service providers are able to provide SLAs that the business needs. This is quite essential considering the limited control organizations have over applications on the cloud. SLAs need to address the concerns of availability, confidentiality and integrity of the application. Further, it should clearly outline service provider responsibilities and penalties for failure to meet agreed service levels [4].

Ease of Implementation: For developing traditional desktop application or web-based applications we normally follow certain frameworks and templates. It is easy for developers to understand those frameworks and develop traditional applications. In case a developer is stuck at any phase of the software development, he can find numerous web references and documentations solving the raised issue. Here, developers even have the advantage of re-using the codes of existing applications. Even though cloud application development is a trending buzz now, it is difficult for organizations to get resources with expert knowledge in cloud application development. Even developers find it difficult to understand the new frameworks and methods used for developing cloud-based applications. Since, we do not have well-formed documentations for the frameworks and APIs supporting cloud, it becomes a tedious and time consuming task for developers to solve any issues raised on this.

3 Techniques for Modernization of Application to Cloud

Many organizations write their software from scratch specifically for hosting in cloud, while others want to keep their existing application software and run it on a cloud platform. Migration across different cloud is not automatic and the amount of effort required could be significantly high since the architecture of the application might not always support the different cloud providers. There are often differences in various infrastructures, the programming models and the libraries available in traditional and cloud model. Like any software development project, migration projects require a lot of planning and proper process to follow. Normally migration approaches where one needs to completely rewrite the application or development of a completely new piece of software is less likely to be chosen due to the cost and time involved. While a cloud migration can include numerous challenges and raise security concerns, cloud hosting enables an organization to reduce capital expenditures and operating costs. Its high availability, multi-tenancy and effective resource allocations are the other key features which most organizations look forward for.

Some approaches for modernization of applications are [7]:

- Wrapping is building a new interface called wrapper, to act as an interpreter between legacy system and target system. Minimum changes are applied to legacy component which implies less cost and lower risk. But wrapping cannot be applied straight forwardly when the functionality of legacy components is not always suitable for the requirements of target system. Further, the lower level structure, such as platform, language, and software architecture, is unchanged in wrapping. It would not result in long-term benefits and improving the maintainability.
- Migration switches legacy components to current environment and retains the original data and functionality. It offers benefits such as higher flexibility and ease of maintenance.
- Maintenance is continuously maintaining the legacy components to meet incoming requirements. The degree of change is less than migration. But the maintainability of system is gradually decreasing. Hence, the cost of maintenance is becoming more and more expensive.
- Redevelopment completely throws away the legacy components and develops new components to meet the requirement of target system. It tends to a high level risk of failure and a higher cost of redevelopment.
- Replace component(s) with cloud supported components(s). Here one or more components are replaced by cloud supported components / services. Using Google App Engine instead of a local MySQL database is an example of this migration type.
- Cloudify is the complete migration of the application. The application functionality is implemented as a composition of services running on the cloud. Cloudification requires the migration of data and business logic to the cloud.

3.1 Existing Technologies

1. SMART: Service-Oriented Migration and Reuse Technique (SMART), proposed by SEI at CMU. It helps organizations analyse legacy systems to determine whether their functionality can be reasonably exposed as services. SMART gathers a wide range of information about legacy components, the target environment, and potential services to produce a service migration strategy for the organization. This approach consists of establishing the context of migration, by discussions with developers using the Services Migration Interview Guide. Once the migration system is feasible, the authors of this approach propose to select from the initial list the candidates services and their inputs and outputs. It is platform/vendor/tool independent [8].
2. ARTIST: ARTIST proposes an approach that starts with the characterization of application from two points of view; technical and business of the current legacy application and how the company expects those aspects to be in the future to provide a gap analysis. It is then followed by a technical feasibility analysis

and business feasibility analysis. Based on this gap analysis using a technical feasibility tool and a business feasibility tool, the migration tasks and their effort are recorded, and it also simulates the impact of the modernized application in the organization [9].

3. Using Mash Ups: The mashup migration strategy has six steps.The first two activities are the modelling of the target enterprise business and the analysis of legacy systems and infrastructure. These activities lead to two main steps which maps model requirements to legacy components and services identifications and models mashup server architecture with Domain Specific Kits , which abstracts legacy components. Both the mapping and architecture design activities might cause a loopback to MODEL and ANALYZE activities to re-reconsider some of the decisions. As a result of these major activities, target system service dependency graph has been constructed and mashup architecture will been designed. Defining the Service Level Agreements, including non-functional and contextual properties, is the next step that will be followed by implementation and deployment activities [10].
4. Service Oriented Modelling and Architecture (SOMA): SOMA is an iterative and incremental method on how to plan and implement migration which was developed by IBM. SOMA is structured into seven main phases. Business Modelling describes the customers business. Solution Management adapts the SOMA method to the project needs. The Service Identification phase has aims at finding appropriate services. Three different activities are accomplished to identify service candidates. One of these methods, called Existing Asset Analysis, analyzes existing legacy systems in order to identify services. During Service Specification, services, components and communications are formally defined. Service Realization describes the components in more detail. Parts of the legacy systems are analyzed deeply to determine if they can be used as service implementation. According to SOMA, appropriate legacy functions can be migrated into service implementations in two ways such as transformation and wrapping. Transformation converts the legacy function into a new technology which can be implemented as service component directly. Wrapping encapsulates a legacy function with an interface without changing it. During Service Implementation the whole system is implemented. New components are coded, existing legacy functions are transformed, wrappers are written and the services are composed. In the Service Deployment phase the system is deployed to the customer [11].
5. Five Phased Approach: This approach follows an iterative waterfall model for migration of application to cloud. In real time software environment, there are situations when the defects are detected in much later phase of SDLC. The iterative waterfall model provides the advantage of going back to the phase where the defect was detected and correcting the defect. This approach contains five phases. In the Feasibility Study phase basically the financial and technical feasibility of cloud migration is studied. It also involves the analysis of the existing application. A detail cost/benefit analysis is performed in this phase. It is also determined whether the migration is not feasible due to high cost, resource constraints, or technical reasons. Requirement Analysis and Planning

phase involves, a detailed assessment of the existing application environment with a view to understand the applications that are appropriate for moving into the cloud. In the Migration phase, the chosen application is ported to the cloud and tested in a structured manner. Testing and deployment phase also called as Go-live phase, in this phase the live production data will be ported onto the cloud. This phase involves a higher degree of monitoring and support. Post migration monitoring is a key requirement in cloud migration. This monitoring and maintenance phase deals with monitoring the cloud application in terms of performance, availability and security [2].

6. Six Phased Approach: It includes phases such as Legacy system understanding (LSU) in which knowledge about the legacy applications, source code characteristics, identifying dependencies and legacy system architecture is obtained. Target system understanding (TSU) phase facilitates the representation of the desired architecture. TSU represents two aspects of the target architecture: (i) the functional aspect, and (ii) the technical aspect. Migration Feasibility Determination checks the feasibility of the migration from technical, economical and organizational perspectives. Candidate Service Identification is categorized into top-down and bottom-up approaches. In the top-down approach, initially a business process is modelled based on the requirements and then the process is subdivided into sub-processes until these can be mapped to legacy functions. The bottom-up approach utilizes the legacy code to identify services using various techniques such as information retrieval, concept analysis, and business rules recovery and source code visualization. The implementation phase is related to the execution of the migration of the legacy applications. Deployment and provisioning phase tests to determine if the expected functionality is exposed correctly as a service when legacy application is migrated as a service [12].
7. Oracles Approach: The various phases involved in migration are: Assessment - includes collecting information related to project management, potential cost of the migration, migration approaches, and tools to use. Analysis and design phase -determining the implementation details on the target environment. Migration tasks involve data and application migration. Testing in a migration project usually comprises tasks such as data verification, testing of migrated business logic testing of application interaction with the new platforms and scripts. Optimization phase addresses the issues in performance, hardware configuration, software installation and configuration, initial data loading and facilitating backup and recovery options. Post-Production supports troubleshooting any issues that may come up immediately after the new environment goes live [13].
8. Cloud Step: It is a process aimed at supporting organizations and application developers in making cloud selection and migration decisions. The process relies on the creation of template-based profiles describing key characteristics of the organization, its target legacy application and candidate cloud providers. These profiles are then cross-analyzed to identify and possibly resolve critical constraints (either technical or non-technical) that may affect application migration to the cloud [7].

4 Portability Issues in Cloud Applications

To proceed with our investigation into the problem of cloud application portability we need to identify specific points of conflict which arises when attempting to deploy an application to multiple cloud platforms.

In other words, we need to identify which aspects of a cloud application may be addressed differently by cloud platforms. In this section we discuss the following few potential conflict points: programming languages and frameworks, platform-specific services and platform specific configuration files [14, 15].

- Programming languages and/or frameworks - The specific programming languages and frameworks that an application has been built will be a major concern for cross platform deployment. Each cloud platform supports certain languages, frameworks, and versions thereof. For example, while Google App Engine (GAE) provides support for Java, amazon uses DotNet.
- Platform specific services - An important characteristic of several cloud platforms is that they provide certain services via specific APIs. A service can be considered as high-level functionality that the provider can use without the need to implement it from scratch. Such examples are analytic tools for handling data sets, APIs for image manipulation etc. Developers can drastically reduce the application development time by using such platform services. Instead of programming every bit of functionality from the ground up, they can integrate it into their application by binding to the respective platform APIs. Each platform provider may offer a wider or smaller range of such specific services. Let us assume that a developer chooses a certain platform in order to develop and deploy the above mentioned application. A portability issue arises when the application needs to be ported to a different cloud platform. There are two cases:

  1. The target platform doesnt provide the full set of services that the application uses. In this case the developer would need to recreate the missing functionality from scratch on the new target platform.
  2. The target platform supports the services that the application uses but provides different APIs in order to use them. In this case the developer would need to modify the application code and align it with the APIs of the new target platform.

  In both cases, the application cannot directly be ported across multiple platforms. The developer needs to modify the application in order to be deployable to different platforms.
- Platform specific configuration files - Similar to the configuration files that traditional software applications require in order to instruct the hosting environment

on how to execute the applications, cloud platforms may require configuration files. For example Google App Engine uses the appengine-web.xml file. The process of adapting the configuration files to each target cloud platform adds to the overall overhead of cross-platform deployment of a cloud application.

5 Proposed Approach

Organizations and the development community is hesitant to create their systems using certain specific technologies and later being charged with unfair rates for its usage. They are even reluctant to choose a technology which may turn out to be inadequate or inefficient in near future. In order to take advantage of the flexible cloud architecture, the applications have to be specifically developed for the chosen cloud platform. For example, in order to offer great elasticity, Google App Engine a PaaS provider imposes a specific programming style and it has its own way to manage data, and thus an application developed specifically for GAE and the data associated with it may not be easily ported to a different PaaS provider. Even if a developer want to host an application in his/her cloud later, additional effort would be required to rebuild the application, redeploy it and migrate all the data. A re-engineering process required to change the cloud provider can be costly and time consuming.

Model-driven engineering (MDE) is a unification of initiatives that aims to improve software development by employing high-level domain models in the design, implementation, and testing of software systems. MDE provides benefits in terms of productivity, ease of communication and software maintenance, allowing software engineers to work in a higher abstraction level. In our proposed approach, we extend MDE by incorporating domain models which facilitates migration to multiple clouds.

Platform independent images offer uniform access for cloud applications independently from the cloud provider and the technologies it supports. These can be controlled and configured by the user. The platform independent image can be beneficial in four scenarios:

- When application development and hosting is not particular to a cloud provider.
- When a cloud service provider aims to improve their services by providing new application development APIs and hosting methodologies.
- When a cloud service provider aims to scale their services (SaaS) by offering the services of a new cloud providers resources (IaaS).
- When a user needs more processing power for his application, he might want to host the application in multiple clouds. The equations are an exception to the prescribed specifications of this template.

For example if you are a developer at an ISV (Independent Software Vendor) that offers CRM application on one of the most popular SaaS platforms available. Now if you want to sell your application to those customers using alternative platforms

and if some of those potential customers want to have the application hosted in a different environment; the application have to be re-written to run on those environments and build a new cloud hosting relationship. As an ISV, this would be very expensive. Such a scenario limits the openness at the platform level. Platforms which use a proprietary programming language, explicitly tied to a single vendors implementation will force the customers to use a specific platform thereafter.

DSkyL [16] targets developers and CRM/ERP vendors by providing them a solution that supports the development of cloud-based CRM/ ERP software applications which are portable across multiple cloud platforms. The concept of feature models is used in DSkyL to drive the development and migration planning process. These feature models are platform-independent that captures the essence of the application to produce a domain specific code (DSL). The deployable file is generated from the cloud configuration file, the final DSL code and the source codes corresponding to the added features. This deployable file is a platform independent image and it is specific to each cloud service provider. Porting an application from one cloud platform to another requires transformations with same set of models, which represent the functionalities and different cloud configuration files generated from the SLAs.

DSkyL is an eclipse plug-in for modeling, validating and generating platform specific image file for a CRM / ERP application. The idea is to provide a default template for the application which can be further refined to add or remove features according to the vendors or developers choice.

Whenever the user creates a new project, a package is created which contains a template feature model with basic functionalities for the selected application type (CRM/ERP), source files corresponding to the default feature model, a Model.xml file for representing the features supported a Model.conf file which shows the feature hierarchy and its dependencies. If the user wishes to modify the default features, user has to create new feature in the model explorer window. Once the feature model is updated, the Model Validator will evaluate the feature dependencies; verify the constraints and impose rules on it.

Considering a scenario from ERP; there are two teams working on different modules of a project. The teams are led by two different project leads but they work under the same project manager. In this case we would have three levels of users 1) Project Manager 2) Project Lead 3) Developer. Each of them will have different set of priorities and specialized set of functionalities that they can avail. User at level 1 can avail features available to level 3 users, but the reverse is not supported. The project management features like project planning, resource planning, costing, billing and activity management cannot be defined at level 3 with features such as project deliverables, project schedule, work breakdown structure and payroll. The level 3 user (the developers) should not have access to project management functionalities. If the ERP developers choose to add the above mention functionalities at the same level without defining proper dependencies and access priorities, the Model Validator will display an error message with the error description. It will also provide suggestions to correct and refine the feature model.

Considering the sales module of the CRM software are focused on helping the Sales team to execute and manage the pre-sales process in an organized manner.

All the relevant data of a prospective sales deal like customer, products interested in, expected budget, expected closing date etc are available in one place. Opportunity Management of the CRM system helps the sales representative to carry out activities such as tracking opportunities, including percent chance of close of deal, RFP received, quotation sent and finally whether the deal is won or lost. When opportunity management reaches a Quotation phase, a quotation is generated which if won gets converted into a sales order. The sales order is then passed on to the back end system for further processes. So constraints like sales order is generated only once the quotation is won also verified by our tool.

After verifying the feature model; the Model validator will check for the added features and its dependencies. If the designed feature model fails the validation the Model validator will produce an error message. It will also show suggestions to correct the design. If the feature model passes the validation, the designed model is ready for execution to generate the source files corresponding to every feature. After model validation when the user runs the project, .jak files are generated for all the features in the feature diagram. DSkyL uses AHEAD composer for converting .jak files to corresponding .java files. When user clicks on the run button, DSkyL calls a batch file which executes commands to convert the .jak files to .java files. The generation of .jak files for all the selected features and calling the batch file for running jak2java command occurs simultaneously once the user clicks on the run button. And finally the source code is generated for every feature.

Once the model verification and source code generation is done, the user then defines the SLAs for the cloud provider he has decided upon. The SLA will consist of VM properties (size of RAM, No of processors, CPU processing speed etc), VM allocation rules and cloudlets properties. SLAs are then converted to XML configuration files. The final image file created by DSkyL will be a combination of the model and this configuration file. This is specific for every cloud provider.

Each PaaS provider offers a special flavor in its design. Portability is possible only between a small numbers of PaaSs. But the number of platforms available is tremendously increasing. DSkyL framework offers a solution for the development of interoperable, portable and cloud-provider independent cloud applications. It enables the developers to build cloud applications in a very flexible way, being completely independent from the cloud providers that offer the resources. A method is provided for managing the SLA using which managing resources like virtual machines and storage required by the application and quality of the services offered are agreed upon. In DSkyL the functionalities supported by the application and cloud resources are modeled in terms of the functionalities offered, and not according to how they work.

DSkyL uses the concept of wrappers to allow users to access multiple cloud resources. Each cloud service provider has his own architecture for storage or communication and hence wrappers are created so that it becomes portable across multiple platforms. When user configure required services; a contract will grant users requirements and the Resource Manager will assign physical resources on the basis of the contract.

The major benefit of DSkyL is that the CRM/ERP vendors do not have to worry about the vendor lock in problem in cloud hosting that is expected when they use a specific cloud providers development platform. If the application vendor plans to update the features or change the cloud service provider, they can edit the feature model and run it to the get DSL and source code, add SLAs corresponding to the new cloud service provider and generate the deployable file. This way the application vendors do not have to re-engineer the entire application. Moreover there is not much need for very good technical expertise for creating the application and hosting it in cloud since DSkyL does most of the works in clicks. All the existing process requires intensive involvement of experts both in the development phase and in the migration phase and includes a lot of manual work as well. DSkyL breaks down the high level architecture components to business features. These services offer a clear functionality. It also allows reconfiguring features to compose them in new ways which does not need much technical expertise nor manual tasks.

6 Conclusion

The lack of standard approaches for portability between cloud providers causes the vendor lock-in problem. This problem is hindering the cloud adoption, because users may opt for different cloud providers because a certain reasons like optimal choice on expenses and resources, contract termination or some legal issues. In order to solve this problem, this chapter presents a model-driven approach for cloud portability. The cloud technologies and MDE, together, can benefit the users by providing better productivity, improved maintenance and reuse. The prototype for the proposed approach is still under development. Initially, the focus will be only with few open source cloud providers. As a future work, DSkyL will be refined to support migration of existing CRM/ ERP applications to multiple clouds both open source and commercial.

References

1. Brandtzæg, E.: CloudML, A DSL for model based realization of applications in cloud. Oslo (2012)
2. Rashmi, M.S., Sahoo, G.: A five phased approach for cloud migration. International Journal of Emerging Technology and Advanced Engineering 2(4), 286–291 (2012)
3. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications 34(1), 1–11 (2010)
4. Seccombe, A., Hutton, A., Meisel, A., Windel, A., Mohammed, A., Licciardi, A.: Security guidance for critical areas of focus in cloud computing. Cloud Security Alliance 3, 1–162 (2011)
5. Seccombe, A., Hutton, A., Meisel, A., Windel, A., Mohammed, A., Licciardi, A.: Security guidance for critical areas of focus in cloud computing. Cloud Security Alliance 2(1), 2–70 (2009)
6. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., Ghalsasi, A.: Cloud computing – The business perspective. Decision Support Systems 51(1), 176–189 (2011)

7. Andrikopoulos, V., Binz, T., Leymann, F., Strauch, S.: How to adapt applications for the cloud environment. Computing 95(6), 493–535 (2013)
8. Lewis, G., Morris, E., Smith, D., Simanta, S.: SMART: Analyzing the reuse potential of legacy components in a service oriented architecture environment. Technical Note, Carnegie Mellon University. pp. 1–35 (2008)
9. Alonso, J., Orue-Echevarria, L., Escalante, M., Gorronogoitia, J., Presenza, D.: Cloud modernization assessment framework: Analyzing the impact of a potential migration to cloud. In: Proceedings of IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), pp. 64–73. IEEE Xplore (2013)
10. Cetin, S., Ilker Altintas, N., Oguztuzun, H., Dogru, A.H., Tufekci, O., Suloglu, S.: Legacy migration to service oriented computing with mashups. In: Proceedings of 2nd International Conference on Software Engineering Advances, Cap Esterel, France (2007)
11. Fuhr, A.: Model-driven software migration into a service oriented architecture. Computer Science - Research and Development 28(1), 65–84 (2009)
12. Khadka, R., Saeidi, A., Jansen, S., Hage, J.: A structured legacy to SOA migration process and its evaluation in practice. In: Proceedings of IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), pp. 1–11. IEEE Xplore (2013)
13. Laszewski, T., Prakash, N.: Migrating to the Cloud - oracle client/server modernization. Elsevier, Inc. (2011)
14. Beserra, P.V., Camara, A., Ximenes, R., Albuquerque, A.B., Mendonca, N.C.: Cloudstep: A step by step decision process to support legacy application migration to the cloud. In: Proceedings of IEEE 6th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), pp. 7–16. IEEE Xplore (2012)
15. Gonidis, F., Paraskakis, I., Simons, A., Kourtesis, D.: Cloud application portability: An initial view. In: Proceedings of 6th Balkan Conference in Informatics, pp. 275–282. ACM (2013)
16. Aparna, V., Dash, P., Neelanarayanan, V.: Migration of enterprise software application to multiple clouds: A feature based approach. Lecture Notes on Software Engineering 3(2), 101–106 (2015)

Cloud Based Big Data Analytics: WAN Optimization Techniques and Solutions

M. Baby Nirmala

Abstract. More advanced applications to run the business and ensure competitiveness includes many factors. Few factors that improve the competitiveness includes more distributed branch offices and users; more reliance on web and wide area network; more remote users insisting on high speed networks; unpredictable response times etc. In addition, escalating malware and malicious content have created lot of pressure on business expansion. Also, ever increasing data volumes, data replication at off-site, and greater than ever use of content-rich applications are mandating IT organizations to optimize their network resources. Trends such as virtualization and cloud computing further emphasize this requirement in the current era of big data. To assist this process, companies are increasingly relying on a new generation of, wide area network (WAN), optimization techniques, ap-pliances, controllers, and platforms. Hence, it displaces standalone physical appli-ances by offering more scalability, flexibility, and manageability. This is achieved by additional inclusion of software to handle big data and bring valuable insights through big data analytics. In addition, network reliability, accessibility, and avail-ability can be increased by an optimized WAN environment. Also, the perform-ance and consistency of data backup, replication, and recovery processes can be progressed. This chapter deals with the study of WAN optimization, tools, techniques, controllers, appliances and the solutions that are available for cloud based big data analytics. In addition, it provides a light on the future trends and the research potentials in this area.

1 Introduction

World is becoming more dynamic and dispersed than ever before. The next generation of information technology is being defined by the increased business relevance of the network. This leads to rapidly growing data and the need to replicate it,

M. Baby Nirmala
Holy Cross College, Tiruchirappali, Tamilnadu, India
e-mail: babynirmala7@yahoo.co.in

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_11

the growing adoption of virtualization in the datacenter, a proliferation of mobile devices, an explosion of video traffic, cloud based application virtualization, and growing service delivery requirements. Virtualization and cloud computing (private or public cloud services) are major trends that are driving more and more traffic over wide area network (WAN) and it creates a lot of research interests in recent years. Therefore, for virtualization, cloud, and software defined network (SDN), there is a need of optimal WAN performance [1]. The following Figure 1 lays profound foundation for convergence of technologies and large scale data generation.

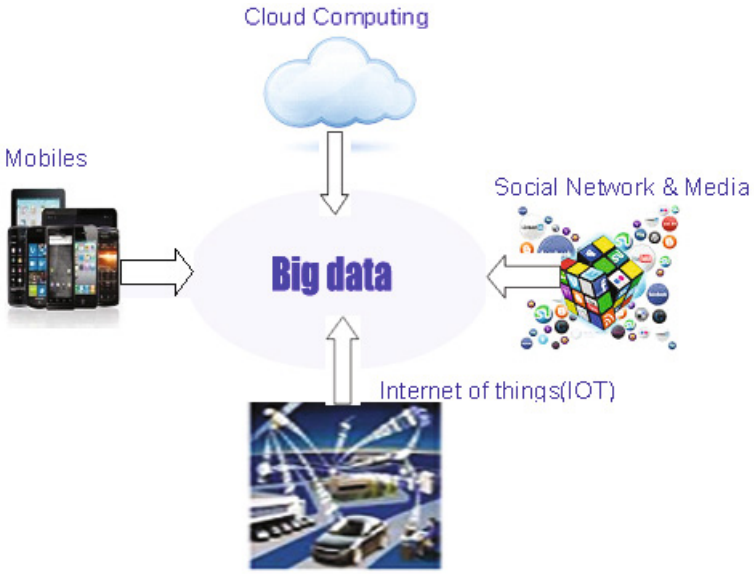


Fig. 1 Convergence of technologies

As per IDC (write Full form) report, 2.5 quintillion bytes of data are created every day and 90% of data in the world today has been created in the last two years. This voluminous data consists of structured, unstructured, machine generated, sensor generated, and mobile data which is otherwise called as big data. These data are very much known for its key characteristics of 5V's such as volume, variety, velocity, value, and veracity. This shows the emergence of the era of big data, third era of technology, where computers learn and process. Companies and organizations are very much interested in getting valuable insights and patterns from these big data. All these leads to a new branch of science called 'big data analytics'. In this current era of big data, cognitive science systems are built to analyze information and draw insights from it [1, 21]. As a result, enterprises, developers, and cloud providers are coming up with robust strategies to resolve the issues and ensure customer satisfaction by optimizing WAN environment and to handle all problems

raised by these big data platforms. In 2008, the WAN optimization market was estimated to be $1 billion and it will grow to $4.4 billion by 2014 according to Gartner, a technology research firm [9]. So, introducing innovative tools, techniques, technologies, products, solutions, appliances, devices for WAN optimization for big data platforms in this cloud environment is essential.

The objectives of this chapter is to give a brief introduction of WAN optimization and its necessity in the paradigm of cloud computing, followed by various techniques and technologies associated with it. Furthermore, it discusses the limitations and barriers faced by WAN due to the flood of data. In addition, it discusses various solutions, techniques, technologies, tools, products, controllers and appliances to handle the above said problems. It also discusses the current and research issues.

2 WAN Optimization

WAN Optimization is a fast growing field. It is defined as a collection of techniques used for increasing or improving data transfer efficiencies across wide area network [1]. According to Nancy Conner (2009), WAN optimization is defined to maximize the business applications over distributed network having the visibility to classify, prioritize applications, and the ability to accelerate one's organization. Also, the layered defenses must protect users and information [19].

The rapid proliferation of branch offices, outsourcing, telecommuting and ever more mobile workforce indicates that the end users may be anywhere in the world. For this, applications are becoming more diverse, centralized or even outsourced. WAN optimization speeds up performance and gives control over distributed network. Business benefits of optimizing WAN anyway are many but increased productivity, decreased cost and regulatory compliance are important to notify [19].

The most common measures of transfer control protocol (TCP), data transfer efficiencies (i.e., optimization), are bandwidth requirements, latency, throughput, protocol optimization, and congestion as manifested in dropped packets. In addition, the WAN itself can be classified with regards to the distance between endpoints and the amounts of data transferred [18]. In general, WAN optimization encompasses the following:

- Ttraffic Shaping: in which traffic is prioritized and bandwidth is allotted accordingly.
- Data De-duplication: which reduces the data that must be sent across a WAN for remote backups, replication, and disaster recovery.
- Compression: shrinks the size of data to limit bandwidth use.
- Data Caching: in which frequently used data is hosted locally or on a local server for faster access. It also monitor the network by detecting non essential traffic, creating and enforcing rules about downloads and Internet use.
- Protocol Spoofing: in which chatty protocols are bundled so that they are, in effect behave as a single protocol.

WAN optimization plays a prominent role in improving the network performance. As, the network meets a lot of traffic, the performance automatically reduces. In this era where large scale of data is generated at greater speed, convergence of technologies such as cloud computing, internet of things (IoT), social networks and mobile phones generate bulk of data across the network. Thus, it suffers greatly because of the great velocity of data at which it arrives. Organizations need the ability to optimize and secure the flow of information otherwise this translates into poor network performance. For business organizations, optimized performance of these networks is considered as a top priority to effectively execute business functions over the widee area network or Internet [21].

2.1 Issues and Challenges

The next generation of information technology (IT) is being described by the growing business relevance in the wide area network. It faces a lot of issues and challenges which includes the followings:

- Rapidly growing data volumes and the need to replicate the big data
- Centralization and response time
- Growing adoption of virtualization in the data center
- Proliferation of mobile devices
- Explosion of video traffic
- Cloud-based application virtualization
- Growing service delivery requirements
- Remote users
- Chatty protocols and latency

Vendors offer WAN optimization solutions to meet these challenges. Therefore, WAN optimization market is growing at a healthy rate and continues to be a very competitive space. Several vendors stake a claim for leadership and mindshare. Though there is significant and growing interest to evaluate virtualized WAN optimization solutions, the actual uptake for these deployments in the enterprise and managed services segments is still in its early stages and hence relatively small. Vendors from the tangential networking market segments can be expected to try to expand their portfolio to address the necessities of WAN optimization market as convergence in the datacenter gains greater acceptance [19].

3 WAN Optimization Techniques

WAN optimization approaches can speed up, application performance, facilitate high availability, and significantly enhance throughput levels. Following are some of the key WAN optimization techniques, which facilitate enterprises and organizations to establish and maintain a sound network to efficiently continue their business processes [1].

1. De-duplication: De-duplication (intelligent compression) may be defined as the process of eliminating redundant data so as to reduce storage needs. It is also referred to as single instance storage or capacity optimization.
2. Compression: Compression is the technique of reducing data size in order to save space or the time of transmission.
3. Latency optimization network performance: Bandwidth and latency are the major factors that determine network performance. Latency optimization should be inculcated in any plan to improve network performance
4. WAN optimization caching / Proxy: The proxy server is caching a response from a server and distributing it directly when it receives an identical subsequent request. A proxy server that passes responses and requests without modification are called gateway or tunneling proxy servers.
5. Forward error correction: Forward error correction is a method of controlling error in the transmission of data by sending redundant data to the receiver or destination on recognizing portions of the data that are effort free.
6. Protocol spoofing: Protocol spoofing is an essential part of data communications that helps to enhance performance.
7. Traffic shaping: Traffic shaping is a form of rate limiting and is used for managing traffic on a network so that the output fits a desired traffic profile.
8. Equalizing / Load balancing servers: Managing large quantities of data, requests for information via the web is a continuing challenge all IT managers face on a day to day basis. This challenge will only grow as connectivity spreads to all parts of the globe.
9. Simple rate limiting: Rate limiting is any technique or process that is used to control the rate at which network traffic is sent.

3.1 WAN Optimization for Video Surveillance

Three simple ways to optimize the bandwidth management in video surveillance. Although an organization's IT infrastructure is typically built to handle large amounts of data, including that of video surveillance applications, many operators are not fully leveraging the functionalities of their investments to optimize bandwidth capacity. Multicasting, multistreaming and video compression are the three innovative methods that users can use to optimize bandwidth management in video surveillance applications. Purpose based multistreaming, multistreaming for remote user access, and multistreaming for maximized tile viewing are to get more with multistreaming capabilities. With technologically advanced video management software (VMS) users can leverage existing hardware and software functionalities to experience benefits such as reduction in bandwidth requirements, optimization of network resources, and decrease of storage needs. Multicasting, multistreaming, and video compression as supported by a highly intelligent VMS ultimately contribute to significant cost savings and long term investment protection [4].

4 Tools to Improve Application Performance

As big data flow in great velocity, IT organizations are in search for ways and means to deal with delivery and performance of key applications on the network. Also, it looks for reliability, security, and cost efficiency. To handle these challenges, companies are progressively turning to application performance management solutions known as tools and techniques. They are used for managing and monitoring applications across the wide area network. A broad overview of layers of application delivery functionality is depicted in the following Figure 2.

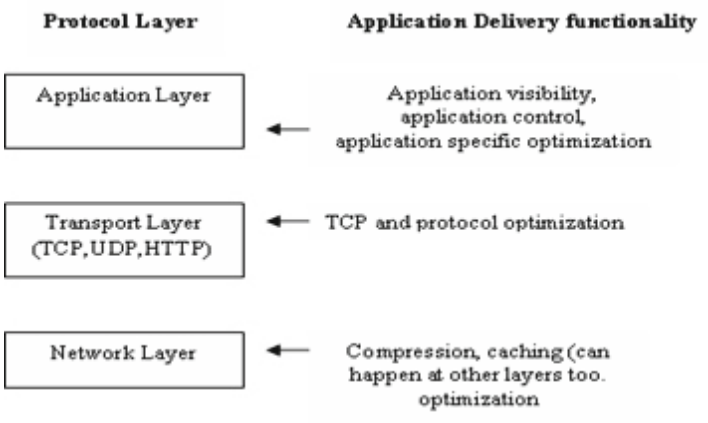


Fig. 2 The layers of application delivery functionality

At application layer, application visibility, application control, application specific optimization is taken care off. TCP and protocol optimization is taken care at transport layer and at network layer, compression and caching optimization is done. These tools may offer comprehensive, visual representations of accomplishments on the network. They offer the analysis needed to take action and make informed decisions too. Application performance management (APM) can help observe and manage the performance of the wide area network, and the business critical applications that run over it, including voice over internet protocol (VoIP).

4.1 Blue Coat Application Delivery Network

As the world is becoming increasingly global and collaborative, Blue Coat has developed an infrastructure design to support and enhance the ever changing WAN environment. Blue Coat offers visibility, acceleration, and security required to optimize and secure the flow of information to any user, on any network, anywhere. The network infrasture combines both application network infrastructure and three core technologies which are application performance monitoring, WAN optimization and secure web gateway [17, 19].

5 WAN Optimization Appliances

WAN optimization appliances, related technology devices, WAN optimization controllers, and other such optimization products support the task of enhancing network performance around the well spread WAN, Internet and the ever increasing flood of data. Techniques used by these appliances and devices are compression, protocol optimization, caching, boosting efficiency of transmission control protocol (TCP), and enforcing quality of service (QoS) methods.

WAN Optimization appliances and devices advances the transaction time between sites by 90% or even more. In addition, it boost up the usability experience for users, employs various techniques to reduce WAN traffic across WAN circuits, compress internet protocol (IP) traffic that is transmitted over the network, and eliminate congestion in the path. Quicker response time for applications, database /resources access, and high throughput are the result of the presence of WAN optimizers at both ends of WAN links which fine tunes the IP traffic.

6 WAN Optimization Controllers

WAN optimization controllers (WOCs) are committed devices that are normally used for enhancing the application response time. They are deployed on a WAN link on either end. A WAN optimization controller is typically a customer premises equipment (CPE) and is connected to WAN routers on the local area network (LAN) side. By purposefully deploying WOCs at data centers and remote locations, application performance over WANs can be substantially improved. WOCs are focused on improving the response time of business sensitive applications that are conducted across WAN links [1, 10].

6.1 Complementing WAN Optimization Controller Investment for Big Data and Bulk Data Transfer

WAN optimization controllers (WOCs) cannot accelerate big data and bulk data transfers typically needed for remote backup, disaster recovery, data migration, demanding business intelligence applications by their nature. In order to accelerate, data intensive transfers dedicated file transfer acceleration (FTA) solutions can be deployed and thereby complement your WAN optimization strategy and leverage the overall investment. Figure 3 shows how dedicated FTA solutions accelerate when large data sets are transferred when compared to WOC's [11].

A WAN optimization strategy that also addresses big data and bulk data transfers should comprise a mixture of standard WAN optimization controllers, and dedicated file transfer acceleration solutions. New data greedy applications require dedicated FTAs. There are only few vendors that provide FTA solutions, each with a different focus. One can choose appropriate FTA vendor by framing few questions in mind.

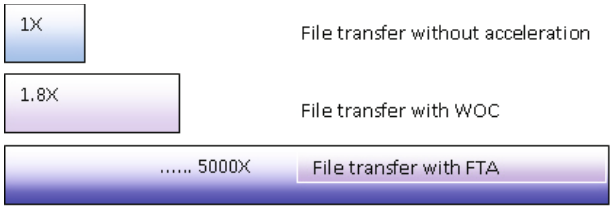


Fig. 3 File transfer with and without FTA

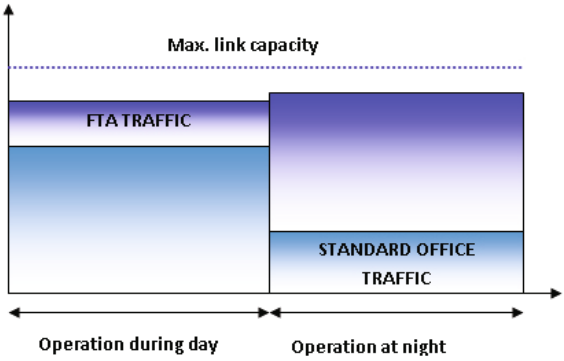


Fig. 4 Deployment of file transfer acceleration

Besides saving a lot of time, when transferring bulk data with dedicated FTA solutions, they form a typical network utilization standard office traffic during day and IT backend traffic at night. Figure 4, shows this outcome. It is crucial to identify the hot spots in this ecosystem and its corresponding processes and deploy carefully chosen FTA solutions accordingly to leverage the impact of the overall investment.

6.2 WAN Optimization Controller Comparison: Evaluating Vendors and Products

Information technology organizations have a growing interest in deploying WAN optimization controllers as a way to implement network and application optimization on branch office networks. One of the factors driving this interest in WOCs is vendors and their products. It is bacause, many organizations have taken applications out of branch offices and consolidated them in centralized data centers. This is combined with the fact that only a minority of employees now work at a headquarters site, means that the vast majority of employees now access applications over a relatively low speed, high latency WAN. In addition to low speed and high latency, the WAN also suffers from quality issues such as dropped packets or packets that are out of order [1].

The amount of application traffic that transits the WAN increases because, companies increase in their use of software as a service and other forms of public and private cloud Computing. This will lead to more interest in deploying WAN optimization and acceleration. WOCs benefit businesses by improving the performance of applications that run over the WAN and / or reducing WAN bandwidth expense. These products improve performance by:

- Reducing the amount of data sent over the WAN by implementing functionality such as caching de-duplication and advanced compression.
- Mitigating the impact of inefficient network protocols such as TCP.
- Protecting traffic that delay sensitive and business critical from being affected by other traffic types through the implementation of mechanisms like QoS classification and traffic shaping.
- Implementing application specific optimization to improve the performance of higher level protocols and applications such as the common internet file system (CIFS) protocol, HTTP, messaging application program interface (MAPI) and share point

One way to segment the WAN optimization market is to analyze a WOC's ability to provide optimization between disparate data centers, branch offices data center and remote users, and one or more data centers [18].

7 WAN Optimization for Big Data Analytics

Advancements in WAN optimization technology are constantly happening, but there is fierce competition in the market for producing WAN optimizers that deliver tangible benefits in terms of performance, scalability, and integrity. Simultaneously, customers look for optimizations, security, scalability, and mobility in WAN optimization devices and related appliances. These demands from customers form the basis for further developments [11, 2]. In addition, a typical WAN optimization solutions must have the following features:

- Bandwidth optimization including compression of data streams using deduplication or caching
- congestion management including traffic shaping and prioritization
- Loss mitigation to fix dropped or out-of-order packets
- Latency mitigation via TCP acceleration
- Ability to monitor application and network performance for bandwidth, latency, and loss

WAN optimization tools are becoming more flexible, agile, and virtualization friendly to accommodate all of these key trends.

7.1 Key Trends in WAN Optimization for Big Data Analytics

In 2008, the WAN optimization market was estimated to be $1 billion and it will grow to $4.4 billion by 2014 according to Gartner, a technology research firm [1]. IDC projects tells that the WAN application delivery market will continue to grow at a compound annual growth rate (CAGR) of 7.4% over the five year forecast period from 2011 to 2016 and will be reaching $1.8 billion by 2016. The IT market has shifted to a new phase with IT managers continuing to invest in this technology. The average IT manager is juggling a multitude of challenges, including the followings [11]:

- Real time data transmission between multiple data centers
- Regular backup of data to and from distributed sites for disaster recovery
- Reduce in infrastructure costs and avoiding hardware refresh cycles
- Inclusion of new big data projects to the list of mission critical systems
- Managing skyrocketing internet usage
- Delivering content to increasingly distributed sites including branches and remote users
- High bandwidth requirements for desktop video and video conferencing
- Must support VoIP and other unified communications (UC) applications that are latency sensitive and require appropriate quality of service
- In time and efficient security, and application updates for distributed desktops
- Meeting business needs for data analytics and real time data in support of business intelligence

7.2 Drivers of WAN Optimization for Big Data

Data is a key accelerator of network performance. By leveraging the Apache Hadoop framework, big data software is typically deployed. The enterprise network is the foundation for transactions between massively parallel servers, server clusters, and existing enterprise storage systems in big data topologies. The enormous amount of data to be transported would increase real time usage. As an outcome, the enterprise network connecting information sources and processing must be strong enough to make sure the data can move quickly and efficiently [1,11].

8 WAN Optimization Solutions

In the previous section, various aspects of WAN optimization is discussed. This section clearly discuss various issues related to WAN optimization solutions [2].

8.1 Infineta Sytems and Qfabric

Infineta systems accelerate big traffic over data center. QFabric WAN optimization solution accelerate replication, high speed backup, storage virtualization, and large

file transfer. They also enable applications such as replication, limited WAN bandwidth, backup to scale LAN speeds in the face of growing storage requirements, running between data centers.

Juniper Networks QFabric system and the Infineta Systems data mobility switch (DMS) optimize data center interconnects to deliver the highest performance for critical applications such as high speed replication, data backup, and live migrations. Juniper Networks QFabric technology and Infineta Systems data mobility switch high speed WAN optimization solution is depicted in Figure 5. The benefits include the followings:

1. Full WAN bandwidth utilization at intercontinental distances between any WAN type
2. Application performance assurance for critical applications
3. Processing latencies averaging 50 μs permit optimization, for most latency sensitive applications
4. Sustained 80 to 90% reduction (What) at 10 Gbps

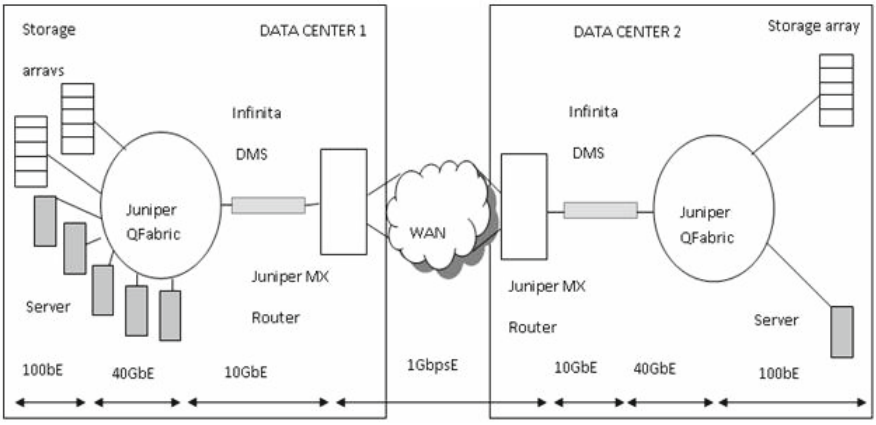


Fig. 5 Juniper Networks QFabric technology and Infineta Systems data mobility switch high speed WAN optimization solution

8.2 BIG-IP WAN Optimization Manager

BIG-IP WAN optimization manager (BIG-IP WOM) is used to provide utmost guarantee on application performance, data replication, and disaster recovery requirements. It prevails over network and application issues on the WAN. In addition, BIG-IP WOM can spectacularly decrease data replication times and facilitate more efficient use of existing bandwidth. These services are accessible and available as an add on module on the BIG-IP local traffic manager device or as a standalone appliance or virtual edition [1]. It also helps in data replication and backup, storage

requirements, data center consolidation drives, and transfer of increasing amount of data between data centers. Though virtualization and cloud computing have many benefits, they also add latency to application delivery [8].

Symmetric Data De-duplication: BIG-IP WOM delivers a highly advanced level of WAN optimization, with symmetric data de-duplication. This expands WAN capacity and provides significantly more bandwidth to improve response times and increase throughput. By the usage of pattern matching and byte catching technologies, redundant data is transferred quickly. Symmetric data de-duplication reduces the amount of data transferred over the WAN by up to 99 percent and ensures high speed application performance. Figure 6 shows the symmetric data de-duplication through BIG-IP WAN optimization manager.

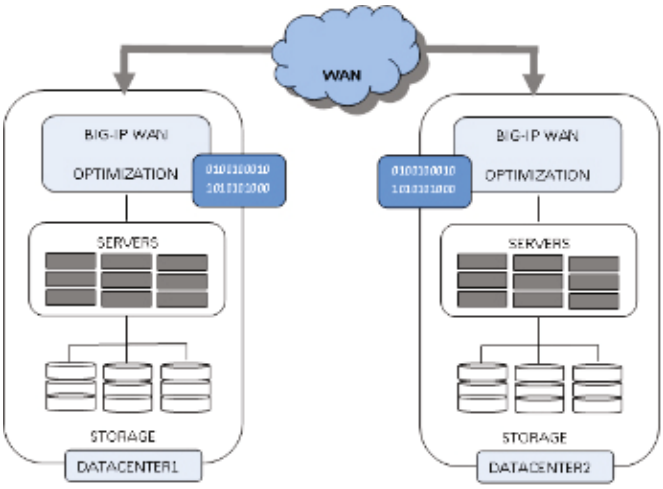


Fig. 6 Symmetric data de-duplication through BIG-IP WOM

8.3 Edge Virtual Server Infrastructure

Riverbed extends from WAN optimization to edge virtual server infrastructure. Riverbed has created a multi-tiered, cross-domain, and integrated solution that links service consolidation based on high performance virtualization with storage consolidation based on first-in-class data streaming technology. It ties them together with WAN optimization to deliver rock solid performance and cost savings for highly distributed customer environments [5]. The complete consolidation of steelhead edge virtual infrastructure is depicted in Figure 7.

8.4 EMC Isilon and Silver Peak WAN Optimization

EMC ISILON replicates big data 90 times faster over the WAN without increasing WAN costs. It enables to aggregate big data onto single, shared, easy to use network

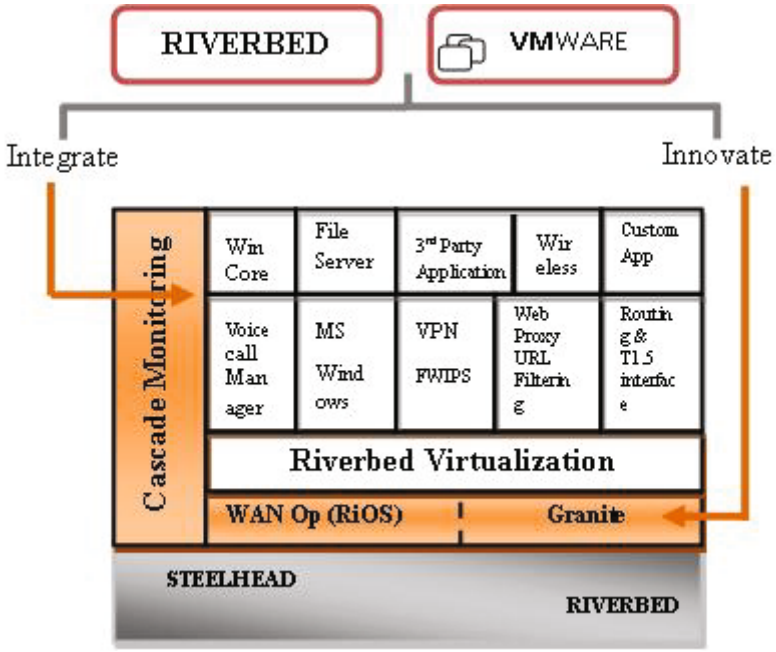


Fig. 7 Complete consolidation with the new steelhead

attached storage (NAS) platform. Silver Peak is an industry leader in data center WAN optimization has partnered with EMC Isilon to overcome the challenges of this current era of data explosion with a great WAN traffic. WAN optimization reduces the amount of traffic sent across the WAN and delivers information locally whenever possible. Silver Peak ensures fast and reliable data replication and seamless access to centralized data, regardless the location of EMC Isilon clusters by conditioning the WAN for optimal data throughput. As a result, enterprise organizations can get the best return on investment by deploying devices in any location, on any type of network.

When Silver Peak WAN optimization and EMC Isilon are deployed together, enterprises can replicate more data over longer distances without investing in costly WAN upgrades. This has a major impact on enterprise disaster recovery initiatives. In addition, remote users can access centralized NAS resources located anywhere in the world. This mazimizes employee collaboration while minimizing ongoing IT costs [1].

Silver Peak's technology and solutions is known for making a high performance, high capacity WAN optimizer with a focus on improving backup, replication, and recovery between data centers, as well as facilitating branch office server and storage centralization by improving WAN performance. Most notably, its software based architecture has allowed it to easily embrace virtualization with comprehensive hyper visor support and scalability. Further, its ability to accelerate all IP traffic allows

network managers to manage the various workloads on their enterprise networks [20].

8.5 F5 WAN Optimization Module (WOM)

Transfer control protocol (TCP) optimizes byte caching compression, and bandwidth. Use of TCP optimization helps in maximizing throughput BIG-IP WAN optimization module (WOM) and allows two BIG-IP devices to communicate. These are used to communicate across the WAN to optimize traffic during data replication and off-site backup.

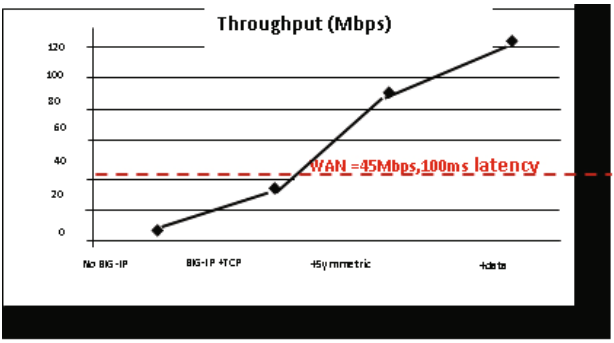


Fig. 8 Throughput of BIG-IP WAN optimization module

8.6 BIG-IP WAN Optimization Module

BIG-IP WAN optimization module (WOM) enables customers to accelerate data replication or data backup across the WAN. Therefore, customers mitigate the effects of latency, optimize existing bandwidth to replicate or backup the same data. Hence, it control the cost and eliminate the need for costly band with upgrades that can guarantee bandwidth and prioritize backup or replication traffic. In addition, it can meet standards for data backups and recovery times. The following Figure 9 depicts the BIG-IP WAN optimization module.

8.7 F5 WAN Optimization for Oracle Database Replication Services Faster Replication across the WAN (Can Title be Short)

The Oracle and F5 have partnered together to produce a solution to ensure the protection of data with Oracle 11g database replication services like data guard, golden gate, recovery manager and Streams. This helps in replication of critical data across WAN between data centers in less time [6, 7]. The following Figure 10 depicts the F5 WAN optimization for Oracle database replication services.

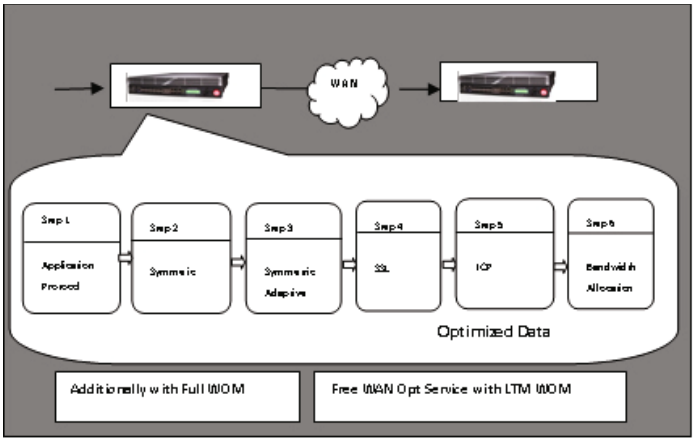


Fig. 9 BIG-IP WAN optimization module

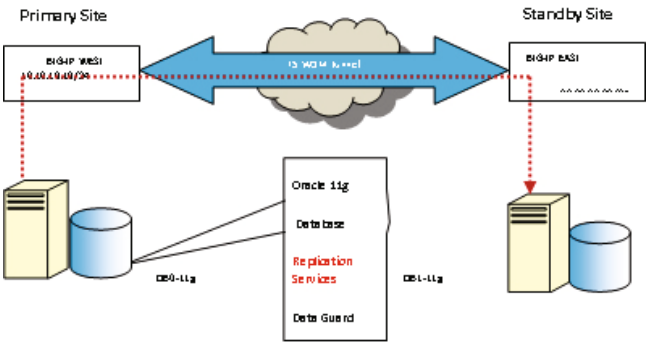


Fig. 10 Oracle Database Replication w/F5 WAN Optimization Module

9 Future Trends and Research Potentials

The next generation of IT is being defined by the increased business relevance of the network. This includes rapidly growing data volumes and the need to replicate the data, growing adoption of virtualization in the datacenter, a proliferation of mobile devices, an explosion of video traffic, cloud based application virtualization, and growing service delivery requirements. Virtualization and cloud computing (private or public) services are major trends that are driving more traffic over wide area network (WAN) and create a lot of research interests. Also, software defined network (SDN) demand optimal WAN performance is of great challenge today [11].

9.1 WAN Optimization in Virtual Data Environments and Cloud Services

WAN optimization in virtual data environments and cloud services is of great challenges today. Though many organizations take these into considerations in their products in near future, organizations will be able to spin up entire data environments on fully virtualized infrastructure. This will be most likely distributed across great distances because of the breakdown of the data center edge and, in a larger sense the LAN and the WAN. So research on WAN optimization for virtualization is a welcoming trend in this current era.

On the other end WAN optimization technology is a fundamental component for the successful delivery of cloud services. Asynchronous and virtual WAN optimization appliances are needed for data migration. Compressing and de-duplicating data before it hits the WAN will shorten the length of time required to move. As a result, that first terabyte of big data can be transferred in days instead of weeks.

9.2 Limitations of WAN Optimization Products

This section discusses various limitations of WAN optimization products. It is essential to overcome these limitations in future research and product development. These limitations are listed below.

1. WAN products available today suffers from scalability. Adoption of better big data structures like B-tree which can scale to much larger sizes while efficiently using the memory hierarchy can be experimented [1, 12].
2. Evaluating data de-duplication techniques is of great challenge as widely playing hash algorithms can meet hash collisions. Applying dynamic chunking algorithms can improve the performance to an extent [13]. Content aware bit reducing algorithms has its own demerits during hyper factor algorithm of IBM and it too faces troubles if the compression occurs before de-duplication [14, 15].
3. Another important limitation is of bandwidth reduction. Better compression algorithms to be implemented like those which reduces bandwidth so as to reduce traffic volume. Header and payload compression techniques utilize pattern matching algorithms to identify short, frequently recurring byte patterns. Any gain compression strategies must vary according to the mix and makeup in WAN traffic. Applying compression across various flows of traffic can still enhance effective bandwidth [16].
4. Faster transport or communication algorithm can be implemented by making use of protocol optimization at various levels of catching, to minimize size and frequency of networked communications [16].
5. Data security over WAN optimization is also another important aspect. Therefore, data protection mechanisms while on the transit, usage and persistence is to be improved.

9.3 Accelerating Data Migration with WAN Optimization

Challenges faced by many enterprises with data center moves and consolidation, not at all different from challenges associated with migrating data to the cloud. There are many ways to tackle these challenges. Trade-offs will need to be made to minimize the cost and risk and to increase the speed. Moving data across the network can take some risk out of the equation and has a much better success rate, but it is time consuming and expensive. Thus, the technology WAN optimization is essential to give it a boost. It is a proven technology and has been used broadly for data center consolidation, remote office application acceleration, and disaster recovery initiatives. WAN optimization technology can help reduce the pain of migrations and so they are completed faster and at low cost. Using WAN optimization, users can finish migrations faster and return to productivity sooner [3].

10 Conclusion

In today's IT environments, with convergence of technologies and due to the generation of great velocity of big data, network managers are facing and coping with a huge number of challenges for the commercial WAN. This includes the necessity to afford real time big data transmission between multiple data centers, delivering content to increasingly distributed sites, and backup of network-intensive applications. To tackle all these challenges, adopting proper WAN optimization tools, technologies, techniques and solutions are dealt in this chapter and for optimizing the entire network. This is an ever growing field, every day new technologies, tools and solutions are introduced by various vendors. Therefore, choosing the proper solution also needs proper skills and decision making capacity of the network managers. In addition, there are many research issues which can be taken by the evolving new research works.

References

1. Nirmala, M.B.: WAN optimization tools, tecniques and research issues for cloud based big data analytics, pp. 280–285. IEEE Xplore (2014), doi:10.1109/WCCCT.2014.72
2. Nirmala, M.B.: A survey of big data analytic systems: Appliances, platforms and frameworks. In: Pethuru Raj, C., Ganesh, C.D. (eds.) Handbook of Research for Cloud Infrastructures to Big Data Analytics, pp. 393–419. IGI Global, USA (2014)
3. McClure, T.: Accelerating data migration with WAN optimization. In: Data Center Consolidation and Construction Trends, pp. 1–3. Enterprise Strategy Group, Inc. (2010)
4. Genetec: Three Simple Ways to Optimize Your Bandwidth Management in Video Surveillance. White paper, pp. 1–13. Genetec, Canada (2010) `http://www.genetec.com/Documents/EN/Whitepapers/EN-Genetec-Three-Simple-Ways-to-Optimize-Your-Bandwidth-Management-in-Video-Surveillance-WhitePaper.pdf`

5. The Taneja Group: Riverbed extends from WAN optimization to edge virtual server infrastructure (Edge-Vsi). Taneja Group, Technology Analysts, pp. 1–8 (2011), http://www.ndm.net/wanoptimization/pdf/Whitepaper-Taneja-Riverbed-Sets-a-New-Standard-for-WAN-Opt.pdf
6. MacVittie, L.: F5 WAN optimization for Oracle database replication services, pp. 1–18. F5 Network, Inc., USA (2012)
7. Akker, C.: F5 WAN optimization for Oracle database replication services faster replication across the WAN. White paper, pp. 1–20. F5 Network, Inc., USA (2011)
8. F5 Network: A data sheet on BIG-IP WAN optimization manager, pp. 1–12. F5 Network, Inc., USA (2013)
9. Gartner: Gartner magic quadrant report on WAN optimization controllers. White paper. Gartner Inc., USA (2013)
10. Aust, A.: Complement your WAN optimization controller investment for big data and bulk data transfer. White paper, pp. 1–4. TIXEL GmbH, Germany (2013)
11. Peak, S.: IDC: The role of virtual WAN optimization in the next generation datacenter. White paper. IPEXPO Online, London (2012)
12. Blender, M.A., Bradley, C.K.: Data structures and algorithms for big databases. State University of NewYork, pp. 1–208 (2012)
13. Machanavajjhala, A.: Algorithms for big data management. Duke University, USA (2013), www.cs.duke.edu/courses/spring13/compsci590.2
14. Moon, Y.C., Jung, H.M., Yoo, C., Ko, Y.W.: Data deduplication using dynamic chunking algorithm. In: Nguyen, N.-T., Hoang, K., Jędrzejowicz, P. (eds.) ICCCI 2012, Part II. LNCS, vol. 7654, pp. 59–68. Springer, Heidelberg (2012)
15. Orlando, K., Bautista, M.M., Mejia, J.R.M., Langnor, R.G.: IBM ProtecTIER Implementation and Best Practices Guide. Redbooks Publications, USA (2014)
16. Ritter, T.: Simplifying branch office management. TechTarget, SearchEnterpriseWAN.com, Newton, MA (2010), http://searchenterprisewan.techtarget.com/ebook/Simplifying-branch-office-management
17. Tittel, E.: Optimized WAN Application Delivery. Realtime Publishers, Inc., USA (2010)
18. Machowinski, M.: WAN optimization appliance market highlights 1Q09. Enterprise Routers and WAN Optimization Appliances, Infonetics Research (2009)
19. Conner, N.W.: WAN Optimization for Dummies, Blue Coat Special, 2nd edn. Wiley Publishing, Inc. (2009)
20. Laliberte, B.: An ROI analysis of virtualized WAN optimization software. Solution Impact Analysis, pp. 1–14. Enterprise Strategy Group, Inc. (2013), http://www.silver-peak.com/sites/default/files/infoctr/esg-case-study-silver-peak-roi-jul-2013.pdf
21. Pethuru Raj, C.: The IT readiness for the digital universe. In: Pethuru Raj, C., Ganesh, C.D. (eds.) Handbook of Research for Cloud Infrastructures to Big Data Analytics, pp. 1–21. IGI Global, USA (2014)

Cloud Based E-Governance Solution: A Case Study

Monika Mital, Ashis K. Pani, and Suma Damodaran

Abstract. The development authorities are facing an exponential increase in the information, and the management of storage and flow of this information is becoming difficult day-by-day, resulting in the definite need of the implementation of information technological tools to maintain the same. Many of the development authorities have implemented or in the process of implementation of IT for complete or partial functioning. Since the development authorities in the state are classified in A/B/C categories depending upon the size and functionality, the priority of the functional requirement also varies to match the budget of IT implementation. To cater the prioritized implementation, the complete application needs to be designed in a modular form consisting of multiple systems, which are individually a complete system in itself but may be integrated with other systems to form efficient information flow. Ghaziabad Development Authority (GDA) had three options: Both the IASP and the citizen portal are on-premise solutions, both the IASP and the citizen portal are on cloud, and the IASP is hosted on premise and the citizen portal is hosted on cloud. The primary aim of the case is to expose students to an infrastructure planning situation in an organization based on the requirements and resource constraints within an organization.

1 Introduction

The ever increasing Internet bandwidth and the fast changing needs of businesses for effectiveness and integration within and with the partners and the distributed or mobile employee force is leading organizations to adopt information systems infrastructures that are cost effective as well as flexible [1]. In a cloud based software as a service (SaaS) business model of software provisioning, the consumer does not manage or control the underlying information systems infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with

Monika Mital · Ashis K. Pani · Suma Damodaran
XLRI, Jamshedpur, India
e-mail: monikaajit@gmail.com,
{akpani,suma}@xlri.ac.in

D.P. Acharjya et al. (eds.), *Computational Intelligence for Big Data Analysis*,
Adaptation, Learning, and Optimization 19, DOI: 10.1007/978-3-319-16598-1_12

the possible exception of limited user-specific application configuration settings [2]. Software as a service refers to the delivery of the software through the internet to multiple customers or tenants (multi-tenant) on a subscription (i.e. fixed or usage based) basis [3, 4]. This able to achieve economies of scale and scope through shared resources infrastructure for the service provider, to be able to achieve service continuity in a cost effective manner, and maintain system reliability, availability, supportability, and manageability [5, 6]. Software as a Service is a subset of the information systems outsourcing domain. Information systems outsourcing started in the 1990's when Kodak outsourced its information system development [7]. SaaS takes advantage of the thin customer technology and provisions SaaS based upon the Internet and semantic technologies, where all the software and the data reside on the server and the customer side needs an interface application like the browser, as against the packaged software provisioning model where the software is sold as a product. Some of the successful examples of SaaS are salesforce.com and NetSuite. Although there are pure SaaS service providers, i.e., only provide SaaS, such as SalesForce and NetSuite, but some traditional packaged service providers such as Oracle, Microsoft, SAP and IBM are fast adopting hybrid SaaS. These provide SaaS as well as packaged software to accommodate customer expectations and preferences [8]. According to the Sand Hill Group and McKinsey & Company report [9], the SME organizations are the biggest adopters of the SaaS model.

E-governance is the delivery of online government services, which provides the opportunity to increase citizen access to government, reduce government bureaucracy, increase citizen participation in democracy and enhance agency responsiveness to citizens needs [10, 11]. E-governance leads to improve efficiency by reducing the time spent upon manual tasks, providing rapid online responses, and improvements in organizational competitiveness within public sector organizations. Implementation of e-Governance is a highly complex process requiring provisioning of hardware, software, networking, in addition to context-specific e-government programs [12]. Most of the public organizations lack funds to be able to develop, implement and deliver e-governance services. Software as a service based e-governance can prove to be a cost effective solution to the problem.

The processes and the needs of the government organizations are very specific to the each organization and so it is not possible for a service provider or the software vendor to be able to generate economies of scale and so be able to manage service provider sustainability. Therefore, customization and cost are the key motivating factors for the customer, standardization to be able to generate economies of scale and sustainability are the key motivations for the service provider. Although cost is an important motivation for choice of SaaS based citizen services by the customer, but successful implementation of a successful SaaS based software solution would require a fit between the cross motives of the customer and the service provider.

2 ACME Development Authorities Management System

The development authorities are autonomous bodies under the department of housing, Government of UP. The development authorities are present in most of the

cities of the state. These development authorities were formed with basic objective to provide planned and controlled growth to the city or township.

With the exponential increase in the information, the management of storage and flow of this information is becoming difficult day-by-day, resulting in the definite need of the implementation of information technological tools to maintain the same. Many of the development authorities have implemented or in the process of implementation of IT for complete or partial functioning. Since the development authorities in the state are classified in A / B / C categories depending upon the size and functionality, the priority of the functional requirement also varies to match the budget of IT implementation. To cater the prioritized implementation, the complete application needs to be designed in a modular form consisting of multiple systems, which are individually a complete system in itself but may be integrated with other systems to form efficient information flow.

ACME development authorities management system (aDAMS) is a pioneer project in the North of India for development authorities. There are 22 development authorities in U.P. out of which 4 are in categoryA and 18 are in CategoryB. A similar project had been sanctioned by the state government in 2004 for the whole of U.P. (integrated project), but the project was declared a failure in 2009 (press releases and the Government order to support the data). The project has two parts-IASP and the Citizen Portal. IASP deals with the core operations and citizen portal is the customer facing portal.

The requirements for the core application are highly customized for development authorities. But once the software is created for a development authority in U.P., it would require around 20% change and customization each to meet the requirements of the other respective development authorities. So we can say that it is 80% similar across authorities.

Many project and schemes have been planned and managed every year, which requires a lot of administrative and staff work. Activities of property planning to disposal need the coordination of several departments for the smooth functioning. Coordinating these activities has its own inherent problems because a lot of paper work is involved. This includes personnel related, access constraints, file storage space etc. Many of these problems can be removed or minimized if the irrelevant paper work and redundancy of data is removed.

ACME DAMS addresses and solves the problems described above by integrating the functioning of all the departments of the development authority and automation of routine departmental work, thus increasing efficiency and speed of workforce and elimination of data redundancy. The aDAMS system caters to three types of needs at Ghaziabad Development Authority (GDA):

1. **Citizen Services:** Easy accessibility of required information, reduce communication gap, transparency in operations, online payment facilities to allottees
2. **Operational Needs:** For efficient management of increased volume of work
3. **MIS Needs:** Monitoring reports and ad-hoc Reports, decision support reports

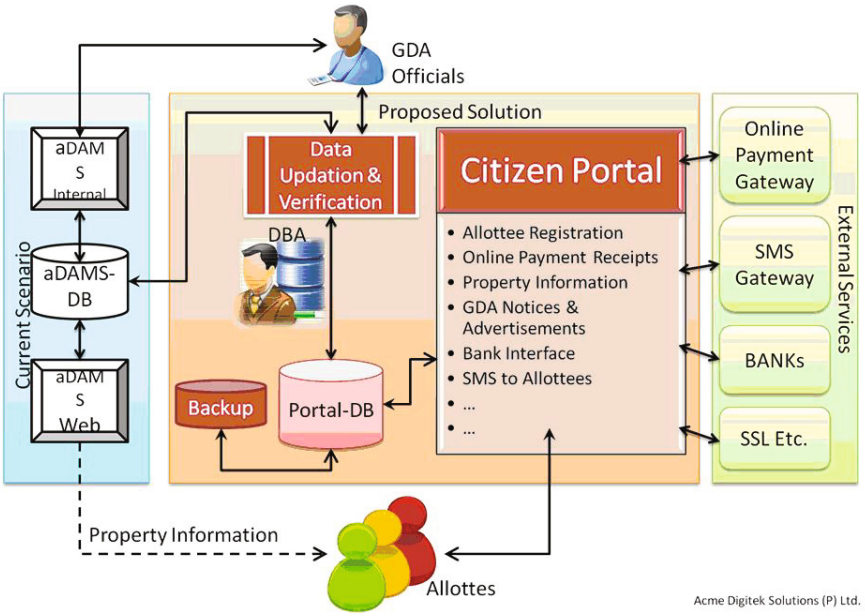


Fig. 1 Citizen portal solution for Ghaziabad development authority

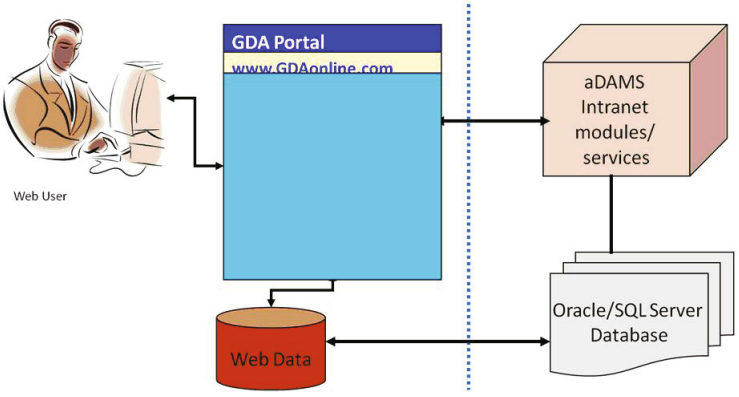


Fig. 2 Web portal access mechanism

The cloud platform of the ACME DAMS Solution was, an application server which is a Windows server; a database server which is a MS SQL server; and a technology known as Microsoft.Net. The citizen portal solution of Ghaziabad development authority is shown in Figure 1 whereas web portal access mechanism is depicted in Figure 2.

3 The Cloud Solution

Cloud environment not only helped GDA scale up their computing requirements on demand but also reduced the cost of operations considerably. The design of the cloud helped creating a robust and fault tolerant infrastructure in a high availability mode. Based on interviews, the researchers found that the service provider tied up with the cloud solution providers to host the customer applications on a high Availability infrastructure. The cloud based solution on VMware cloud architecture was a dedicated cluster of highly available, scalable, secure and redundant server environment. Cloud solution was an all inclusive solution which included provisioning, setup, licensing, monitoring and complete management of the cluster, thereby guaranteeing enhanced security, improved back-up options, experienced technical support for maintenance, better equipment and zero downtime. A pictorial representation on the cloud solution is presented in the following Figure 3.

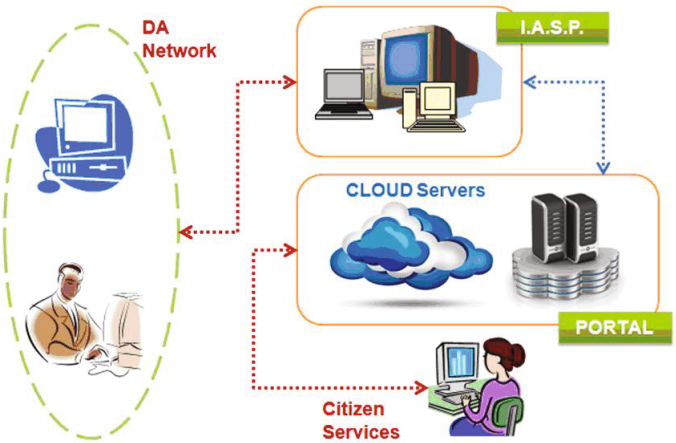


Fig. 3 The cloud solution

The cloud solution gave a 'high availability (HA)' and 'fault tolerance' features. Both these features have the ability to offer continuous availability for hosted virtual machines in case of a host failure. The high availability provides a base level protection for hosted VMs by restarting them in the event of a host failure. VMware fault tolerance provides a higher level of availability, allowing users to protect any VM from a host failure with no loss of data, transactions, or connections. The movement of servers in high availability environment is depicted in Figure 4.

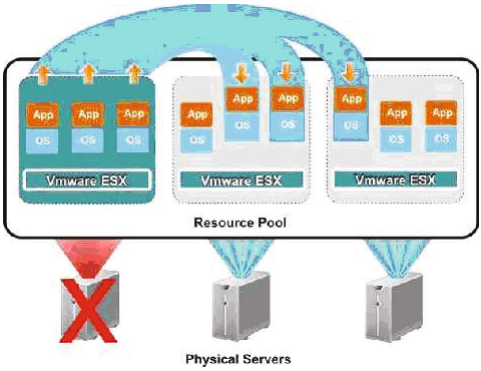


Fig. 4 The movement of servers in a high availability environment

3.1 Technical Solution Architecture

The developed system was implemented with the following hardware and system software. These are cloud servers, application server with 1 KVA on-line UPS with 1 hour battery backup, database server with 1 KVA on-line UPS with 1 hour battery backup, Windows 2008 Server 32 Bit MOLP standard edition with 5 User license with downgrade features, Microsoft SQL server standard 2008 with 5 CAL, desktop computers (Clients), laser printers, and networking in office. The development authority network at Ghaziabad development authority is depicted in Figure 5.

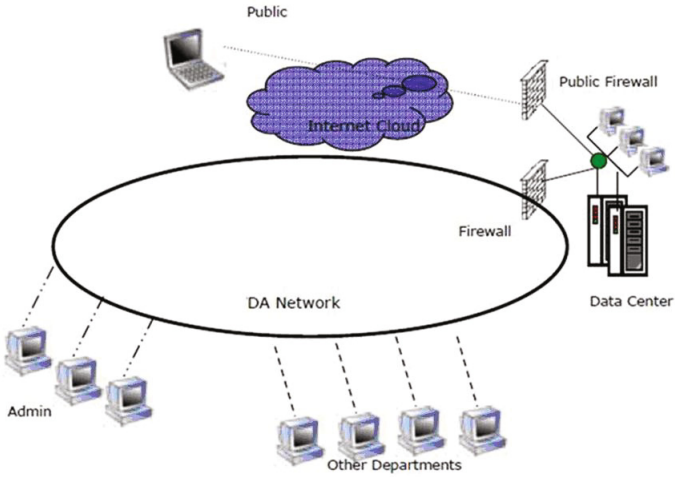


Fig. 5 The development authority network at GDA

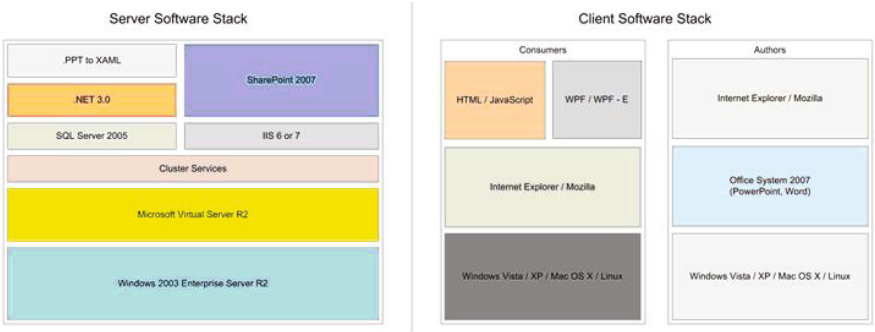


Fig. 6 The logical solution architecture at GDA

The aDAMS solution was designed with a combination of two-tier and three-tier architecture approach depending on the need of information flow at various stages by various categories of the users. The solution mainly works on three-tier architecture. At the backend, the database servers and storage, preferably on a SAN cluster for implementing 99.99% availability is used. The database shall be MS-SQL server. In the middle tier, the application servers reside and the other existing servers used for mail / messaging, proxy, antivirus, domain controller etc. The front tier consists for the clients running the web browser. The overview of the complete application may be viewed as shown in the following Figure6. The logical solution architecture, as shown in Figure 6, is accessible over the internet or over a LAN or WAN with full security.

3.2 The Modular aDAMS Solution

The aDAMS consists of various modules which provide connectivity between various departments and the functions of the Ghaziabad Development Authority. It is modular in nature and can be implemented in phases as per the priority decided by higher authorities. The modules covered are as shown in the following Figure 7.

The dependencies between various modules is depicted in the following Figure 8. The compulsory modules are property management and disposal, online property information, budget and financial accounting syatem, and building plan approval system. These are the core and important modules for GDA as of now. The optional modules include legal case monitoring, right to information, and personnel information and payroll system. These modules shall be used additionally by development authorities. As of now these modules are being used by GDA. The other modules are not being used by the GDA at present.

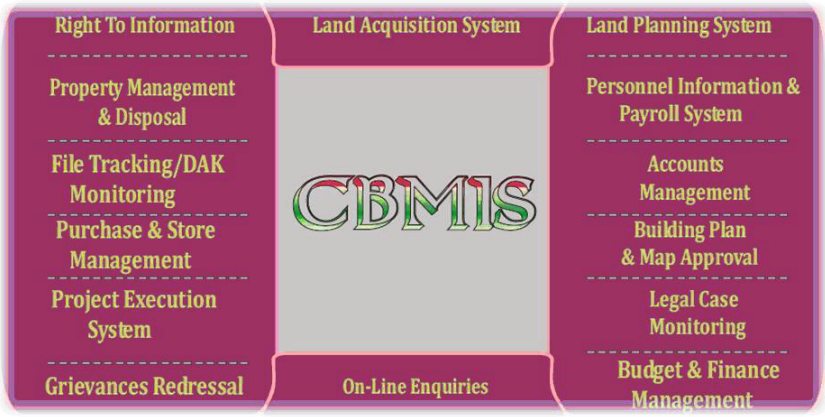


Fig. 7 A modular aDAMS solution

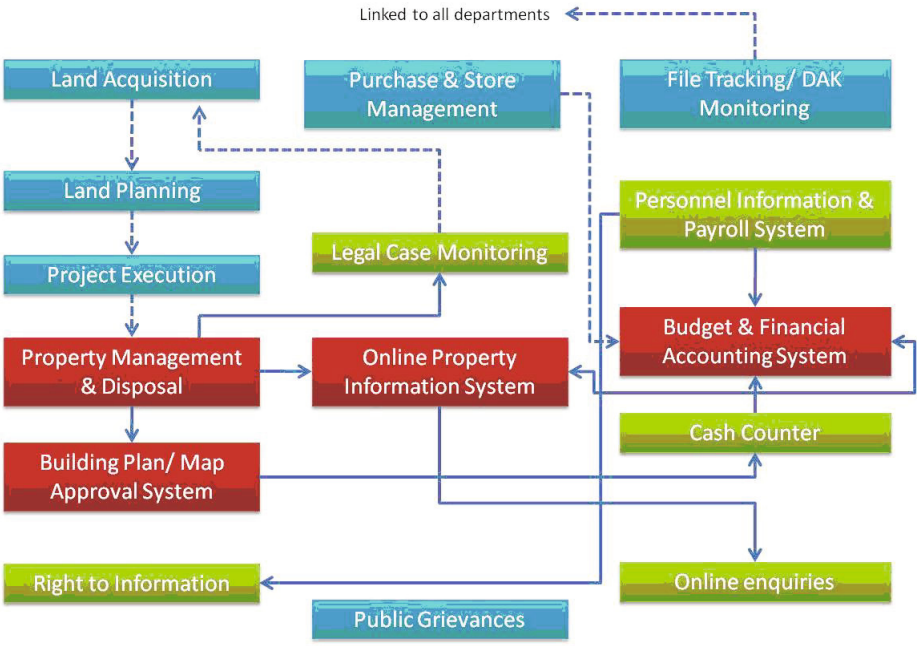


Fig. 8 Dependencies between modules

Table 1 Costing analysis: Design and build model against SaaS and cloud model

Module	A	category	authority		B	category	authority	
	Design build	and	SAAS cloud	and	Design build	and	SAAS cloud	and
	Appl.	AMC	Appl.	Monthly cost	Appl.	AMC	Appl.	Monthly cost
Property management and disposal	20.00	4.00	10.00	1.50	16.00	3.20	8.00	1.20
Building plan / Map approval	5.00	1.00	2.50	0.38	4.00	0.80	2.00	0.30
PIS and payroll	5.00	1.00	2.50	0.38	4.00	0.80	2.00	0.30
Budget and financial accounting	5.00	1.00	2.50	0.38	4.00	0.80	2.00	0.30
Right to information	3.00	0.60	1.50	0.23	2.40	0.48	1.20	0.18
Legal case monitoring	4.00	0.80	2.00	0.30	3.20	0.64	1.60	0.24
Cash counter	3.00	0.60	1.50	0.23	2.40	0.48	1.20	0.18
Total	45.00	9.00	22.50	3.38	36.00	7.20	18.00	2.70

The different characteristics of this e-governance are listed below.

- GDA is not using any other software or application, hence these modules are not dependent to any other external module.
- Online property information system (OPIS) is linked with third party SMS API interface to send SMS, and with ICICI payment gateway to collect online payment.
- Dotted arrow in the dependency modules represents that, at present there is no link between the modules but the link should be there, and it shall be created in future when required.
- Solid arrow in the dependency module represents data sharing between modules. There is both way data sharing between budget and financial accounting and OPIS.
- Since file monitoring, public grievances and some other modules are not being used by GDA at present, for this reason they are not connected to any other modules.

The service provider capability to be able to create a solution design which is modular, flexible and customizable will help the service provider generate economies of scale and scope. This indicates a strong business sense in the service provider which helped them manage sustainability of the service provisioning model. Previous researches have shown that the failure of service providers to be able to

manage sustainability leads to service discontinuity or degradation in quality of software services provided. A modular design enables replication, which supports the multitenant structure of the SaaS solution and allows for economies of scale. A modular design also enables customization, by allowing the customer to choose modules, which allows for economies of scale. So where for the customer a modular design means customization and consistency of software services, for the service provider a modular design is able to generate economies of scale and scope and thus ensures sustainability of the SaaS model of software services provisioning.

The following Table 1 shows that cost savings are one of the important motivations for the customer. The data in Table 1 shows that although the recurring costs are almost the same for inhouse implementation and cloud implementation, but the initial capital expenditures is one third in case of cloud implementation.

4 Conclusion

The development authorities are facing a high rate of increase in data and information. This chapter discusses a cloud based e-governance solution. Since the study is based on a case study, the generalization of the results might be limited. The framework needs to be tested in more organizations and contexts. SaaS based enterprise resource planning (ERP) adoption is at a very nascent stage in India. Therefore, a quantitative study was not possible. In this study we only used data triangulation. It would strengthen the framework if theoretical and methodological triangulation could also be employed.

References

1. Dubey, A., Wagle, D.: Delivering software as a service. The McKinsey Quarterly 6, 1–7 (2007)
2. Mell, P., Grance, T.: The NIST definition of cloud computing. NIST special publication. 800-145, 1–3 (2011)
3. Benlian, A., Hess, T.: Opportunities and risks of software as a service: Findings from a survey of IT executives. Decision Support Systems 52(1), 232–246 (2011)
4. Ramachandran, M.: Advances in cloud computing research. Nova Science Publishers, USA (2014)
5. Chong, F., Carraro, G.: Architecture strategies for catching the long tail. MSDN Library, Microsoft Corporation. pp. 9–10 (2006)
6. Ramachandran, M., Chang, V.: Modelling financial SaaS as service components. In: Chang, V., Wills, G., Walters, R. (eds.) Emerging Software as a Service and Analytics, pp. 13–20. SCITE Press, Portugal (2014)
7. Loh, L., Venkataraman, N.: Determinants of information technology outsourcing: A cross sectional analysis. Journal of Management Information Systems 9(1), 7–24 (1992)
8. Barett, L.: Datamation. QuinStreet, Inc. (2010)
9. Dubey, A., Mohiuddin, J., Baijal, A., Rangaswami, M.: Enterprise software customer survey 2008. Customer survey. McKinsey and Company, SandHill Group (2008)

10. Alateyah, S.A., Crowder, R.M., Wills, G.B.: Factors affecting the citizen's intention to adopt e-government services in Saudi Arabia. International Journal of Social, Management, Economics and Business Engineering 7(9), 1287–1292 (2013)
11. Weerakkody, V., Dwivedi, Y.K., Kurunananda, A.: Implementing e-government in Sri Lanka: Lessons from the UK. Information Technology for Development 15(3), 171–192 (2009)
12. Weerakkody, V., Dwivedi, Y.K., Brooks, L., Williams, M.D., Mwange, A.: E-government implementation in Zambia: contributing factors. Electronic Government 4(4), 484–508 (2007)

Author Index