

Big Data Clustering Based Recommendation System Model Through Correlations

A Thesis

submitted in partial fulfillment of the requirements for the award of degree of

Doctor of Philosophy

by

Divya Pandove

(901203008)

under the guidance of

Dr. Rinkle Rani

Associate Professor, Computer Science and Engineering Department,

Thapar University, Patiala, Punjab, India

Dr. Shivani Goel

Professor, Department of Computer Science Engineering,

Bennett University, Greater Noida, U.P., India



Computer Science and Engineering Department

Thapar University, Patiala, India

July 2017

Contents

List of Figures	ix
List of Tables	xii
List of Algorithms	xiii
Certificate	xiv
Acknowledgements	xv
Abstract	xvii
1 Introduction	1
1.1 Data clustering as a computational problem	2
A mathematical theorem for clustering	3
1.1.1 Clustering tendency	3
1.1.2 Cluster validation	4
1.2 Significance of correlations in data	5
1.3 Correlation clustering	6
1.3.1 Problem of correlation clustering	7
1.3.2 Solutions to the problem of correlation clustering	7
1.4 Challenges of dealing with high-dimensional data	17
1.4.1 Curse of dimensionality	18
1.5 Thesis Organization	20

2	Literature Review	24
2.1	Correlations in data	24
2.2	Clustering large-scale datasets	28
2.2.1	Related research	29
2.2.2	Clustering approaches for big data mining	30
	Hierarchical approach	31
	Hierarchical clustering algorithms to handle large data sets	34
	K-means clustering approach	35
	Random sampling approach	36
	Randomized search approach	36
	Condensation-based approach	37
	Density-based approach	37
	Clustering graphs and network data	37
	Grid-based approach	39
	Probabilistic model-based clustering methods	39
	Hadoop and big data	40
2.3	Clustering high-dimensional datasets	53
2.4	Applications of large-scale and high-dimensional data clustering	59
2.5	Latest trends in large scale and high-dimensional data clustering	65
3	Problem Formulation	71
3.1	Research Gaps	72
3.2	Research objectives	73
3.3	Methodology for achieving the objectives	73
4	Preliminaries	80

5	Proposed Correlation Clustering Approaches	89
5.1	General Correlation Coefficient based Agglomerative Clustering (GBAC)	90
5.1.1	General (rank based) correlation coefficient (G)	91
5.1.2	Proposed framework for GBAC	98
5.2	Local Graph based Correlation Clustering (LGBACC)	100
5.2.1	Proposed framework for LGBACC	102
5.3	Conclusion	105
6	Experiments and Implementation Details	106
6.1	Datasets	106
6.2	Evaluation of General (Rank based) Correlation Coefficient	109
6.2.1	Performance analysis	110
6.3	Evaluation of GBAC framework	115
6.3.1	Performance analysis	117
6.3.2	Cluster quality analysis	122
6.4	Evaluation of LGBACC framework	124
6.4.1	Performance analysis	125
	PCA and Graph analysis phase	125
	Cluster analysis	130
	Cluster quality assessment	133
6.5	Evaluation of scalability for proposed approaches	137
6.6	Comparison between the proposed approaches	144
7	Recommendation system model	145
7.1	Preliminaires	145
7.1.1	Model Explanation	148

7.1.2	Experimental Evaluation	150
8	Conclusion and future scope	155
	References	159

List of Figures

1.1	Interpretation of a value of a correlation coefficient	6
1.2	Solutions to the problems of correlation clustering	8
1.3	Edge errors as a result of functional imperfections	11
1.4	Process of pivoting in KWIKCLUSTERING algorithm	13
1.5	Representation of clustered dataset in 3-D space	18
1.6	Projection on subspace (x,y)	18
2.1	Clustering approaches for big data	32
2.2	Single linkage hierarchical clustering	34
2.3	Complete linkage hierarchical clustering	34
2.4	Average linkage hierarchical clustering	34
2.5	Clustering high-dimensional data sets	54
2.6	A generative graphic model	62
2.7	Mapping of clustering approaches to specific applications	64
3.1	Flow of the work done	74
3.2	Layout of experimental results and discussion	78
3.3	Mapping of objectives with publications	79
4.1	A flowchart depicting the process of PCA	84

4.2	Graph Laplacian of G	88
5.1	Criteria for a predictive metric	91
5.2	The hierarchy of need and motivation for G	92
5.3	Proposed framework for General correlation coefficient based agglomerative clustering	99
5.4	Proposed framework for local graph based correlation clustering	101
6.1	Mind map representing attributes of World Bank dataset	107
6.2	Scatterplot for USArrests dataset	108
6.3	Scatterplot for Mtcars dataset	108
6.4	Scatterplot for Wages	109
6.5	Scatterplot for El Nino	109
6.6	Comparison of G with ρ and τ for Binomial distribution	113
6.7	Comparison of G with ρ and τ for Exponential distribution	113
6.8	Comparison of G with ρ and τ for Normal distribution	113
6.9	Comparison of G with ρ and τ for Poisson distribution	113
6.10	Comparison of G with ρ and τ for Uniform distribution	113
6.11	Graphical representation of comparison of correlation coefficients on the World Bank dataset	115
6.12	Heat map for USArrests dataset	118
6.13	Correlation plot using G	118
6.14	Banner for AGNES on USArrests dataset	118
6.15	Dendrogram for USArrests dataset	118
6.16	Bk plots of Hierarchical clustering based on G v/s standard hierarchical clustering for USArrests dataset	119

6.17 Heat map for Mt cars data set	119
6.18 Correlation plot using G	119
6.19 Banner for AGNES on Mtcars	120
6.20 Dendrogram for data set Mtcars	120
6.21 Bk plots of Hierarchical clustering based on G v/s standard hierarchical clustering for Mtcars dataset	120
6.22 Banner for AGNES on World Bank dataset	122
6.23 Dendrogram for World Bank dataset	122
6.24 Bk plots of Hierarchical clustering based on G v/s standard hierarchical clustering for World Bank dataset	123
6.25 PCA for USArrests dataset	128
6.26 PCA for Wages dataset	128
6.27 PCA for El Nino dataset	129
6.28 PCA for World Bank dataset	129
6.29 Graph for USArrests dataset	130
6.30 Graph for Mtcars dataset	130
6.31 Graph for Wages dataset	130
6.32 Graph for El Nino dataset	130
6.33 Graph for World Bank dataset	130
6.34 Cluster analysis of USArrests dataset	131
6.35 Cluster analysis of Mtcars dataset	132
6.36 Cluster analysis of Wages dataset	133
6.37 Cluster analysis of El Nino dataset	134
6.38 Cluster analysis of World Bank dataset	135
6.39 Graph representing clusters formed in USArrests dataset	135

6.40	Graph representing clusters formed in Mtcars dataset	135
6.41	Graph representing clusters formed in Wages dataset	136
6.42	Graph representing clusters formed in El Nino dataset	136
6.43	Graph representing clusters formed in World Bank dataset	136
6.44	Bk plots of traditional hierarchical clustering algorithms v/s LGBACC for USArrests dataset	138
6.45	Bk plots of traditional hierarchical clustering algorithms v/s LGBACC for Mtcars dataset	138
6.46	Bk plots of traditional hierarchical clustering algorithms v/s LGBACC for Wages dataset	139
6.47	Bk plots of traditional hierarchical clustering algorithms v/s LGBACC for El Nino dataset	139
6.48	Bk plots of traditional hierarchical clustering algorithms v/s LGBACC for World Bank dataset	139
6.49	SparkR architecture used for experimentation	140
6.50	Correlation plot showing correlation between the clusters	141
6.51	Clustering by GBAC	141
6.52	Clustering by GBAC	141
6.53	Clustering on first half of LGBACC, before performing PCA	141
6.54	Percentage of variance v/s Dimensions	142
6.55	Variance factor map for PCA	142
6.56	Hierarchical clustering on the factor map	142
6.57	Cluster plot of the distinct clusters	142
6.58	Execution time of clustering algorithms on DS1	143
6.59	Execution time of clustering algorithms on DS2	143

6.60	Execution time of clustering algorithms on DS3	143
6.61	Execution time of clustering algorithms on DS4	143
7.1	Recommender System Model	149
7.2	Precision-Recall curve for LGBACC	154
7.3	Precision-Recall curve for GBAC	154
7.4	ROC curve for LGBACC	154
7.5	ROC for GBAC	154
7.6	Comparison based on Precision-recall	154
7.7	Comparison based on ROC	154

List of Tables

1.1	Fundamental results of various correlation clustering approaches	16
2.1	Comparative analysis of correlation measures used on different datasets . .	28
2.2	Comparative study of the existing research/surveys	31
2.3	A comprehensive comparison of the clustering approaches for big data . . .	43
2.3	A comprehensive comparison of the clustering approaches for big data . . .	44
2.3	A comprehensive comparison of the clustering approaches for big data . . .	45
2.3	A comprehensive comparison of the clustering approaches for big data . . .	46
2.3	A comprehensive comparison of the clustering approaches for big data . . .	47
2.3	A comprehensive comparison of the clustering approaches for big data . . .	48
2.3	A comprehensive comparison of the clustering approaches for big data . . .	49
2.3	A comprehensive comparison of the clustering approaches for big data . . .	50
2.3	A comprehensive comparison of the clustering approaches for big data . . .	51
2.3	A comprehensive comparison of the clustering approaches for big data . . .	52
2.4	Comparison of properties of clustering algorithms	70
5.1	Symbol table for derivation of denominator D	94
6.1	Summary of datasets	109

6.2	Simulations on random number distributions to compare correlation coefficients	112
6.3	Comparison of correlation coefficients on World Bank dataset	114
6.4	Descriptive statistics for USArrests dissimilarity matrix	118
6.5	Descriptive statistics for dissimilarity matrix of Mtcars dataset	119
6.6	Descriptive statistics for dissimilarity matrix of World Bank dataset	120
6.7	Correlation matrix and dissimilarity matrix with attributes of World Bank dataset	121
6.8	Agglomerative Coefficient table	123
6.9	FM Index table	124
6.10	Execution time and memory requirement of datasets	124
6.11	PCA statistics for Wages dataset	126
6.12	PCA statistics for El Nino dataset	126
6.13	PCA statistics for Mtcars dataset	127
6.14	PCA statistics for World Bank dataset	127
6.15	PCA statistics for USArrests dataset	128
6.16	Descriptive statistics and Agglomerative Coefficient	134
6.17	Fowlkes-Mallow index table	137
6.18	Execution time and memory requirement of datasets	138
6.19	Comparison of cluster quality for the proposed algorithms	140
6.20	Description of datasets	142
6.21	Comparison between different correlation clustering algorithms	144
6.22	Comparison between proposed algorithms	144
7.1	Representation of confusion matrix	147
7.2	Descriptive statistics of movies.csv	151

7.3	Descriptive statistics of ratings.csv	151
7.4	Quality parameters for the two approaches	151
7.5	Evaluation of predicted values	152

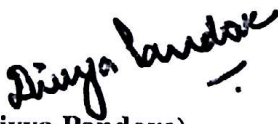
List of Algorithms

- 1.1 Constant factor approximation for minimizing number of disagreements . . . 9
- 5.2 General (rank based) correlation coefficient calculation 94
- 5.3 G based Correlation Clustering algorithm (GBAC) 100
- 5.4 Local Graph based Correlation Clustering (LBACC) Algorithm 103

Certificate

I hereby certify that the work which is being presented in this thesis entitled “**Big Data Clustering Based Recommendation System Model Through Correlations**”, in partial fulfillment of the requirement for the award of degree of “Doctor of Philosophy” submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Rinkle Rani, Dr. Shivani Goel and refers other research works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.


(Divya Pandove)

Regn. No. 901203008

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

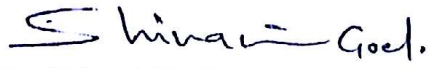

(Dr. Rinkle Rani)

Associate Professor

Computer Science & Engineering

Department, Thapar University,

Patiala, Punjab, India


(Dr. Shivani Goel)

Professor

Department of Computer Science

Engineering, Bennett University,

Greater Noida, U.P., India

Acknowledgement

I would like to express sincere appreciation to my supervisors, **Dr. Rinkle Rani** and **Dr. Shivani Goel**, for being pillars of support and encouragement throughout my research work. Their experience, strength, tenderness and willfulness, have taught me valuable lessons of life, that are going to be of immense help to me in taking decisions throughout my life.

My sincere thanks are due to **Dr. Maninder Singh**, Professor and Head, CSED, Thapar University for providing me the necessary administrative assistance and infrastructure, that helped me in completion of my research work. I am thankful to my doctoral committee members, **Dr. Sushma Jain**, Assistant Professor, CSED, **Dr. Anju Bala**, Assistant Professor, CSED, and **Dr. Kulbir Singh**, Professor, ECED, Thapar University, Patiala for their constructive suggestions and regularly ensuring the progress of my research work. I am thankful to all the **faculty**, **staff** members and **research scholars** of CSED, Thapar University for their support.

I offer my deepest gratitude to my loving father, **Mr. R.P. Pandove** whose love and constant encouragement has been a major source of inspiration in turning my vision into a reality. I am also thankful to my mother **Mrs. Sangeeta Pandove**, my Badi Ma **Mrs. Manju Pandove**, my sister **Tanya** and my brothers **Akul** and **Kunal** for their love, encouragement, motivation and confidence in me. I am blessed to have powerful mentors in the form of my uncles **Mr. M.P. Pandove** and **Mr. K.P. Pandove**. I thank them both for inspiring me to always strive for achieving bigger things in life.

My new family has given me immense support and encouragement to complete this mammoth task, I am extremely thankful to my mother-in-law **Mrs. Rita Narula**, father-in-law **Mr. Ashok Narula** and my husband **Rahul Narula**. I want to express my sincere appreciation to Rahul for his patience, support and positive attitude towards my work. I am truly fortunate to have him as my life partner.

I also acknowledge the cooperation and encouragement extended to me by my friends, especially **Dr. Avleen Kaur Malhi**, for always being the biggest pillar of support throughout this long journey, and **Siddharth Singh Thakur**, for pushing me in the right direction. The chain of gratitude would be definitely incomplete without thanking the **Almighty**, the prime mover, for inspiring and guiding me (a humble being) to complete this task successfully.

Patiala

July, 2017

Divya Pandove
(Divya Pandove)

Abstract

Technological advancement has enabled us to store and process huge amounts of data in relatively short spans of time. The nature of data is rapidly increasing its dimensionality to become multi and high-dimensional. There is an immediate need to expand our focus to include analysis of high-dimensional and large datasets. Data analysis is becoming a mammoth task as a result of incremental increase in data volume and complexity in terms of heterogeneity of data. It is because of this dynamic computing environment that the existing techniques either need to be modified or discarded to handle new data in multiple high-dimensions. Data clustering is a tool that is used in many disciplines, including data mining, so that meaningful knowledge can be extracted from seemingly unstructured data. Correlation clustering possibly represents the most intuitive form of clustering construction. It gives solutions that can be approximated while automatically selecting the number of clusters. This approach handles scenarios where the focus is on relationships between the objects instead of on actual representations of the objects. The suitability of this method extends to the structured objects for which feature vectors are not easy to obtain. Given the increasing scale of data these days, correlation clustering has become a powerful addition to the fields of data mining and agnostic learning. In this thesis, we start by proposing an algorithm that defines an intuitive and accurate correlation coefficient metric, known as the General (rank based) correlation coefficient (G). Further, a framework is proposed, based on this algorithm, and is named as G Based Agglomerative Clustering (GBAC). Our approach has been found to be effective for small, large and high-dimensional data that generate high quality clusters. This framework combines the predictive power of correlation coefficients with the ability to find patterns in data obtained from agglomerative hierarchical clustering.

To explore complex relationships in data, there is a need to integrate dimensionality-reduction techniques with data-mining approaches and graph theory. We propose another ap-

proach called Local Graph based Correlation Clustering (LGBACC). This approach merges hierarchical clustering with PCA to uncover complex hierarchical relationships and uses graph models to visualize the results. Visualization of data is an important output and is knitted into the fabric of the framework. LGBACC is found to produce high quality clusters across a wide spectrum of dimensionality. Finally, both of these algorithms have been tested on real-life datasets using distributed and parallel computing (SparkR). We have used four large datasets (varying between 260 GB and 1 TB) to prove the scalability of the proposed approaches. They are found to be scalable and performing better than the existing hierarchical clustering algorithms. These algorithms are then integrated into recommendation algorithms and we define a recommendation system model through correlations. This model has been validated using a real-time, large dataset, and the results prove that combining correlated points with the predictive power of recommendation algorithms produces better-quality recommendations.

Chapter 1

Introduction

Data clustering is a very important form of knowledge discovery tool. A plethora of approaches that deal with the implementation of the data clustering process. An article concerning anthropological data was published in 1954. It had the term data clustering in its title. This was the first time when this term came into print [1]. Cluster analysis has its origin in the domains such as machine learning, artificial intelligence, data mining, biology, statistics etc. Different fields use distinctive names for cluster analysis. Some of them are: Q Analysis, data visualization, typology, numerical taxonomy, clumping, etc. [2] [3]. Clustering is a familiar form of data mining for the classification of data objects. This classification is on the basis of similarity and dissimilarity amongst data objects. In data mining, data objects are represented as points. The point to note is that the classification of data into clusters is unsupervised and can be denoted as:

$$X = C_1 \cup \dots \cup C_i \cup C_n; C_i \cap C_j = \Phi (i \neq j),$$

X denotes original dataset C_i, C_j are clusters formed in X and n denotes the number of clusters formed [4]. Unsupervised knowledge means that one can have previous information of the domain of the data but not of the structural characteristics. The characteristic most commonly unknown is the spatial distribution of data. It represents information about a cluster

such as its volume, density, shape or orientation [5].

1.1 Data clustering as a computational problem

Clustering is a common form of data mining having applications in many fields such as data compression, computer vision, texture segmentation, and vector quantization. Most of the clustering methods become computationally intractable with an increase in the problem scale, as these methods have combinatorial nature [1]. For some specific objective functions, if the number of clusters exceeds three, clustering becomes an NP hard problem. This may be explained as when a program based on an algorithm is devised and run, it may deliver optimal results when tested with test examples but when using real time datasets, the test results may vary from the expected results. This may be because of an inefficient algorithm or a complex computational problem, or a combination of the two [6]. While solving the cluster analysis problem, we take the original set of data points and partition them to form k groups. These groups are mutually exclusive and exhaustive in nature. Most common criterion to determine the heterogeneity of a given cluster is the comparison of distances. The three possible conditions are:

1. Define the maximum cluster distance and minimize it
2. Calculate the sum of averages of the cluster distances and minimize it
3. Calculate the total cluster distance and minimize it

When we take the possible number of clusters as ' k ', the computational analysis of all partitions demonstrated by all the algorithms give good solutions but they may or may not be optimal solutions [7].

A mathematical theorem for clustering

Let there be a partition P_k , that divides the data space into k clusters. It is assumed that it is a perfect partition. It is given that all the distances amongst data points of a cluster are zero. There is no condition on the distance between the cluster distances. Hence the clusters have to be separated well. Let $d(P_k)$, be the class of the criteria “minimize”, where $d(P_k)$ is a function that calculates distances within a cluster such that $d(P_k) \geq 0$ and $d(P_k) = 0$, if and only if, (P_k) is a perfect partition. This can be written in the form of a theorem as: Optimal clustering is NP- Hard for any criterion minimize $d(P_k)$ such that $d(P_k) \geq 0$ and $d(P_k) = 0$ if and only if P_k is a perfect partition [8].

1.1.1 Clustering tendency

The data must be tested to see if it has some inherent grouping structure, to determine clustering tendency or clusterability of data. It is a hard task as there are so many different definitions of clusters. Some clusterability assessment methods can be used for this purpose, discussed as follows [9]:

1. Spatial Histogram: In this approach, the histogram generated from random samples is contrasted with a d -dimensional histogram generated from the input dataset D . Dataset D , is clusterable if the distributions of the histograms are rather different. Each dimension is divided into q -width bins, and the number of points in each cell are counted. Then, the empirical joint probability mass function (EPMF) is calculated. This is done for the randomly sampled data as well. The difference in the two is calculated by using the Kullback-Leibler (KL) divergence value. If the two values are different then, we say that the dataset is clusterable [10].
2. Distance distribution: Compare the pairwise point distance from the data with those

from randomly generated samples.

3. Hopkins statistic: The spatial randomness is tested by using this sparse sampling test.

1.1.2 Cluster validation

It has been observed that different clustering algorithms tend to produce different results when applied to same datasets. Even when a given algorithm is altered by changing its parameters, it tends to produce different results. This goes on to prove the *impossibility theorem*, which states that “No single clustering algorithm simultaneously satisfies the three basic axioms of data clustering, i. e., scale-invariance, consistency and richness” [11]. Another issue with clustering validation is that, being an unsupervised method, there is no expert to judge goodness of cluster. While evaluating a cluster, the goodness of the clustering is measured. It can be done by evaluating the clustering stability for understanding the clustering result’s sensitivity to different parameters of algorithm, i.e., number of clusters etc. Clustering tendency is another factor for assessing the cluster suitability, i. e., whether data has any inherent grouping structure or not. To validate the clusters produced by the clustering algorithms, we broadly consider three software engineering criteria:

1. Internal indices: The clusters obtained from a given clustering algorithm are compared with data only; it is an unsupervised criterion derived from data itself. Goodness of the clustering can be evaluated by understanding cluster separation and compactness. Silhouette Coefficient is a good example of this case.
2. External indices: External knowledge is collected over time to provide a knowledge base. A comparison is made between this knowledge base, and the clusters obtained from the application of the clustering algorithm. It is a supervised index and employs criteria not inherent to the dataset.

3. Relative indices: A comparison of clusters produced by applying different clustering techniques on the same datasets. They are used to compare different clusterings, generally retrieved through distinct settings of parameters when the algorithm considered is the same. For example, different number of clusters for the same algorithm may generate different results.

1.2 Significance of correlations in data

The practice of finding correlations between data points has been prevalent for a very long time. This concept was introduced by Sir Francis Galton, who noticed a relationship between the height of a man and the length of his forearm [12]. The only thing that has changed is the volume of data and the availability of better tools for analysis and data storage. As a result, correlations in data emerge more rapidly, without incurring much cost. Today, the focus of methods for finding correlations in the data is centered on the “what” part of the data and not the “why” part. This means that to successfully find correlations in a given dataset, it is not important to understand all the underlying variables of the system and their relationships (the “why” part of data). We just need to find the variables that help in making predictions (the “what” part of data). The correlations in these variables help us find patterns in data. The “why” aspect of data might be interesting and appealing to the human mind, but it does not generate valuable insights into relationships between data points. Instead of focusing on finding the cause-and-effect relationships in a particular data collection, we try to find patterns and correlations. This helps us visualize links in data that have not been seen before. The premise of this approach is that causality can rarely be proved [13]. Though correlations play an important part in analyzing small datasets, they truly shine when dealing with large data. Nowadays, experts are developing necessary tools to identify and compare non-linear

correlations. The techniques of analysis are being aided and enhanced by fast-growing novel approaches, and software that can extract non-causal relationships in data [14].

The value of a correlation coefficient implies a dependency relationship between two data values. If the value of correlation is high, it means that when one datum value changes, the likelihood of the other one changing is also very high. A weak correlation implies that there might be little-or-no effect of one variable on the other, i.e., there is little-to-no dependency between the two variables. A correlation coefficient is used to express the degree of dependence between the two variables. Figure 1.1 shows the interpretation of various values of a correlation coefficient. These values lie between -1 and 1. With poor correlations, there

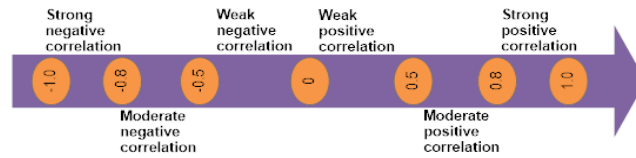


Figure 1.1: Interpretation of a value of a correlation coefficient

is low probability of similarity between objects. But if the correlation is strong, then the probability of similarity is quite high. To fully utilize the power of correlations, there is a need to understand the relationship between the two variables under consideration.

1.3 Correlation clustering

Correlation clustering is a form of subspace clustering. It represents the most intuitive form of clustering construction. It gives solutions that can be approximated while automatically selecting the number of clusters. This approach handles scenarios where the focus is on relationships between the objects, instead of on actual representations of the objects. The suitability of this method extends to the structured objects for which feature vectors are not easy to obtain. Given the increasing scale of data these days, correlation clustering has

become a powerful addition to the fields of data mining and agnostic learning. The task is to find a clustering that either maximizes agreements or minimizes disagreements. Many methodologies, such as approximations and linear programming formulations, have been used to approach this problem. Correlation clustering algorithms integrate dimensionality reduction algorithms with clustering approaches.

1.3.1 Problem of correlation clustering

The problem of correlation clustering was defined by [15] in 2002. The authors consider a fully connected graph as having ‘ n ’ vertices. Every edge (u,v) is assigned a ‘+’ label if the vertices are similar, and a ‘-’ label if they are different. The basic aim is to produce a clustering or partition of vertices that agree with the edge labels as much as possible. This means that the desired partition maximizes the number of positive edges. The summation of the number of positive edges within clusters and the number of negative edges between the clusters should be maximum. This representation is that of maximization of number of agreements. When the aim is to minimize disagreements, summation of the number of ‘-’ edges inside clusters and the number of ‘+’ edges between clusters is taken as the goal. The maximization and minimization goals may seem equivalent in terms of optimality but are different from an approximation view point. A unique feature of this type of clustering is that, unlike many previous approaches, it does not take the number of clusters as a parameter.

1.3.2 Solutions to the problem of correlation clustering

Over the years, many solutions have been proposed and proved. A mind map is used to summarize various solutions for the problem of correlation clustering in Figure 1.2. It shows the connections linking the techniques and also how they differ from one another.

- Minimizing disagreements and maximizing agreements: [16] proposed solutions on

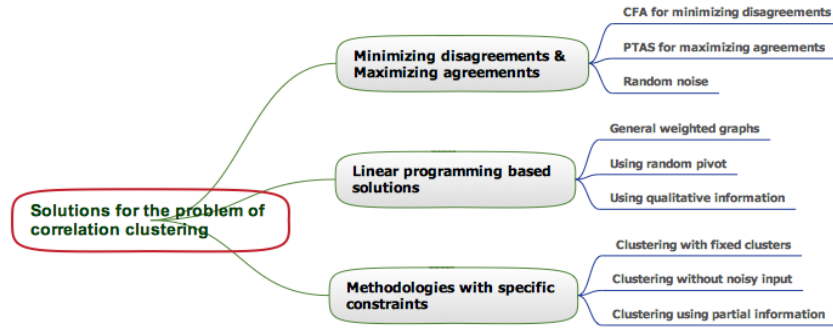


Figure 1.2: Solutions to the problems of correlation clustering

the same lines as the solutions of the problems of minimizing disagreements and maximizing agreements, both of which are NP-Complete problems. Agreements and disagreements are evaluated by two variants: weighted and un-weighted. The NP hardness of the un-weighted variant can be proved by reduction of the problem of partition into triangle, to the problem of minimizing disagreements. The weighted variant assigns arbitrary weights to edges with an aim to minimize weighted disagreements and reduce the problem to multiway cut in order to prove NP-hardness [17]. The approaches proposed by the authors [16] are discussed below:

- A constant factor approximation (CFA) for minimizing number of disagreements: In this approach, a 3-approximation is applied to two clusters in order to obtain the best (clean) clustering. This means that if the cluster partition obtained by two clusters (best), when represented in a graph contains x mistakes, then the approach used for minimizing disagreements will produce at most $3x$ mistakes. For an n -approximation, the algorithm will produce not more than nx mistakes. ‘OPT(2)’ represents the best clustering that contains two clusters, C_1 and C_2 . All the clusters are considered in the form of $(N^+(v), N^-(v))$ for $v \in V$. The output should contain minimum mistakes. The methodology for minimizing the number of disagreements (using constant factor approximation) mainly consists of three steps. Algorithm 1.1 contains the detailed

flow of the constant factor approximation for minimizing disagreements.

Algorithm 1.1 Constant factor approximation for minimizing number of disagreements

Ensure: : A Graph containing $(n - 1)$ clusters, for best clustering, it has n -approximation

Require: An algorithm that will produce clustering with not more than ‘ nx ’ mistakes

A) Clean clusters are used to provide a clustering for a portion of graph, ‘ OPT ’ bounds the mistakes present in this portion and fulfills the following conditions:

1. If $r_1, r_2, \dots, r_m$ represent erroneous triangles (fractional packaging) then, $\sum_i r_i \leq OPT$
2. If V represents a clustering then it is assumed that for some $\delta \leq 1/4$, all clusters are δ – *clean*. There exists a fractional packing $r_i, T_{i=1}^m$ such that this clustering has at most $4 \sum_i r_i$ or $4m_{OPT}$ mistakes
3. The mistakes made in the given portion of the graph are charged of to erroneous triangles

B) If OPT represents a a nearly optimal clustering then clean, non-singleton clusters must fulfil following condition:

1. Each non-singleton cluster is δ – *clean*
2. $m_{OPT'} \leq (\frac{9}{\delta^2})m_{OPT}$

C) The final clustering obtained, consists of the data points of the entire graph having only clean and singleton clusters such that:

1. Mistakes which have endpoints lying in the singleton clusters are bound by OPT'
 2. Mistakes with endpoints in clean clusters are bound by $4 \sum_i r_i$
-

- A polynomial time approximation scheme (PTAS) for maximizing agreements: The idea behind this approach is to maximize the positive edges (denoted by $\delta^+(v_1, v_2)$) inside the clusters and the negative edges (denoted by $\delta^-(v_1, v_2)$) between the clusters. The goal is to obtain a clustering having at least $OPT - \epsilon(n^2)$ agreements, (where, $OPT(\epsilon)$ is an optimal clustering, such that $OPT \geq n(n - 1)/4$ and has non singleton clusters having size greater than ϵn). It uses PTAS for maximizing agreements. Two cases completely describe this approach.
- Random noise: For n items, there is considered to be some true clustering ‘ OPT ’. This clustering is not perfect, as the function $f(A, B)$, defined to label the edges, contains some random error. This can be interpreted to mean that there is a constant noise rate of $\nu < \frac{1}{2}$ caused as a result of the edges being mislabeled independently with the probability ‘ OPT ’. This is termed a “problem of learning with random noise” [18]. The analysis is similar to “generic transformation of Kearns” [19], and is closer to the “analysis of graph Partitioning” [20]. It nearly matches a special case of “planted

partition problem” [21]. The noise analysis as given individually by [22], and [16] is presented as:

One-sided noise: Each true ‘+’ edge is flipped to ‘-’ with a probability p . If different clusters of ‘OPT’ are considered, and ‘a’ and ‘b’ are placed in them (not in the same cluster), then noise z can be represented as:

$$|Z^+(a) \cap Z^+(b)| = 0 \quad (1.1)$$

If a and b are members of the same cluster, then every other member node of the cluster has a $(1 - p)^2$ probability of being a common node (neighbor) to both of them. If the cluster under consideration is larger than $Z^+(a)$ then $Z^-(b)$ will have a high probability of having a non-empty intersection. If an arbitrary node v is selected, then the cluster is represented as:

$$C_v = \{a : |Z^+(a) \cap Z^+(b)| > 0\} \quad (1.2)$$

This process is repeated on $V - C_v$ vertices. There is a high probability that for all the nodes placed in ‘OPT’ clusters, having sizes equivalent to $\omega(\log n)$, the rest of the nodes are members of the clusters of a very small size. The overall edge mistakes remain $O(n)$.

Two-sided noise: In this case, the symmetric difference between $Z^+(a)$ and $Z^-(b)$ is considered. If a and b are members of the same cluster of OPT, then every node ‘t’ ($w \notin a, b$) has a $2p(1 - p)$ probability of belonging to this symmetric difference. If a and b belong to same clusters, then all nodes w in $OPT(a) \cup OPT(b)$ have the probability $(1 - p)^2 + p^2 = 1 - p(1 - p)$ of belonging to the symmetric difference. Since the value of the constant $2p(1 - p)$ is less than $\frac{1}{2}$, we can say that a and b will

belong to different clusters, given that:

$$|OPT(a) \cup OPT(b)| = \omega(\sqrt{n \log n}) \quad (1.3)$$

Also, $Z^+(p)$ can be used to sort the vertices approximately according to their cluster sizes. These two facts can be combined to produce correct large clusters of all the vertices. The remaining vertices are placed in the clusters containing only one element. This results in a total of $O(n^{\frac{3}{2}})$ edge mistakes. Various functional imperfections resulting into edge errors have been depicted in Figure 1.3.

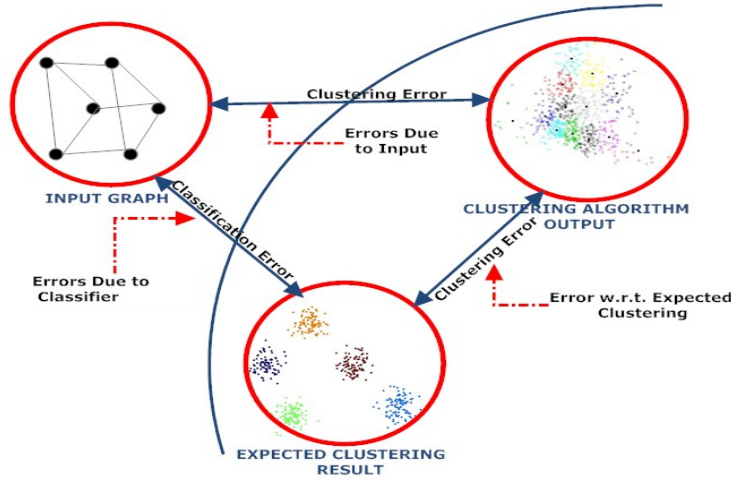


Figure 1.3: Edge errors as a result of functional imperfections

- Linear programming based solutions: A major limitation of the two algorithms discussed above is that they assume that the input graph is a complete graph. This may not be true for many real-world problems. When the problems of a graph that is not complete are generalized, bad triangles are no longer a good lower bound of ‘OPT’. In this scenario, a combinatorial algorithm can be used that uses bad cycles (having exactly one ‘-’ edge) to bind the approximation factor. Some of the researchers have formulated the problem as that of linear programming, and have used the region-growing

techniques that bind them. Various approaches using this methodology are discussed below:

- Correlation clustering in general weighted graphs: [23] have proposed a linear programming formulation of this problem in order to design an approximation algorithm. The flow of the algorithm begins with solving the linear program, then goes on to the interpretation of the fractional values as distance between the vertices. The closer vertices are deemed as similar, and the farther vertices are labeled dissimilar. The last part of the algorithm explores the region-growing technique to perform grouping of the close vertices, and thus round the fractional variables.
- Correlation clustering using random pivot: This approach also focuses on minimizing disagreements and is given by [24]. The disagreements are regarded as bad triplets (having two (+) and one (-) relations), and the clustering instance with no bad triplets is considered to be a consistent clustering. The authors have proposed an algorithm, namely KWIKCLUSTERING, which begins by taking (V, E^+, E^-) as an instance of correlation clustering. A random pivot $i \in V$ is selected. This is the core step of this algorithm and determines the cost of algorithm. If a vertex k is chosen as a pivot out of three vertices (i, j, k) , such that all three of them are in the same recursive call, then the pair i, j will incur a unit cost. After selecting the pivot, it is assigned to a cluster C . Now, $\forall j \in V, j \neq i$ it is checked whether $(i, j) \in E^+$. If they do then j is also added to the cluster C . Another vertex set V' is defined, such that if $(i, j) \in E^-$ then j is added to V' . V' induces a subgraph G' . Let there be a set T that has various instances (t) of bad triplets. For each $t = (i, j, k) \in T$, A_t is defined as an event depicting all three points under consideration as a part of the same recursive call when the first among them was selected as the pivot. The probability of A_t is p_i . The KWIKCLUSTERING process is represented in Figure 1.4.

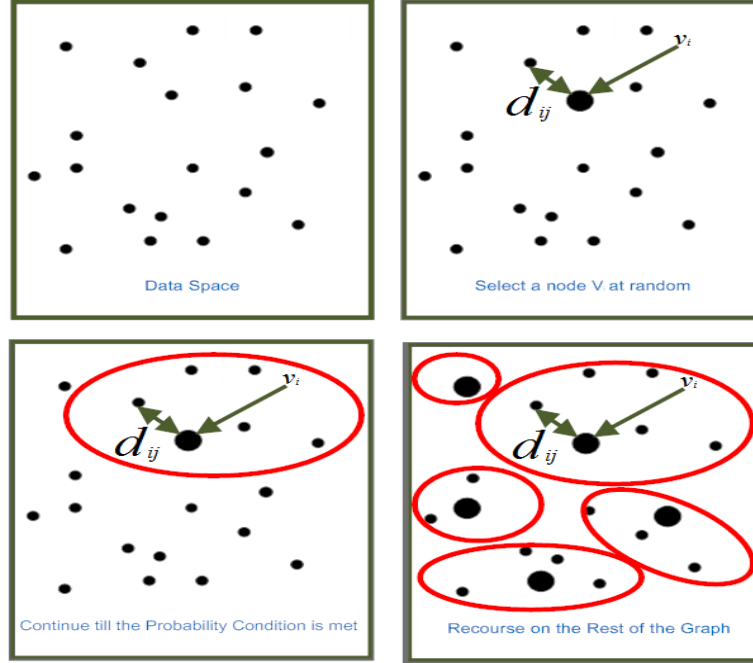


Figure 1.4: Process of pivoting in KWIKCLUSTERING algorithm

- Correlation clustering using qualitative information: The authors [25] considered the problem of correlation clustering and answered many questions left unanswered by [16]. They have provided an overview of clustering obtained by using qualitative information. They have considered minimization in complete graphs using factor 4 approximation, and proved that correlation clustering is APX. For general graphs, they have proved a factor $O(\log n)$ approximation. For maximization of agreements, they have given 0.7664 approximation for general graphs. This proves that the problem is APX hard, and hence it is unlikely that it is a PTAS problem. The authors have not used the standard distance measures. Rather, they have qualitative information, given by a judge, in the form of labelling each pair of elements, in a matrix, as either similar or dissimilar. There is no quantitative information about the distance between a pair of points. The clusters are formulated by combining the similar objects and separating the dissimilar objects to the maximum extent possible. If all the edges of a graph are correctly clustered, then the problem can be easily solved by putting the connected

components (containing similar pairs) of the graph into a single cluster. When the judge makes mistakes, the primary concern is to find a clustering that has the least deviation from the judge's verdict.

- Correlation clustering methodologies with specific constraints: These methodologies are discussed as followed:
 - Correlation clustering with fixed clusters: This approach by [26] focuses on the situation where a small number 'k' is specified as number of clusters. They further prove that for every k, there is a PTAS for the approaches of both maximizing agreements and minimizing disagreements. This problem becomes NP hard for every k greater than or equal to 2. They focus on minimizing disagreements, as maximizing agreements work on the same lines as property tester for MAX k-cut [27]. The authors point out that the qualitative information required for correlation clustering is not provided, it is assumed as a prerequisite for most of the correlation clustering techniques. They consider the maximum number of clusters that can be stipulated from a particular dataspace to be k. They make variants of the MAXAGREE and MINDISAGREE algorithms and name them as MAXAGREE[k] and MINDISAGREE[k]. They also point out that if the number of clusters is not specified, then the original solutions will produce the promised results. They are trying to convert the number of constraints into an external constraint. The existing algorithms are also not easy to modify in context of taking k clusters. This means that the k-clustering variants are difficult to deal with in terms of finding an adequate solution, as a different approach needs to be followed. This approach assumes an input of the graph having every edge labeled as '+' or '-'. The general variants aim at capturing special instances where the classifier responsible for generating fully labeled graphs is not available. The

input in this case is an arbitrary graph G having $\pm$ labels on its edges.

- Correlation clustering with noisy input: A semi-random noisy model was given by [28]. The authors note that the maximization and minimization goals are not equivalent. Minimization goals are harder to achieve, that is given; but the question is, how hard is it to achieve this goal. The algorithm whose results are best known, according to the authors, has an approximation ratio of $O(\log n)$ [23]. They concentrate on the minimization version with complete information, and make an assumption that the input comes from a noisy model.
- Correlation clustering using partial information: Another variant for solving the problem of correlation clustering binds the input by considering partial information. This approach has been given by [29], by considering minimizing disagreements in general graphs having arbitrary weights. They give an $O(\log n)$ approximation algorithm (for general graphs), and for graphs excluding the complete bipartite graphs, they deduce an $O(r^3)$ approximation algorithm. The $O(\log n)$ approximation is based on linear programming, rounding and region-growing techniques [30][31]. The authors have proved that their rounding technique gives a good approximation and have used the ideas developed by [32]. They have then extended their results, by using lemma of [33] to an $O(r^3)$ approximation for $K_{r,r}$ minor free graphs [32]. They also prove that the gap between linear programming is $\omega(\log n)$; this means that the bounds given by them are tight for any algorithm that is based on rounding of this linear program. The hardness of this algorithm comes out to be as hard as the APX-hard minimum multicut [34]. For the given problem, the best-known approximation is $O(\log k)$ for the parameter k [7]. Results similar to these have been independently obtained by [24] and [23]. They have given $O(\log n)$ approximations for minimization of disagreements.

Table 1.1: Fundamental results of various correlation clustering approaches

Correlation clustering approach	Approach scheme	Approximation constant	Runtime	Noise handling	Underlying Technique
PTAS for maximizing agreements[16]	PTAS	$\approx 20,000$	$\Omega(n^2)$	✓	-The problem of general partitioning
CFA for minimizing agreements [16]	Factor 3 approximation	3	$(\frac{1}{\epsilon})R + 1$	✓	-Maximizing agreements -Partitioning into triangles -Multiway cut. -Region growing
CC in general weighed graphs [23]	$O(\log n)$ approximation	$4\log(n)$	LP	×	-Problem of minimizing disagreements -Weighed multicast in general graphs. -MINAGREE and MINDIS-
CC with qualitative information [25]	Factor 4 approximation	4	LP	×	AGREE for general weighed graphs.
Aggregating inconsistent information ranking and clustering [24]	Optimization of CC and consensus clustering	3	$O(m)^*$	×	-Problem of rank aggregation -Problem of consensus clustering.
Aggregating inconsistent information ranking and clustering [24]	Optimizing rank correlation	2.5	LP	×	-Problem of feedback arc set -Problem of ensemble clustering.
True cost of error minimization [35]	pseudometric clustering	< 3	$O(n) + \text{cost}(\text{OPT})$	✓	-Error minimization problem.
CC with fixed clusters [26]	PTAS	≥ 2	$O(\sqrt{\text{Log}n})$	×	-MINDISAGREE and MAXA-
CC with noisy input [28]	$O(\log n)$ approximation	-	LP	✓	GREE of general graphs. -Maximum bisection -Maximum Clique.
CC with partial information [29]	$-O(\log n)$ for general graphs $-O(r^3)$ for complete bipartite graphs	-	LP	×	-Region growing -Minimum multicast -Rounding.

Table 1.1 enlists fundamental results of various correlation clustering approaches. This table takes into consideration criteria such as approximation schemes, approximation constants, runtime, noise handling, and underlying techniques on which various correlation clustering techniques are based. It helps to summarize the finer points of comparison between the approaches discussed above. A detailed comparison of these

approaches is given in Appendix A.

1.4 Challenges of dealing with high-dimensional data

Most of the correlation clustering approaches are applied on 2-dimensional data. They may not work well when the number of dimensions grows to, say, 20. Many applications, such as text documents, or DNA micro array data, may need to handle tens of thousands of dimensions of data. Cluster formulation in high-dimensional data is a problematic task, as there are a high number of irrelevant features of the data objects. The problem further complicates when false correlations emerge, not only among the main features, but also among the subsets of these features. The biggest challenge arises while determining the relevancy of features according to a given cluster, in case feature selection is being used to form clusters. Additionally, different correlations among attributes may be relevant for different clusters. This phenomenon is known as feature relevance or local feature correlation. The most common way of overcoming these challenges is to perform dimensionality reduction, and then apply data mining techniques such as clustering on the reduced datasets. There are various approaches for dimensionality reduction, such as Principal Component Analysis (PCA), that might be deployed. These methods map the original high-dimensional data space to a lower-dimensional data space, so as to form better clusters. However, approaches such as these, are not suitable for all clustering problems. They are global in nature, which means that they compute an entire data space as a single subspace, while in local feature correlation, multiple subspaces need to be evaluated as there might be different subspace for each cluster. Hence, general aim of clustering algorithms designed for high-dimensional data is to find clusters in arbitrarily oriented subspaces of the original data space. The search space of all possible subspaces housing clusters is restricted, but still exists in $O(2^d)$ [36].

1.4.1 Curse of dimensionality

‘Curse of dimensionality’ has been defined as a malediction that has plagued scientists from earliest days [37]. Two common manifestations of this curse are:

- In high-dimensional spaces, the pair of points are equally close to one another.
- Any two vectors taken at random are almost orthogonal.

In terms of clustering, the problems faced because of the curse of dimensionality are listed as follows [38]:

1. With an increase in the number of dimensions, the probability of more values occurring in the data space also increases. It adversely affects the enumeration approach, as tabularization and visualization of numbers becomes difficult. As the number of variables keeps increasing, finding coherent relationships among the data points becomes almost impossible [39]. When a dataset is clustered, data members of the set are generated by using many mathematical functions. A clustering model identifies the relationships and dependencies present in various functions. The increase in dimensionality makes functions more complex as the number of attributes contributing to the relationships also increases.

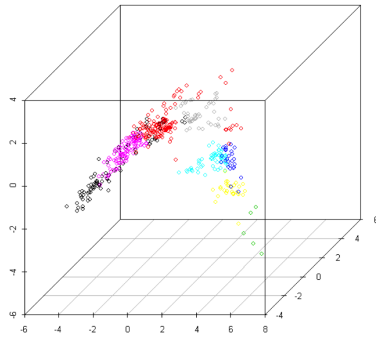


Figure 1.5: Representation of clustered dataset in 3-D space

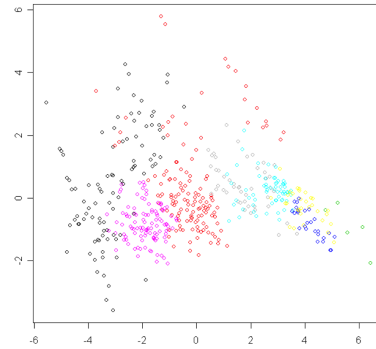


Figure 1.6: Projection on sub-space (x,y)

2. The basic manifestation of the ‘curse of dimensionality’ stands true for the problem of clustering as well. The distinction among the points across the spectrum (nearest to farthest) deduced by using the distance measures becomes quite poor. This results in overlapping of different clusters. The relative distance of the points across the spectrum (nearest and farthest), in a high-dimensional space, converges to zero as the dimensionality ‘d’ increases. It can be represented as [40]:

$$\lim_{d \rightarrow \infty} \frac{dist_{max} - dist_{min}}{dist_{min}} \rightarrow 0 \quad (1.4)$$

The representation of data clusters in 3-D spaces and its projection in 2-D subspaces are depicted in Figure 1.5 and Figure 1.6, respectively. These images clearly depict the curse of dimensionality as the clusters formulated are not clean and distinct.

3. In a particular data analysis process, data accumulation is in abundance, and the single entities are defined by a number of related attributes. In doing so, there might be many irrelevant attributes. In fact, irrelevant attributes are expected to occur. This is because of the fact that relevancy is relative. It is different for different groups of objects present in the same dataset. This is the local feature relevance problem and cannot be resolved by global feature reduction methods. This means that if there is no global noise, but irrelevancy is causing noisy attributes in some objects, then the cluster formulation will not be clean and distinct. This issue causes the problem of subspace hierarchies in high-dimensional spaces. This problem states that the lower-dimensional clusters are embedded in the ones in higher dimensions. There is a need to generate information on hierarchical relationships among various subspaces of clusters [41].

A solution to the ‘curse of dimensionality’ is seen in the fact that the embedding dimensionality of a dataset is greater than the intrinsic dimensionality. This observation is limited

to the spatial queries [42] [43] [44] [45]. This solution is not effective when applied to the above-mentioned clustering problems. Finding correct subspaces in order to define an appropriate group of objects is a difficult problem. The cluster objects are scattered in subspaces that are not only arbitrarily oriented, but are also affine. Certain methods are used to provide better visualization of the data objects, one of them is graph theory.

In this thesis, the focus is on understanding the concept of the use of correlations in data clustering, and using this understanding to propose novel techniques for finding meaningful relationships in data objects. The scope of this research extends to exploring different aspects of correlation analysis, and proposing approaches that are useful for producing better-quality data clusters. These approaches, are also tested on large and high-dimensional datasets, to make sure that they are scalable. Finally, these approaches are merged with recommendation algorithms to develop a recommender system model, that uses the power of correlation to make better recommendations.

1.5 Thesis Organization

The organization of the thesis is as follows:-

Chapter 1: Introduction

This chapter introduces the concept of data clustering, describing it as a computational problem. It touches on subjects such as clustering tendency, and cluster validation to explain how the problem has been defined in the literature. Then, it moves on to describe the significance of correlations in data, laying a foundation for the introduction of the concept of correlation clustering. The problem of correlation clustering has been stated, and various solutions to this problem are also enlisted. It also discusses the challenges faced while dealing with high-

dimensional data, and the curse of dimensionality. In a nutshell, chapter explains the basic concepts of data clustering, and shows how its marriage with the concept of correlations gives rise to the problem of correlation clustering.

Chapter 2: Literature Review

This chapter contains a detailed review of the literature on clustering large and high-dimensional datasets, with special emphasis on correlation clustering. It discusses the increasing use of correlation analysis in finding undiscoverable patterns in data. There is a list of case studies, which show that finding correlations between data objects is the best way to deal with the problem of ever increasing data. A segment on related research discusses the existing surveys on this topic. A significant portion of this chapter consists of an extensive list of clustering approaches for big data mining. There is an in-depth discussion on hierarchical clustering, the algorithm that has been used in both the proposed approaches. It also touches other approaches, that are being used frequently by researchers and developers while mining big data. However, covering only the big data clustering approaches is not enough, there is a detailed section on clustering high-dimensional datasets as well. The discussed approaches have been used to develop many important applications, some of which have been listed, along with challenges and latest trends in clustering large scale, and high-dimensional data. Finally, it also contains a few comparative studies highlighting the advantages and drawbacks of the discussed approaches.

Chapter 3: Problem Formulation

This chapter lists the research gaps that were identified initially, while proposing the research objectives. These gaps identify the problems with the existing data clustering algorithms. They point out the fact that there is a need for working on developing approaches, that are based on finding correlations among significant data points, and applying them for developing better applications. Based on these gaps, there are four research objectives, that

have been addressed in different chapters of this thesis. The objectives are followed by the methodology for achieving the objective. Finally, a mapping of the objectives with publications is given.

Chapter 4: Preliminaries

This chapter gives a background of the proposed approaches. There are many terms and concepts that have to be defined to understand the phases of the proposed frameworks. It enlists concepts like Correlation and Dissimilarity matrix, Fowlkes-Mallow index, Agglomerative Coefficient, PCA, Local covariance matrix, Laplacian transform, etc. All these concepts play an important part in the preceding chapters, and have been described at one place for ease of understanding.

Chapter 5: Proposed Correlation Clustering Approaches

In this chapter, the proposed correlation clustering approaches have been introduced. The first approach is based on the General (rank based) correlation coefficient (G). There is a detailed description of the need for G, and its derivation. G is described in the form of an algorithm, followed by its advantages as a predictive metric. The next part of the chapter describes GBAC (G based Agglomerative Clustering) framework. The phases of this approach have been described in detail and an algorithm has also been added to describe the steps of implementation. The last part of the chapter discusses the framework of the other proposed approach - LGBACC (Local Graph based Correlation Clustering). The phases of this framework have also been described in detail, followed by an algorithm. This chapter explains the proposed approaches, using the preliminaries defined in the previous chapter.

Chapter 6: Experiments and Implementation details

This chapter has the details of the experiments performed on real and synthetic data sets, to validate the proposed approaches. The first phase of experimentation is performed on testing G, and comparing it with the existing correlation clustering metrics. It contains detailed

graphs and results to validate G. The next phase of experimentation involves testing the proposed approaches (GBAC and LGBACC) on real datasets. Standard parameters have been used to evaluate the quality of the clusters obtained by applying these approaches. The next step is to check the scalability of the proposed approaches by running them on very large datasets (from 260 GB to 1.01 TB), using parallel and distributed computing (Spark and R) platform. This chapter shows that the proposed approaches are suitable for tackling large and high-dimensional datasets. The last part of the chapter presents a comparative analysis of the proposed approaches, stating their advantages and disadvantages, along with a short comparison to other correlation clustering approaches.

Chapter 7: Recommendation system model

This chapter proposes a Recommendation system model, developed using the approaches proposed, and validated in the previous chapters. The correlated data points obtained by applying LGBACC or GBAC, are used as similarity measures in the recommendation algorithms. The model is explained by introducing some basic concepts of the model, followed by an explanation of its phases. The last part of the chapter contains details of the implementation of the model on a large, real life data set, and a comparative study of the proposed approach with an existing recommendation algorithm.

Chapter 8: Conclusion and Future Scope

This chapter concludes the thesis by providing a brief overview of the proposed framework, and providing an insight into the future scope of work. Correlation clustering is an evolving field, and this chapter brings to light the huge amount of scope for developing approaches, and applications in this domain. It highlights the contributions made by our work and enlists the points that need to be worked on in the future.

Chapter 2

Literature Review

This chapter contains a detailed review of the literature present on clustering high-dimensional and big datasets. Special emphasis is given to explaining the importance of correlations in data, along with a survey of the approaches on correlation clustering. A major chunk of this survey includes listing of techniques used to cluster large scale data. This section covers the existing surveys in this area and clustering approaches for big data mining. The next major section covers a detailed literature on clustering high-dimensional datasets. As data clustering is an evolving field, a list of applications of large-scale and high-dimensional data clustering is also given, including a brief summary on the recommendation systems. Finally, a list of latest trends in this area are surveyed and listed.

2.1 Correlations in data

When working with huge amounts of data, establishing correlations among data points, brings out a clearer picture of the nature of the data at hand. The weak and strong correlations in data values help us to capture the present and the future. There have been many successful instances where correlations in data points have been used for understanding available but

seemingly irrelevant data and carving out results that have been amazingly accurate.

Amazon is a pioneer in the field of big data analysis. Initially, it collected information regarding the books sold on its website and used it to make recommendations to specific customers. This was done by finding similarities in the customer database, known as user-to-user collaborative filtering [46]. Later, it started finding correlations among the products themselves. This association technique came to be known as item-to-item collaborative filtering [47]. The mathematical determination of the data (finding correlations) in advance increased the efficiency of the system many-fold and cut across product categories [48]. Walmart, one of the largest retailers in the world, had the data analyzed that it had collected over the years, and discovered interesting correlations that helped it develop important business strategies [49]. In 2011, FICO (Fair Isaac Corporation) developed the medical adherence score to determine how often people consume their prescribed medication. Correlations were found in variables that may sound irrelevant, such as marital status, how long have they stayed at an address, if they own a car, etc. Yet these relationships helped them predict which patients are most likely to forget to take their medication, and they can therefore send them reminders [50]. If properly analyzed, data can reveal extraordinary insights into the medical condition of a patient. The predictive analysis algorithms, used on the data collected in intensive care units, regarding premature babies have been able to predict the onset of infections in advance. The data shows correlation and not causality for detection and prevention of many deadly diseases [51]. Google used its enormous amount of data to predict the spread of H1N1 virus, the winter flu, by looking for a correlation between the frequency of a certain query, and the spread of flu over a given time and space. It used frequently searched terms as a proxy for flu and ran 450 million varied models, and found 45 search queries that, when used in a mathematical model, had strong correlation between their predictions and official figures [52]. In the year 2000, the Sloan digital sky survey began collecting astronomical data. A

telescope was located in New Mexico. In the initial weeks, it gathered more astronomical data than had ever been gathered before. By 2010, the data in the archives of the survey were huge, reaching almost 140 terabytes. But its successor, installed in 2016, acquires this quantity of data every five days. Such huge amounts of data can only make sense by using correlational non-causal analysis [53]. In the big-data age, where the datasets are too big and the area under consideration is too complex, the hypotheses are no longer driven by trial and error. Earlier, they were based on developing abstract ideas concerning a phenomenon and collecting data to verify those theories. This was usually done by selecting a proxy to study the relationship between two variables and verifying how good the proxy was by using correlation analysis. Now, sophisticated computational analysis helps to identify the most optimum proxy for the given problem [54]. This means that instead of a hypothesis-driven approach, we use a data-driven approach.

Correlation measures have been used by many researchers to analyze data, taken from different kinds of datasets. A lot of work in this area has been done on fuzzy datasets and their variants. [55] expressed a need for a measure of correlation between two fuzzy membership functions. They focused on studying properties possessed by a correlation measure. Then they defined the measure of correlation based on these properties. [56] used Spearman's rank correlation coefficient to find correlation between two fuzzy sets. The members of the sets have to be ranked according to fuzzy membership values. They proposed a membership value-based fuzzy correlation measure. [57] worked on a correlation coefficient of random numbers. They developed a methodology for calculating a correlation coefficient whose value is a fuzzy number. [58] strictly focused on giving an exact solution of the fuzzy correlation coefficient, without the aid of programming. [59] used an approach based on adaptive neuro fuzzy inference system (ANFIS) for designing SoS reliability that outperforms the fuzzy inference system when its performance is compared with the fuzzy inference system

on the same datasets. [60] focused on developing a method for calculating a correlation coefficient of intuitionistic fuzzy sets by deploying mathematical statistics. They proposed a formula that gives not only strength, but also polarity of the relationship between the datasets. They also extended their research to interval-valued intuitionistic fuzzy sets, i.e., the values of these sets lie in the interval $[-1, 1]$. [61] defined a correlation coefficient between two intuitionistic fuzzy sets. They assumed that an ensemble of ordinary fuzzy sets can be interpreted as an intuitionistic fuzzy set. [62] have worked on correlation coefficients of hesitant fuzzy sets and their applications to clustering analysis. The hesitant fuzzy sets allocate degree of membership to an element in a set. This means that an element can be represented as a member of a set with several possible values. An element can be a member of more than one set. More recently, [63] worked on hesitant fuzzy linguistic term sets (HFLTS). These sets represent hesitant qualitative information in decision-making. The author proposed HFLTS based on traditional correlation coefficients of fuzzy sets, intuitionistic fuzzy sets, and hesitant fuzzy sets. Given that HFLTS have different weights, weighted correlation coefficients and ordered weighted correlation coefficients have also been investigated. A fuzzy-ruled-based system is proposed for estimation of efforts in selection of the components for developing an application based on component based software engineering [64]. Distance correlation has also been applied to random variables by [65]. They proved that distance correlation can be implemented by an $O(n \log n)$ algorithm that is comparable with other computationally efficient algorithms. [66] examined mortality and lifestyle factors that run in families. They used distance correlation to measure pairwise differences between related and random people. [67] have also worked in the area of life sciences using distance correlation. They dealt with very high-dimensional data, and developed an independent screening procedure based on distance correlation. [68] emphasized the uniqueness of distance correlation. Their work has been extended by [69] to general metric space. A comparative analysis of the correlation

Table 2.1: Comparative analysis of correlation measures used on different datasets

Proposed work	Type of Dataset	Correlation Measures	Application Domain	Application Examples
Chaudhri et al. [56]	Fuzzy	Membership-value based fuzzy correlation measure	Random variables	Image processing
Hung et al. [60]	Intuitionistic and interval-valued intuitionistic fuzzy	Correlation coefficient of intuitionistic fuzzy sets	Mathematical statistics	— — —
Liu et al. [57]	Fuzzy	Fuzzy correlation coefficient	Statistical analysis, fuzzy environment	Correlation between technology and management in 15 machinery firms in Taiwan.
Mitchell et al. [61]	Intuitionistic fuzzy and normal fuzzy	Intuitively satisfying correlation coefficient	Fuzzy environment	— — —
Hong et al. [58]	L-R fuzzy numbers	Fuzzy correlation coefficient	T_W based fuzzy arithmetic operations	— — —
Kong et al. [66]	Real data from subpopulation of the Beaver Dam Eye Study	Distance correlations	Mortality and lifestyle factors in families and random people	Relationships between multiple clusters of variables with real valued attributes
Li et al. [67]	Very high dimensional data	Distance correlation	Diverse scientific fields	Screening features in very high dimensional data
Chen et al. [62]	Hesitant fuzzy sets, interval valued hesitant fuzzy sets	Correlation coefficients of hesitant fuzzy sets	Cluster analysis of correlation coefficients of hesitant fuzzy sets	Software evaluation, classification and assessment of business risk failure
Liao et al. [63]	Hesitant fuzzy linguistic term	Correlation coefficient, weighted correlation coefficients, ordered weighted correlation coefficients	Qualitative decision making	Traditional Chinese medical diagnosis
Huo et al. [65]	Synthetic datasets, Random variables	Distance correlation	Applications where statistical dependence needs to be calculated	Feature screening in ultra high dimensional data analysis

measures used by various researchers on different datasets is given in Table 2.1.

2.2 Clustering large-scale datasets

With the evolution of technology, the volume of data collected for any and all applications is multiplying. The definition of large data is very subjective as there is no specific data size above which data can be classified as large. The parameters for such classification are also not set in stone. Large data sets can be defined in terms of volume, processing time, pattern approximation, etc., depending on the application and evaluation criterion. As the volume of incoming data keeps increasing, the challenges of handling this data are also increasing exponentially. The increase in complexity of data mainly results in two major challenges:

1. Developing approaches to handle massive data sets
2. Tackling the problem of high dimensionality in data sets

2.2.1 Related research

Data clustering is an evolving field, many researchers have contributed to literature in this area. In this section, we present a brief summary of the contributions made by other researchers in this area. [70] is one of the earlier contributions in survey of clustering algorithms. The authors have surveyed clustering algorithms used in different domains and their applications on benchmark datasets and computational problems. [71] have talked about applications, challenges, techniques and technologies on big data. They have tried to present a list of technologies used these days to deal with big data. Their focus is on clustering techniques based on MapReduce and parallel classification using MapReduce. [72] have given a review on rise of big data on cloud computing, and have addressed open research issues. [73] have used taxonomy and empirical analysis to survey clustering algorithms in big data. They have presented a survey of different categories of existing clustering algorithms (partitioning-based, model-based, grid-based, hierarchical-based and density-based). The authors have given a comparison between these five categories of data clustering. Their goal is to find the best-performing algorithm for big data clustering. In [74], the authors have presented the most popular algorithms in literature and a comparison between them. The authors have also presented ideas to build a clustering technique based on fuzzy clustering methods. [75] have surveyed big data from data processing and usage perspective. This research barely touches data clustering, but presents a whole new perspective to surveying big data. [76] focus only on the clustering algorithms on density-based data-stream clustering. Clustering of data streams is emerging as an important field of research. [77] is a survey of clustering techniques for big data analysis. The authors have used comprehensive compari-

son to find the most appropriate big data clustering algorithm. [78] define big data and list the challenges presented by data of such huge magnitude. They have reviewed both single and multiple machine clustering techniques. Some of the work corresponds to clustering algorithms dealing with some specific domains, such as machine learning, ontology, data mining, information extraction, text clustering, pattern recognition, bio-informatics, mobile ad-hoc networks, etc. [79], [80], [81], and [82]. [83] have given a survey on the need for designing sequential, parallel and distributed association data mining rules explaining their advantages and disadvantages. The review studied the performance of these algorithms in terms of response time, scalability with minimum overheads, and improved security and privacy concerns.

Table 2.2 presents a comparison of existing surveys on clustering large-scale and high-dimensional data.

2.2.2 Clustering approaches for big data mining

Once the data-mining problem is stated clearly and it is clear that data clustering is the best machine-learning approach to solve it, then the next step is to select the clustering algorithm that will provide an optimal solution for the stated problem. Innumerable clustering approaches are available, but our aim is to enlist and compare the approaches best suitable for large data sets. We also enlist some of the classical approaches for data clustering that are not in themselves equipped for dealing with massive data set problems, but that do form the basis for many other hybrid approaches. The clustering approaches for big data have been given in Figure 2.1.

Table 2.2: Comparative study of the existing research/surveys

Research Paper	Domain	Clustering Process	Clustering big data	Preprocessing of data	HD data clustering	Trends in cluster analysis	Applications of clustering	Challenges faced
Xu et al. [70]	Machine learning, statistics, Computer science	✓	✓		✓		✓	
Brank et al. [81]	Ontology evolution						✓	
Boyinbode et al. [79]	Wireless sensor networks						✓	✓
Aggarwal et al. [80]	Text clustering					✓	✓	✓
Chen et al. [71]	Granular computing, cloud computing, bio-inspired computing, quantum computing						✓	✓
Hashem et al. [72]	Cloud computing							✓
Fahad et al. [73]	Machine learning, statistics, Computer science		✓		✓	✓		
Chen et al. [75]	Cloud computing, internet of things			✓	✓		✓	✓
Amini et al. [76]	Data stream		✓		✓	✓		✓
Zerhari et al. [78]	Health, bio-medical, marketing, finance, marketing, transport		✓		✓			✓
Ayed et al. [74]	Machine learning, statistics, Computer science		✓		✓		✓	
Arora et al. [77]	Machine learning, statistics, Computer science		✓		✓			

Hierarchical approach

This approach is best suited for clustering problems involving relatively small data sets. The basic principle involves checking the closeness of data points. The definition of closeness is subjective and can be picked up from many definitions of similarity and distance measures.

The minimal distance can be calculated using the function:

$$F(x_i, x_j) = \min F(x_m, x_L), \text{ given that } 1 \leq m, L \leq N \text{ (initial number of clusters) and } m \neq L.$$

In the beginning, each point forms its own cluster, which is why this type of clustering is also known as agglomerative clustering. Then, these clusters are iteratively merged depending on the chosen definition of closeness. The merger of clusters ends based on a particular criterion defined beforehand. The last iteration gives distinctively defined clusters. This algorithm is of cubic time complexity. The initial step takes $O(n^2)$ time. The following steps take

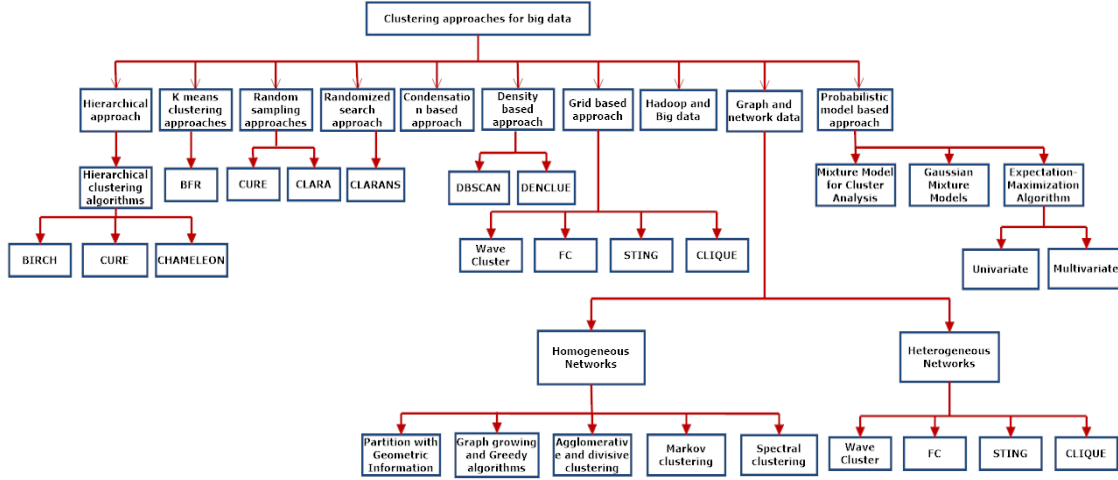


Figure 2.1: Clustering approaches for big data

$(n-1)^2, (n-2)^2$ time. The summation of squares is $O(n^3)$, making it cubic [84]. The result of a hierarchical algorithm is usually described in the form of a dendrogram, where data set is represented by its root, and data object by each leaf node. The distance measure between the objects, as well as the clusters, is represented by the height of the dendrogram. In order to visualize the real hierarchical relationships in data, we can cut the dendrogram at different levels [85]. Based on the distance measure, there are mainly three agglomerative clustering approaches:

1. Single linkage: Checks the closest distance between the two objects in different clusters, also known as the nearest neighbor search [86]. Figure 2.2 shows the single linkage between two clusters. This kind of search focuses more on the close regions and ignores the overall structure of the cluster. It is capable of clustering non-elliptical shaped group of objects and is sensitive to outliers and noise.
2. Complete linkage: Checks the farthest distance measure of two objects to determine the distance between any two clusters [87]. Figure 2.3 represents complete linkage between two clusters. The similarity between the two most dissimilar members is used to indicate the similarity between the two clusters. It merges two clusters to form

one with the smallest diameter. and is nonlocal in behavior. It helps in obtaining compact-shaped clusters and is sensitive to outliers.

3. Average linkage: It is a more complicated agglomerative approach, which considers the mean distance between two objects belonging to different clusters [88]. Figure 2.4 represents average linkage between one element in the first cluster and another element in the second cluster. C_a and C_b represent two clusters. N_a and N_b represent the count of elements in clusters C_a and C_b respectively. This measure is expensive to compute. Recently, a new algorithm, Twister ties [89] has been proposed that uses local sensitivity hashing, combined with a novel data structure called twister, tries to provide approximate clustering for average linkage. This approach only requires linear space. Further, the time complexity is linear in the number of items to be clustered, making it feasible to be applied to larger scale.

Both single and complete linkage can be represented by a recurrence formula [90]:

$$D(C_l, (C_i, C_j)) = \alpha_i D(C_l, C_i) + \alpha_j D(C_l, C_j) + \beta D(C_i, C_j) + \gamma |D(C_l, C_i) - D(C_l, C_j)|$$

Here, $D(**)$ is the distance function, α_i ; α_j ; β and γ are coefficients, that take value according to the scheme used.

For Single Linkage, the conditions are:

$$\alpha_i = \alpha_j = 1/2, \beta = 0 \text{ and } \gamma = -1/2, \text{ hence formula reduces to: } D(C_l, (C_i, C_j)) = \min(D(C_l, C_i), D(C_l, C_j))$$

For Complete Linkage, the conditions are:

$$\alpha_i = \alpha_j = \gamma = 1/2, \beta = 0, \text{ hence the derived formula is: } D(C_l, (C_i, C_j)) = \max(D(C_l, C_i), D(C_l, C_j))$$

This particular approach is not appropriate for handling large-scale data sets as it has the least computational complexity and incurs high costs of implementation, with no mechanism to correct the misclassification of data, if any. Also, the clusters in this case tend to

form spherical shapes and show reversal phenomenon; this tends to distort the hierarchical structure.

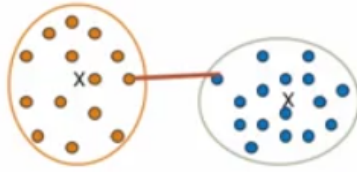


Figure 2.2: Single linkage hierarchical clustering

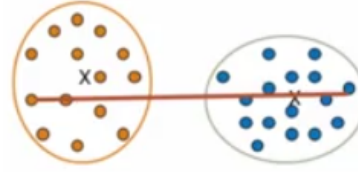


Figure 2.3: Complete linkage hierarchical clustering

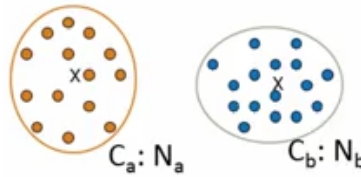


Figure 2.4: Average linkage hierarchical clustering

Hierarchical clustering algorithms to handle large data sets

We discuss the three major algorithms based on hierarchical approach that can be used to cluster large data sets:

1. BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies) [91]: It is a multiple clustering algorithm that uses hierarchical clustering approach. It constructs a clustering feature (CF) tree incrementally. This is a hierarchical data structure for multiphase clustering. It mainly has two phases. In the first phase, an initial in-memory CF tree is built by scanning the database, where data is compressed in a multilevel way, trying to pressure the inherent clustering structure of data. In the next phase, CF tree's leaf nodes are clustered by employing an arbitrary clustering algorithm.
2. CURE (Clustering Using Well Scattered Representatives) [92]: It is used to represent a cluster, using a set of well distributed representative points. The minimum distance

between the representative points chosen is the cluster distance. This means that this technique incorporates both the single and average link methodologies. CURE is used to capture clusters of arbitrary shapes by choosing a scatter plot.

3. CHAMELEON (Graph Partitioning on the KNN graph of the Data) [93]: It uses a dynamic model to measure similarity between data points. If the closeness (proximity) and interconnectivity between any two clusters are high, with respect to the cluster's internal interconnectivity and the items' closeness within clusters, then the two clusters are merged into one. This is a two-phased algorithm that is graph-based. The cluster objects are partitioned into a large count of comparatively small sub-clusters (graphlets) by using graph partitioning algorithm in the first phase. Then, genuine clusters are found by repeatedly combining these sub-clusters, by using agglomerative hierarchical clustering algorithm in the second phase.

K-means clustering approach

K-means algorithm is the point assignment type of approach, that makes an assumption about the initial cluster count; then iterative fusions or divisions give the optimal number of clusters. If criteria such as the average diameter or radius are taken for calculating the distance measure between two clusters, then value grows until the assumed numbers of clusters remain equal or more than the actual count of clusters obtained. For selecting the right value k , values as 1,2,4,8 can be taken, until the two values v and $2v$ are obtained. The decrease in the average radius or diameter among these values is very small. The value of k lies between $v/2$ and v . Now binary search can be used to locate the value of k . Taking total clusterings as $2\log_2 v$, the best match of k can be found in $(\log_2 v)$ clustering operations. After the initial value of k is selected, these points are made centroids of the clusters. The remaining points are then represented as 'p'. These points are then checked using a for loop, which is the heart

of this approach. For each p , the centroid closest to it is found and p is added to the cluster of the respective centroid. The centroids are also adjusted to account for p [94]. This brute force method is quite inefficient as the cost of computation becomes manyfold when this approach is implemented. Even in small-scale clustering problems, the number of partitions becomes exceedingly large [95].

Random sampling approach

In order to deal with dynamic data applications, we have to opt for randomized sampling. Here, appropriate samples are chosen so that important geometrical properties of the data clusters are maintained. Also, lower limit bound of the minimum sample size is given so as to lower the probability that important points are missed in the cluster sample [92]. The examples of such an approach are CURE and CLARA. CURE is used to cluster large data sets. The shape of the clusters requires no initial assumptions. The data distribution can be in the form of rings, S-shapes or strange bends. The clusters are represented by a group of representation points instead of centroids [96]. CLARA is also a variation of this type of scheme, but the difference remains that this approach forms clusters with a medoid [97].

Randomized search approach

This is a very interesting approach in which one can visualize the search being done as equivalent to traversing a graph. The search begins randomly by selecting an arbitrary data item and designating it as the current node. Then, the nearest neighbors of this node are searched in order to find the one with lower cost. Now this becomes the current node and the entire process is repeated. Each node here corresponds to k medoids [98]. CLARANS is the example of such an approach [99].

Condensation-based approach

This approach gives a condensed summary of the original data. The best example of this type of large-scale clustering is BIRCH. It not only has the effective power to deal with massive data sets, but it can also efficiently deal with outliers. This approach introduces a new data structure known as the clustering feature tree [100].

Density-based approach

In this approach, it is assumed that the generation of data objects is done by adopting various probability distributions. Different types of density functions are used in order to derive them. If the distribution is known, then it becomes simple to find clusters in the given data. If prior probability for cluster $c_i, i = 1, \dots, k$, and conditional probability density $p(x|c_i, \theta_i)$, given θ_i being the unknown parameter vector, whole data's mixture probability density can be represented as [101] [102]:

$$p(x|\theta) = \sum_{i=1}^k p(x|c_i, \theta_i)P(C_i) \quad (2.1)$$

Here, $(\theta = (\theta_1, \dots, \theta_k))$ and $\sum_{i=1}^k P(c_i) = 1$

Arbitrary shaped clusters are discovered by this approach to handle noise effectively. It just performs one scan to examine the local region in order to justify density and need-density parameters as termination conditions. Examples for this type of clustering approach are DBSCAN [103] and DENCLUE [104].

Clustering graphs and network data

Most real-world data are interconnected forming gigantic networks or graphs. Networks can be homogenous or heterogenous. Clustering groups objects into a class that can be hard-

like k-means or soft like-EM; complete or partial; balanced or skewed. A network or a graph is represented as $G=(V,E)$, where a set of vertices or nodes is denoted by V and a set of edges or links is denoted by E . If more than one edge exists between the same pair of vertices, it is called Multi-edge graph and if a vertex is connected to itself by an edge, then it is called a loop. If a network neither has self edges nor multi edges, it is called a simple network. There are different ways to represent a network like [105]: Adjacency Matrix($A_{ij} = 1$ if there is an edge between vertices i and j ; 0 otherwise); Directed Graph(if each edge has a direction ($tail \rightarrow head$ i.e. $A_{ij} = 1$ if there is an edge from j to i ; 0 otherwise); Weighted Graph(if a weight w_{ij} usually a real number is associated with each edge v_{ij}). There are popular graph-based approaches like SimRank [106]. It follows a random walk model in order to check similarity between two nodes. Spectral clustering is one of the most popular clustering algorithms that makes no assumption about the shape of the clusters. The process of this approach starts with construction of a similarity graph for all data points [107]. RankClus [108] is another algorithm that generates a simple network in the beginning and then is ranked. Based on this ranking, the clustering is finalized. This is an efficient method of cluster estimation and follows the EM method. NetClus [109] approach captures more semantics with multiple type networks. The foremost step of this algorithm involves generating the target object's initial partitions and inducing first net clusters from the original work. PathSim [110] is based on the concepts of similarity search and meta paths. Meta paths is a meta level description of a path between two objects types. Different meta paths may depict different semantics. It uses a random walk model along with P-Page rank similarity measure.

Grid-based approach

Grid-based clustering explores multi-resolution grid data structure in clustering. The data space is partitioned into a definite number of cells forming a grid structure; then, it finds clusters (dense regions) from the cells in the grid structure. In this approach, the number of cells should be less than the number of data points, for obtaining high efficiency and scalability. It is limited by predefined cell sizes, borders, and density threshold, and is not the right approach to handle high-density data. It can be explained by algorithms such as WaveCluster [111] and Fractal Clustering [112]. Other approaches that are based on this type of approach are STING [113] and CLIQUE [114]. The spatial area is divided into rectangular cells by a Statistical Information Grid Approach (STING) at different levels of resolution, forming a tree structure. CLIQUE (Clustering In QUEst) is a clustering algorithm based on density, as well as on grid subspace. The data space is discretized through grid, where a grid estimates the density by the count of points present in a grid cell. A new approach has been defined for improving the replica selection in data grids using the rough set theory [115]. In this method, an unsupervised learning is changed into supervised learning using Quickreduct algorithm, and optimum rules are obtained by using Rule algorithm.

Probabilistic model-based clustering methods

A probabilistic model gets data from a generative process. It assumes that data is produced by a combination of underlying probabilistic distributions. It attempts to optimize the fit between observed data and any mathematical model using the probabilistic method. In probable model-based clustering, the mathematical representation of each cluster can be done using parametric probability distribution (e.g., Gaussian or Poisson distribution). The clusters will be formulated of the data points that most likely belong to the same distribution. There must also be some parameter estimation, so that the data points will have a maximal

tendency for fitting in the model by a mixture of k component distributions (i. e., K clusters) [116].

Hadoop and big data

Apache hadoop is an open source tool written in Java that facilitates distributed processing of large datasets across clusters of commodity hardwares [117]. MapReduce is the programming model used in Hadoop. Proposed by a Dean and Ghemawat at google [118], this model is composed of two parts, known as mappers and reducers. At the high level, the mappers read the data from HDFS, process it, and generate some intermediate results to the reducers. In the second phase, reducers are used to aggregate intermediate results to generate the final output, which is again written to HDFS. In a typical Hadoop job, it is required to run several mappers and reducers across different nodes in the cluster [119]. Recently, there have been attempts to improve the efficiency of MapReduce in order to facilitate implementation of real-world tasks, such as search engines and machine learning. An example of this is MapReduce-Merge [120]. It adds a merge phase that efficiently merges data already partitioned and sorted by map and reduce modules. This new model can both express relational algebra operations and implement join algorithms. As the volume of data keeps increasing, it becomes harder to cluster it. Many researchers have tried to design efficient parallel clustering algorithms. [121] introduced the concept of peer enterprises, where organizations share their under-utilized resources by participation in a peer-to-peer network, which offers computing power the same as that of a cloud without any extra investment. [122] have proposed a parallel k -means clustering algorithm based on MapReduce. [123] have implemented a parallel Apriori algorithm based on MapReduce. [124] have proposed an improved k -means clustering algorithm to conduct clustering analysis based on medical data employing MapReduce computing framework. [125] have presented k -Medoids clustering algorithm based on the

MapReduce paradigm to be able to perform clustering on large-scale data. [126] have proposed an approach for heavily skewed data. There have also been efforts to develop big data benchmarks in order to efficiently analyze data and derive meaningful results. These benchmarks are designed, keeping in mind the four v's of big data: volume, veracity, velocity, and variety [127]. A few benchmarks use Hadoop as the software stacks. Some of them are: Hi-bench [128], developed to perform offline analytics and real time analytics. The applications that can be associated with it are sort, word-count, tersort, page-rank, bayes classification, nutch indexing, etc. GridMix [129] is used for online services to perform applications like sort and sampling. Performance benchmark [130] is also used for online services with wider range of applications, such as data loading, analyzing URL links, etc. BigBench [131] is another benchmark that is used for offline analytics and real-time analytics. It can perform applications such as search engines, social networks, connected components, e-commerce collaborative, naive bayes, relational database query analysis. Many intelligent systems have evolved as a result of these emerging distributed computing platforms. [132] summarized five hybrid intelligent systems in his book by giving the advantages and disadvantages of these. The five hybrid intelligent systems given are expert systems, fuzzy logic, neural networks, genetic algorithms and case-based reasoning. The features of the two technologies neural networks and expert systems were combined to make the powerful systems to make the integrated system by utilizing the complimentary features of both [133]. There are many applications for such type of a platform, one of which is that the auditors have been trained in strategic systems for using the online computerized systems [134], which can affect the knowledge acquisition of auditors. The strategic systems auditing is an important application in auditing firms for computerized aids. A case study on auditing failure at Satyam Computer services, with the focus on whether it loses reputation after the audit failure, with feedback taken from Indian market, is an example [135]. A major concern for setting up cloud-based

parallel and distributed environment is the issue of security. [136] worked on detecting Distributed Denial of Service (DDoS) attacks by using IP traceback in which threshold value is maintained for identifying the normal traffic from abnormal traffic for detection of DDoS attacks. These type of attacks are very common nowadays on cloud-based distributed and parallel systems. The security of cloud data was provided by using a third-party auditor, where file contents were encrypted at the user level for ensuring the data owner and the client that the data is intact [137]. It also helps in preserving n numbered client server consistency and dynamics of data. The cross layer mechanism for SVC video transmission [138] provided control of different Access Classifications (ACs) for video packets by enabling interaction with Medium Access Control (MAC) layer. A comprehensive comparison of the clustering approaches for big data is given in Table 2.3.

Table 2.3: A comprehensive comparison of the clustering approaches for big data

Approach	Proposed by	Model	Experimental results					Unique Functional Properties	Experimental Highlights	
			Capability of tackling large data sets	Sensitive to outlier/noise	Time Complexity	Space Complexity				
Hierarchical approach										
Single linkage	[86]	Static	No	Yes	$O(n^2 \log n)$	$O(n^2)$	Represented as: $D(C_i, C_j, C_l) = \min(D(C_i, C_j), D(C_i, C_l))$ 1. Is more complex agglomerative approach.	–		
Average linkage	[88]	Static	No	Yes	$O(n^2)$	$O(n^2)$	2. Includes average, median, centroid linkage and wards method	–		
Complete linkage	[87]	Static	No	No	Obvious Algorithm: $O(n^3)$ Priority Queues: $O(n^2 \log n)$ Euclidean Plane: $O(n \log^2 n)$ Approximation Algorithm: $O(n \log n)$ $+ n \log 2(1/\epsilon)$	$O(n^2)$, $O(n)$, $O(n)$	1. Represented as: $D(C_i, C_j, C_l) = \max(D(C_i, C_j), D(C_i, C_l))$. 2. Forms spherical shapes and show reversal phenomenon	–		
Hierarchical clustering algorithms										
BIRCH	[91]	Dynamic	Yes	No	$O(n)$	–	1. The important information is captured effectively while reducing the storage required doing the same. 2. This algorithm successfully eliminates the outliers.	1. It is a multilevel clustering approach. 2. The first phase constructs CF trees in memory. 3. The second phase clusters data.		

Table 2.3: A comprehensive comparison of the clustering approaches for big data

Approach	Proposed by	Model	Experimental results					Experimental Highlights
			Capability of tackling large data sets	Sensitive to outlier/noise	Time Complexity	Space Complexity	Unique Functional Properties	
CURE	[92]	Static	Yes	No	$O(n^2 \log n)$	$O(n)$	1. It does not make any assumptions about the shape of clusters. 2. The distribution can be in the form of rings, S-shapes or strange bends. 3. Clusters are represented by a group of representation points instead of centroid.	1. It is a multilevel clustering approach. 2. The first phase constructs CF trees in memory. 3. The second phase clusters data.
CHAMELEON	[93]	Dynamic	Yes	Yes	$O(n(\log_2 n + m))$	—	1. It is based on nearest neighbor graph. 2. It uses minimal edge cut.	1. It is a graph-based, two-phased algorithm. 2. The objects are partitioned into sub-graphs by using a graph partitioning, and they are then combined using hierarchical clustering.
K-Means approach								
K-Means clustering	[139]	Static	No	Yes	$O(NKd)$	$O(N + K)$	Works best with compact and hyperspherical cluster shapes.	1. Value of number of clusters is assumed. 2. Binary search makes finding the correct value of k easier. 3. The brute force method is very costly.
BFR	[140]	Dynamic	Yes	No	—	—	It makes an assumption about the shape of clusters, that they must be normally distributed about a centroid.	1. The data points are read in chunks. 2. The data in the memory is divided into discard, compressed and retained sets.

Table 2.3: A comprehensive comparison of the clustering approaches for big data

Approach	Proposed by	Model	Experimental results					Experimental Highlights
			Capability of tackling large data sets	Sensitive to outlier/noise	Time Complexity	Space Complexity	Unique Functional Properties	
K means via PCA	[141]	Dynamic	Yes	No	–	–	Principal components are continuous solutions to discrete cluster membership indicators for K-means clustering.	1. Unsupervised dimension reduction is closely related to unsupervised learning. 2. New bounds of K-means clustering are within 0.5 - 1.5% of the optimal values.
ORCLUS	[142]	–	Yes	–	$O(K_0^3 + K_0Nd + K_0^2d^3)$	$O(K_0d^2)$		1. It begins with a set of K_0 seeds that are randomly selected and have full dimensionality. 2. The dimensionality and the number of clusters keep on reducing until the predefined value of clusters is reached.
Random sampling approach								
CURE	[92]	Dynamic	Yes	No	$O(n^2 \log n)$	$O(n)$	The initial clusters are formed using a standard clustering approach. Representative points are selected to merge clusters.	1. It does not make any assumptions about the shape of clusters. 2. The distribution can be in the form of rings, S-shapes or strange bends. 3. Clusters are represented by a group of representation points instead of centroid.
CLARA	[97]	Dynamic	Yes	No	$O(K(40+K)^2 + K(N-K))$	–	Similar to CURE	Forms clusters with a medoid
Randomized search approach								

Table 2.3: A comprehensive comparison of the clustering approaches for big data

Approach	Proposed by	Model	Experimental results					Experimental Highlights
			Capability of tackling large data sets	Sensitive to outlier/noise	Time Complexity	Space Complexity	Unique Functional Properties	
CLARANS	[98]	Dynamic	Yes	Yes	$O(n)$	—	Search can be visualized as being equivalent to traversing a graph.	1. Search Process in the form of a graph, where each node corresponds to a set of k medoids. 2. Search can be visualized.
Condensation-based approach								
BIRCH	[91]	Dynamic	Yes	No	$O(n)$	—	A new data structure clustering feature tree is used.	1. The important information is captured effectively while reducing the storage required doing the same. 2. This algorithm successfully eliminates the outliers.
Density-based approach								
DBSCAN	[103]	Dynamic	Yes	No	$O(N \log N)$	$O(n^2)$	Based on the principle that if an object is a member of a cluster, then its neighborhood density should be reasonably high.	1. A data object creates new data objects by absorbing all objects in its neighborhood. 2. The data structure used is R*-tree structure.
DENCLUE	[111]	Dynamic	Yes	No	$O(N \log N)$	$O(n^2)$	Influences the data objects in the data space corresponding to it.	Generates clusters that are created by taking the maximum value of the density function.
Grid-based approach								

Table 2.3: A comprehensive comparison of the clustering approaches for big data

Approach	Proposed by	Model	Experimental results					Experimental Highlights
			Capability of tackling large data sets	Sensitive to outlier/noise	Time Complexity	Space Complexity	Unique Functional Properties	
WaveCluster	[111]	Dynamic	Yes	No	$O(N)$	—	It maps objects to frequency domain.	Defines a set of units in the given feature space, and assigns data objects to these units.
Fractal Clustering	[112]	Dynamic	Yes	No	$O(N)$	—	There is a predefined condition of cluster's fractal dimension.	It has a predefined condition that fractal dimension of the cluster should be kept relatively stable.
STING	[113]	Dynamic	Yes	No	$O(N \log N)$	—	It works on the principle of forming a statistical grid.	It divides the spatial area into rectangular cells at different levels of resolution, and these cells form a tree structure.
CLIQUE	[114]	Dynamic	Yes	No	$O(N)$ for number of objects, $O(r^2)$ for no. of dimensions	—	Generates clusters that are created by taking the maximum value of the density function.	1. It is based on both density and grid subspace clustering. 2. It defines subspace clusters as dense cells set in arbitrary subspaces. 3. Quality of clusters is dependent on the appropriate selection of the number as well as the width of the partitions and grid cells.
OptGrid	[143]	Dynamic	Yes	No	Between $O(Nd)$ and $O(Nd \log N)$	—		1. Designed to get a prime grid partitioning. 2. Achieved by creating cutting hyperplanes through a set of projections.

Graph-based approach

Table 2.3: A comprehensive comparison of the clustering approaches for big data

Approach	Proposed by	Model	Experimental results					Experimental Highlights
			Capability of tackling large data sets	Sensitive to outlier/noise	Time Complexity	Space Complexity	Unique Functional Properties	
FADE	[144]	Static	Yes	No	$O(e + n \log n)$, where n and e are the numbers of nodes and edges	—	It uses decomposition trees to represent hierarchical clustering of the nodes.	1. It uses edges and multilevel visual abstraction. 2. Decomposition tree provides a systematic way to determine the degree of closeness between nodes without explicitly calculating the distance between each node.
			Yes	No	$O(n^3)$	—	The data points are embedded in low-dimensional space where clusters are more obvious.	1. This type of clustering makes no assumption on the shapes of clusters and can handle intertwined spirals etc. 2. It uses graph Laplacian's eigenvectors.
Ng-Jordon-Weiss(NJW) algorithm	[146]	Static	Yes	No	$O(n^2)$	—	The method is well suited to the problem of appearance-based, on-line topological mapping for mobile robots.	1. Spectral clustering algorithm is applied to the affinity matrix after each row/column is added - which makes it possible to inspect the clusters as new data points are added. 2. only a fraction of the entries in the affinity matrix are computed.

Table 2.3: A comprehensive comparison of the clustering approaches for big data

Approach	Proposed by	Model	Experimental results					Experimental Highlights
			Capability of tackling large data sets	Sensitive to outlier/noise	Time Complexity	Space Complexity	Unique Functional Properties	
Improved min-max cut graph clustering with nonnegative relaxation	[147]	Static	Yes	No	$O(n^2)$	—	With the explicit nonnegative constraint, the proposed solutions are very close to the ideal class indicator matrix and can directly assign clusters to data points.	1. The authors apply additional nonnegative constraint into MinMax Cut graph clustering. 2. The algorithm always converges and significantly outperforms the traditional spectral relaxation approach.
Spectral Embedded Clustering	[148]	Static	Yes	No	—	—	In this framework, a linearity regularization is explicitly added into the objective function of Spectral Clustering methods.	1. The proposed framework can naturally deal with out-of-sample data. 2. A new Laplacian matrix is constructed, from a local regression, of each pattern, and incorporated into the framework to capture both local and global discriminative information for clustering.
Constrained Laplacian Rank Algorithm for Graph-Based Clustering	[149]	Dynamic	Yes	No	—	—	The data graph itself is adjusted as part of the clustering procedure.	1. Two versions of this method are developed, based upon the L1-norm and the L2-norm, giving two new graph-based clustering objectives. 2. Optimization algorithms are used to solve these objectives.

Table 2.3: A comprehensive comparison of the clustering approaches for big data

Approach	Proposed by	Model	Experimental results					Experimental Highlights
			Capability of tackling large data sets	Sensitive to outlier/noise	Time Complexity	Space Complexity	Unique Functional Properties	
Initialization independent clustering with actively self-training method	[150]	Dynamic	Yes	No	—	—	This method is free of initialization, while traditional clustering methods are usually dependent on initialization.	1. It is an actively self-training clustering method. 2. The framework is developed on graph to perform semi-supervised learning, where Bayes error can be efficiently estimated.
Randomized greedy modularity clustering algorithm and the core group graph clustering scheme	[151]	Static	Yes	No	$O(n)$	—	It is a nondeterministic agglomerative hierarchical clustering approach which finds locally optimal solutions.	1. It is an ensemble-learning clustering method which combines the local solutions of several base algorithms to form a good start solution. 2. The similarity of the randomized greedy modularity algorithm with incomplete solvers for the final algorithm is analyzed.
A partitioning-based divisive clustering technique for maximizing the modularity	[152]	Static	Yes	No	$O(V K + E)$ for a K-way clustering of a graph with $ V $ vertices and $ E $ edges.	—	The algorithm pursues a divisive clustering approach and uses established graph partitioning algorithms and techniques to compute recursive bipartitions of the input as well as to refine clusters.	The algorithm accepts a weighted graph $G = (V, E, \omega)$, and returns the number of clusters K^* and the clustering $C^* = \{C_1^*, C_2^*, \dots, C_{K^*}^*\}$. It uses a bisection heuristic to compute a clustering of the given graph. Initially, all the vertices are in a single cluster.

Table 2.3: A comprehensive comparison of the clustering approaches for big data

Approach	Proposed by	Model	Experimental results					Experimental Highlights
			Capability of tackling large data sets	Sensitive to outlier/noise	Time Complexity	Space Complexity	Unique Functional Properties	
Scalable and Accurate Graph Clustering and Community Structure Detection	[153]	Static	Yes	No	$O(n \log n)$.	—	The problem of finding a partition maximizing the modularity of a given graph G , can be reduced to a minimum weighted cut.	—
RankClus	[108]	Dynamic	Yes	No	$O(t_1 E + t_2(K E + k + mK) + mK^2)$, t_1 is iteration of whole algorithm, t_2 is iteration of mixture model, m is distance between each object, $ E $ — is no of links.	—	The problem of generating clusters for a specified type of objects, as well as ranking information for all types of objects based on these clusters in a multi-typed (i.e., heterogeneous) information network is addressed.	1. RankClus can generate more accurate clusters and in a more efficient way than the state-of-the-art, link-based clustering methods. 2. Clustering results with ranks can provide more informative views of data compared with traditional clustering.
NetClus	[109]	Dynamic	Yes	No	$O(kmn)$	$O(mn)$	A loop like an EM framework is started and steps are repeated until the clusters do not change significantly.	1. The algorithm's foremost step is generating the target objects' initial partitions and inducing first net-clusters from original network. 2. Further, NetClus generates informative clusters, presenting good ranking and cluster membership information for each attribute object in each net-cluster.

Table 2.3: A comprehensive comparison of the clustering approaches for big data

Approach	Proposed by	Model	Experimental results					Experimental Highlights
			Capability of tackling large data sets	Sensitive to outlier/noise	Time Complexity	Space Complexity	Unique Functional Properties	
PathSim	[110]	Dynamic	Yes	No	$((KN^2d^2)$, K is no. of iterations, N is total number of objects, and d is average neighbor size.	$O(n^2)$	Based on the principle that similarity search that is defined among the same types of objects in heterogeneous networks.	1. It is an efficient solution that partially materializes short meta paths, and then concatenates them online to compute top-k results. 2. It is able to find peer objects in the network.

2.3 Clustering high-dimensional datasets

Cluster formulation in high-dimensional data is a problematic task, as there are a high number of irrelevant features of the data objects. The problem further complicates when false correlations emerge not only among the main features, but also among the subsets of these features. The biggest challenge arises while determining the relevancy of features according to a given cluster, in case feature selection is being used to form clusters. Additionally, different correlations among attributes may be relevant for different clusters. This phenomenon is known as feature relevance or local feature correlation. The most common way of overcoming these challenges is to perform dimensionality reduction, and then apply data-mining techniques such as clustering on the reduced datasets. There are various approaches for dimensionality reduction, such as Principal Component Analysis (PCA), that might be deployed. These methods map the original high-dimensional data space to a lower-dimensional data space, so as to form better clusters. However, approaches such as these are not suitable for all clustering problems. They are global in nature, which means that they compute an entire data space as a single subspace whereas, in local feature correlation, multiple subspaces need to be evaluated as there might be different subspace for each cluster. Hence the general aim of clustering algorithms designed for high-dimensional data is to find clusters in arbitrarily oriented subspaces of the original data space. The search space of all possible subspaces housing clusters is restricted, but still exists in $O(2^d)$ [36]. Different techniques are used to deal with forming clusters in high dimensional data. Figure 2.5 shows the scope of this section.

- Subspace clustering approaches: Subspace clustering focuses on the integral feature space, by finding clusters in all subspaces. In CLIQUE [154], the entire data space is partitioned into equal-sized units by using a grid, and units containing a specified num-

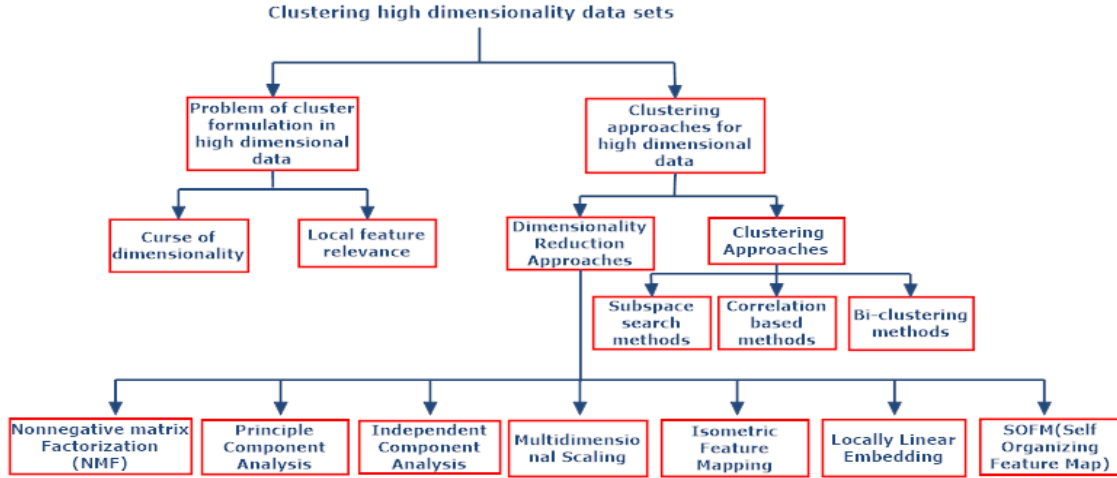


Figure 2.5: Clustering high-dimensional data sets

ber of points are considered to be dense. The maximum set of dense points adjacent to each other is termed as cluster. The dense units satisfy the downward closure property. There is another variant of CLIQUE, known as ENCLUS [155] that instead of dense units, searches for subspaces containing one or more clusters relying on a fixed grid only. MAFIA [156], which adopts the property of adaptive grid with subspace cluster generation, is similar to CLIQUE. In nCluster [157], 1- dimensional grid unit is formed by allowing the overlap of windows of δ length. SUBCLUE [158] searches in the density-based clusters, with the resulting clusters exhibiting the arbitrary shape or size in the corresponding subspaces. It computes the clusters by employing DB-SCAN cluster model [103] applied to the particular subspace, but requires high runtime. DUSC [159] is a dimensionality unbiased cluster model, having a drawback that it lacks search space pruning and anti-monotonic properties.

- Correlation clustering approaches: Correlation clustering algorithms are categorized by the dimensionality reduction approaches they are based on. They are discussed as follows:

- PCA-based approaches: Dimensionality reduction as a concept is required in

few instances where it is more useful in constructing a new space contrary to adopting some original data's subspaces. PCA is one such technique that aims at constructing data vectors, which indicate data variance in the given data in the best possible way. It can be described as [160] [161]: If an input matrix is $X = [x_1 \dots x_N]^T$, the linear mapping $Y = XV$ projects x_i to a low dimensional subspace. Here, the resultant matrix is Y and projection matrix is V . This approach estimates projection matrix and aims to minimize the summation of error squares that occur in approximation of the input matrix. The majority of correlation clustering approaches are based on this type of approach. HICO [162] is a hierarchical approach that uses "local correlation dimensionality" to define distance between data points. It also calculates the subspace orientation of the data points and uses "hierarchical density-based clustering" in order to derive hierarchy of clusters. ORCLUS [163] picks k seeds first and assigns data objects to these seeds. The assignment is based on a distance function founded on an Eigenvector system. This approach chooses the highest value of k , hence increasing the efficiency of the algorithm. The local dimensionality reduction technique finds correlations among data objects in the locally correlated clusters [164]. This approach enhances multi-dimensional indexing, though it is a variant of ORCLUS. Another closely related variant has been proposed in [165]. It is more efficient than ORCLUS as it works to produce higher-quality correlation clusters from noisy data. The algorithm 4C [166] is based on density of the data points. The number of clusters is not decided in advance; rather clusters expand around a seed until the density criteria is fulfilled. The density criteria specifies a minimum number of required points that should lie within the defined neighborhood of points. The definition of neighborhood comes from the distance matrix

generated from the Eigen systems of two data points. COPAC [167] is a similar to 4C, but it improves some of the ambiguities in 4C. It partitions the data space such that the search is limited to finding clusters having data points demonstrating equal “local correlation dimensionality.” This approach helps to reduce the time complexity rapidly to d^2 . ERIC [168] is another approach based on Eigen values of a data point, derived from k-nearest neighbors. The neighborhood in this case is defined by approximate linear dependency of each data point. It also takes into consideration affine distance that is further defined by the strong Eigen vectors of each point. The resulting clusters are ordered hierarchically to produce a hierarchy of subspace clusters. In [169], CURLER has been proposed. It is a method to detect arbitrary, non-linear correlations. It merges several clusters according to their co-sharing level. This makes it less sensitive to a predefined number of clusters k .

- Hough transform-based approaches: Hough transform maps points from a 2-d space of Euclidean coordinates into a parameter space. This means that each point in data space is associated with an infinite number of points, represented by a trigonometric function. The main feature of this transform is that the data points from the original data space are not kept under consideration. Objects may be far apart in the original data space, but they are associated with a common line. The Hough transform does not require locality assumption and hence helps to obtain a global subspace clustering approach [170]. CASH uses grid-based methodology on the parameter space in order to carve out dense regions. It divides the data-space based on attributes, and calculates the functions that are intersecting the hyperboxes formed as a result of the division of data space. If no correlation clusters are found in the original data space, then it can be inferred that there are

no dense regions in the parameter space [171]. A heuristic for task allocation problem was proposed for a homogeneous multiprocessor environment that is fully connected [172]. A new clustering algorithm was proposed, based on the CCLoad concept and demonstrated its efficiency.

- Fractal dimension: This concept is based on the principle of self similarity (fractal dimension) to cluster data sets. Here, the locality of patterns is not assumed. It is not as successful for correlation clustering as PCA. It does not provide information concerning the primary direction of data points within a given data space. It will take a great deal of effort to base a clustering approach on the fractal dimension. At the moment, it is an interesting field to explore [173] [174].
- Random sampling: This concept makes assumptions regarding the number of clusters (n), and the data points being evenly distributed. n is chosen as a threshold up to which the sampling is repeated. This ensures that the points obtained from one sample belong to a common cluster. This concept has been proposed in the area of pattern recognition [175]. A similar idea has been proposed in [176]. This approach also has many research gaps that can be enhanced further.
- Projected clustering approaches: In the projected clustering algorithms, subspace clusters are formed by searching for unique assignment of points. PROCLUS [177] is a clustering algorithm in which, initially, a set of M potential cluster centers is determined on a points sample. Further, for each of the k medoids, the standard deviation of the distances of points in the neighborhood of medoid to the corresponding medoid (lying along each dimension) is minimized for determining the subspace assigning the points to the closest medoid. Another similar approach called Locally Adaptive Clustering (LAC) [178] was given, which starts with k sets of d weights (for each of d attributes) and k centroids. The weights are adapted by approximating a set of

k Gaussians. LAC can be differentiated from PROCLUS in that LAC, all attributes are considered while computing weights, while in PROCLUS, average cluster dimension is specified. FINDIT [179] and SSPC [180] are variations of PROCLUS, where clustering accuracy is enhanced by defining additional heuristics and by adopting the domain knowledge in the form of labeled attributes, respectively. PReDeCon [181] uses the density-based full-dimensional clustering algorithm, which adopts specialized distance measure to capture the subspace of each cluster. It helps in determining the number of clusters automatically. Clusters may be present in corresponding subspace and may be of any shape or size. The only drawback of this approach is that the user needs to specify the number of input parameters. CLTree [157] is a method in which the existing points are assigned a common class label. The additional points are labeled as different classes. The two classes are then separated by training a decision tree.

- Hybrid clustering approaches: Hybrid clustering algorithms are those which neither assign each data point to a cluster nor find all clusters in all subspaces. These algorithms are intended to find the subspaces of only potential interest to users. A global density threshold is used to define subspace clustering in DOC [182] by using fixed-length hypercubes containing at least α points with the help of proposed random search algorithm. DOC needs to run at least k times to identify the k clusters. MINECLUS [183] [184], a variant of DOC, finds an optimal projected cluster from a given seed point by using a deterministic method. COSA [185] uses an arbitrary clustering algorithm to find a similarity matrix. The matrix also contains the weights for each point. These weights specify the subspace preferences for each point. DiSH [186] uses a hierarchical clustering model by following the same idea as PReDeCon, to determine the hierarchies of subspace clusters. This means that it checks to see if a lower di-

mensional cluster is embedded in a higher dimensional one. It uses OPTICS [187] inspired-algorithm, which is a density based hierarchical clustering algorithm. A related paper presents a systematic review of the optic sensors, their types, recent trends and future challenges [188]. HARP [189] uses the different distance function for hierarchical clustering between clusters by not producing subspace cluster hierarchy. It merges the clusters in an iterative manner until the clusters formed have a minimum number of relevant attributes. FIRES [190] uses any clustering technique to compute one-dimensional clusters which are further assimilated by applying clustering of clusters. SCHISM [191] can be used to solve the related problem by mining interesting subspaces to look for clusters, and to find the maximally interested subspaces by employing the depth-first search with backtracking. P3C [192] [193] starts with one-dimensional intervals that can be used to find approximate values of high-dimensional clusters. The resulting maximal dimensional cluster subspaces are called cluster cores. Each cluster core is then treated as an initial cluster to be refined by EM algorithm. The resulting output of the P3C algorithm is a matrix to record each data point's probability of belonging to a projected cluster. A comparison of all the properties of clustering algorithms is given in Table 2.4.

2.4 Applications of large-scale and high-dimensional data clustering

Some applications of clustering high-dimensional and big datasets are given below:

- Name disambiguation [199]: It is one task in data cleaning performed by link-based clustering. In this case, object reconciliation is done instead of object distinction. Object reconciliation is recognizing that different objects/people share the same names,

while object distinction says that the same name is assigned to different objects. The challenge in performing object distinction is that sexual similarity cannot be detected, since they have the exact same name. In this scenario, link analysis may take advantage of redundancy to facilitate entity cross-checking and validation. An example of such an approach is an algorithm that applies object distinction by information network-based cluster analysis [200]. Another way to solve this challenge is to find neighborhood similarity. The similarity between their contexts is indicated by neighbor tuples of each reference. A major problem is to identify the boundaries to cut the links to get distinct clusters. The most widely used approach is to use self-boosting in order to train the same bulky data set to set the boundaries. In order to train with same data set, we use following steps:

- Build a training set automatically by following steps:
 1. Select distinct names.
 2. Some similarity is shared by the collaboration behavior within the same community.
 3. The parameters are trained by employing a common and large set of unambiguous examples.
- A model is learned using SVM for combination of various join paths. It involves the following steps:
 - Each join path is used as two attributes (with neighborhood similarity and link-based similarity).
 - The model is all attributes' weighted sum.

Data clustering is also done by measuring similarity between clusters. Different cases can be used to measure similarity. We compare them to see which one is the best for

name disambiguation:

- Single link: This means getting the highest similarity between points. Using this is not an optimum solution as there is an easy connection between different objects' references.
 - Complete link: Minimal similarity exists among them. It is also not an optimum solution as there may be a weak connection between reference to the same object.
 - Average link: Average similarity exists among points in two clusters. It is a superior measure. A refinement over it is average neighborhood similarity and collective random walk probability.
- Evolution of heterogenous networks [201]: Networks evolve with time, they make sequences based on patterns. The motivation is to model evolution of communities in heterogeneous networks. The focus is to automatically detect the best number of communities in each timestamp, then model the smoothness between communities of adjacent timestamps. The evolution structure has to be modeled explicitly, like birth, death and split of subnetworks. An example of this application is an algorithm called the EvoNetClus [202]. It is clustering for dynamic heterogenous networks. We see co-evolution within a community. This means that within the cluster, there are heterogeneous multi-types objects/links. This approach discovers evolution structures among different communities. The general idea is to find evolutionary communities from network sequences. A graphical model represents heterogenous networks in the best possible way. It is a generative model based on the Dirichlet process mixture model [203]. At each timestamp, a cluster is dependent on historical cluster and background cluster distribution. Figure 2.6 represents a generative graphic model. Here, each graph is generated using numerous parameters, and it is then transformed to the next

graph by applying some transformative operations on it, i.e., $G_1 \rightarrow G_2 \rightarrow \dots \rightarrow G_t$.

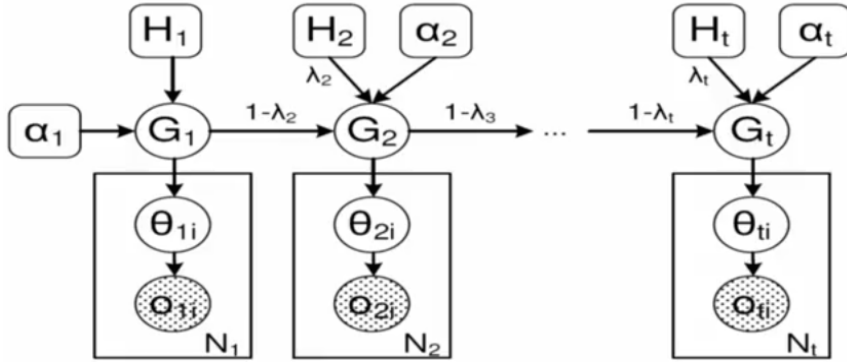


Figure 2.6: A generative graphic model

To generate a new object o_i , one must decide whether to join an existing community or a new one. Join an existing community k with probability $n_k/(l-1+\alpha)$. To join a new community k with probability $\alpha/(l-1+\alpha)$, its prior is decided. It is based on either background distribution (λ) or historical communities ($((1-\lambda)\pi_k)$), with different probabilities, and drawing the attribute distribution from the prior. o_i is generated according to attribute distribution:

$$p(o_{i,t} \mid z_{i,t} = k, \Theta_t) = p(o_{i,t} \mid \theta_{k,t}) = p(a_{i,t} \mid \theta_{k,t}^A) p(c_{i,t} \mid \theta_{k,t}^C) p(d_{i,t} \mid \theta_{k,t}^D) = \prod_{j=1}^{|A|} \theta_{k,t}^A(j)^{a_{ij,t}} \prod_{j=1}^{|C|} \theta_{k,t}^C(j)^{c_{ij,t}} \prod_{j=1}^{|D|} \theta_{k,t}^D(j)^{d_{ij,t}}$$

Greedy inference for each timestamp uses Collapse Gibbs sampling, which is trying to sample cluster labels for each target object. Based on this, we can understand community evolution discovery algorithm.

- Customer segmentation and collaborative filtering [47]: Cluster segmentation is clustering customer behavior on the similarities of their profiles and shopping behaviors, while collaborative filtering and recommendation systems are based on customer ratings of products.

- Text mining [204]: It uses a lot of cluster analysis, as it involves co-clustering of words and documents. There have been many recent studies on how to do clustering, typing, profiling and in-depth analysis of large collection of text/web documents.
- Clustering of multimedia and social media [205]: It is widely used for summarizing multimedia data (e.g., videos) and integrated analysis of news and tweets.
- Temporal and sequence applications [206]: This area has applications in a wide variety of domains, like clustering and web log click streams, to find patterns of users and to reorganize web page structures or clustering biological sequence data to construct biological models and discover anomalies.
- Social network analysis [207]: Clustering analysis is used for finding hidden social communities, i.e., community detection. It is also used for subsequent analysis for link prediction, influence analysis, classification and anomaly detection. Apart from those, there are many data-mining and broad social applications. This is an emerging field, and numerous approaches and applications are being proposed and developed every day.
- Recommendation system: It is a type of information filtering system that predicts the rating or preferences given to an item by the user. They are currently extremely popular and are used in a variety of areas. Researchers have done considerable work in this area. A personalized recommendation algorithm [208] was proposed that employs slope one scheme technology for filling up the vacant ratings in user profiles by applying user-based collaborative filtering for producing the recommendations. The accuracy of the collaborative filtering recommender system was improved using the proposed technique. An item-based collaborative filtering recommendation algorithm was proposed by [209], which used global similarity. First, the items are clustered into

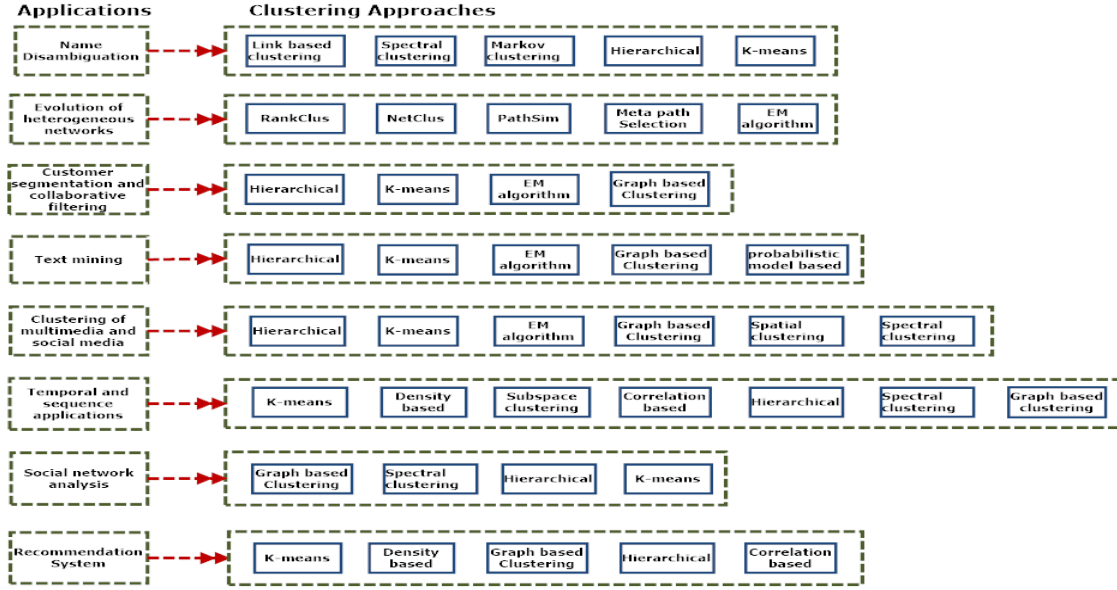


Figure 2.7: Mapping of clustering approaches to specific applications

various classes based on user ratings by using K-means algorithm for calculating local user similarity. Second, the target users' neighbor selection is optimized by employing new global similarity between users. Hence, the accuracy of prediction was improved by enhancing the quality of recommendation. [210] proposed the solution to memory-based collaborative filtering problem by using the individual user-rating as a prediction for missing user ratings. Both the user-based and the item-based approaches are used in the proposed framework. The proposed method is more robust to data sparsity. The technique given by [211] developed a procedure based on mining association rules for supporting online recommendation systems. The recommendation system implemented was tested on a sample of sales records from databases to demonstrate its efficiency. An overview of the recommender systems, and collaborative filtering methods and algorithms was presented explaining its evolution, classification and future implementation areas [212]. Mapping of clustering approaches to specific applications is given in Figure 2.7

2.5 Latest trends in large scale and high-dimensional data clustering

Some of the latest advancements in the field of data clustering are presented.

- Clustering data streams: Data streams are very popular in the age of big data, as there is considerable data coming from sensors, video feed, tweets, blogs, etc., that record data in the online data version. Data streams are continuous, ordered, changing, and fast, and have huge volume. A very important algorithm to deal with this type of data is the single-scan algorithm. Multiple data streams are flowing in a stream processing system. The users/applications are continuously sending queries to this system. The stream processing system, taking the multiple streams, tries to return query results in real time [213]. The challenge is how we can perform cluster analysis in such fast-changing data streams. Some clustering approaches have been developed to deal with this type of data:

- K-Median approach [214]: It is k-median method-based approach. Data stream points are from metric space, so medians can be easily computed. K-clusters are found in the stream to minimize the summation of distances from data points to their closest centers. As the approach deals with a data stream, it works as a constant factor approximation algorithm, as a simple two-step algorithm in small memory space. $O(k)$ centers in the string, like $S_1, \dots, S_i$, are searched by S_i for each set of M records. Each point in S_i is assigned to its closest center by performing local clustering. Let S' be centre-weighted by the number of points assigned to it. This means that the points and their weights are considered. Finally, cluster S' is used to find k centers. This is a typical approach to summarizing data stream in a real-time environment. It involves an online process dealing with a

large number of micro-clusters. Micro-clusters represent local density estimates by aggregating the information of many data points in a defined area [215].

- Hierarchical clustering tree method [216]: When there are many data points, but no capacity to process all of them, this method is used. In this approach at maximum level(m) - i medians are being maintained. At O(k) level -(i+1), medians of weights equal to the summation of the intermediate medians' weights assigned to them are developed on seeing m of them. The major concern is that quality will suffer for evolving data streams (maintaining only m level -i median), and the functionality is limited in exploring clusters over various portions of stream over time.
- CluStream Approach [217]: The design goals of this algorithm include getting a clustering of high-quality evolving data streams with stream mining and rich functionality. Stream mining consists of only one pass over the stream data with restricted space usage, but carrying high efficacy. This clustering methodology has three major points:
 1. Tilted time frame work: Most recent data is stored, so that dynamic changes are not lost.
 2. Micro-Clustering: Data is stored in a fine clustering scale having superior quality than k-means/ k-median. It is incremental, where online processing is facilitated and cluster maintenance is ensured.
 3. Two-stage processing: micro-clustering (for maintaining online data streams and their history) and macro-clustering (to answer ad-hoc queries). This will ensure that with limited overhead, scalability, high efficiency, and power of evolution/change, detection and quality of results can be achieved.

The following concepts play an important role in understanding the CluStream ap-

proach, and are also potential areas of research in the field of data clustering.

- Time dimension - A tilted time model [218]: A tilted time frame is a trade-off between space and granularity. It decides if snapshots of the statistical information are stored, and at what moment. The general design has natural (follow the natural time frame, like quarterly, weekly, monthly, etc.), logarithmic (jumps in logarithmic way, like $t, t, 2t, 4t, 8t, 16t, 32, 64t, \dots$), and pyramidal (designed by clustering) tilted time frames.
- Pyramidal tilted timeframe: It has been adopted by CluStream, and can be explained by taking an example. Six frames can be considered where each accepts a maximum of three snapshots. For a snapshot number N , if $N \bmod 2^d = 0$, it is inserted to frame numbered d . The oldest snapshot is eliminated if there are more than three. The pyramidal pattern is followed in storing the snapshots of micro-clusters set. They are accumulated at different granularity levels, which is dependent on the recency. The classification of snapshots is done into different orders varying from 1 to $\log(T)$. The snapshots of i^{th} order appear at intervals of α^i with $\alpha \geq 1$. The last $(\alpha + 1)$ snapshots are only stored.

The CluStream Framework is a micro-clustering approach that uses BIRCH CF-Tree structure. A height-balanced tree for storing the clustering features (CFs) is known as CF tree. The sums of the CFs of their children are stored in non-leaf nodes. Since all the points are not stored in the tree, the leaf nodes store micro clusters. The data locality's statistical information is contained in these. The temporal extension of the cluster feature vector $\bar{X}_1, \dots, \bar{X}_k, \dots$ is in it because it contains the multi-dimensional points with time stamps $T_1, \dots, T_k, \dots$. The d dimensions, i.e., $\bar{x}_t = (x_t^1, \dots, x_t^d)$ are contained by each point. The n points micro-cluster is termed as a $(2d + 3)$ tuple. Keeping

in mind the dynamic nature of data these days, CluStream divides clustering process into two components:

1. Online components: These are essentially micro-cluster maintenance. When the data stream is received, its statistical summary is periodically stored. Q micro clusters are initially created. Here, Q is generally significantly larger than the natural clusters count. When the online incremental updates come, it is then checked if the new point is within max-boundary, and then the point is inserted into micro cluster. Otherwise, a new cluster is created. Obsolete micro clusters may be deleted, or the two closest clusters might be merged.
 2. Offline components: This is a query-based macro clustering. When the user query is received, they are answered based on stored summary statistics. Based on a user-specific time horizon h and the count of macro clusters k , the computation of macro-clusters is done by employing k -means, or some other clustering algorithms. Quality clusters with minimal overhead are generated.
- Ensemble clustering [219]: It tries to combine many clustering models' results in creating a more robust clustering. Since clustering is unsupervised, the optimal clustering is captured by no single model or criterion, but a more robust solution is provided by an ensemble clustering of models. The methodology for the same can be described as:
 - Generate k different clustering (i.e., ensemble components): The selection of the ensemble components can be model-based (combine different parameters of different models), or data-selection based (select data from different samples of data).
 - The different results are combined into a single or more robust solution. This can be done by two major methods:

1. Hypergraph partitioning: Each data point is considered as a vertex, and a cluster in any of the ensemble components is represented as a hyper-edge.
2. Meta-clustering: It is a graph-based approach, except for the reason that in the ensemble components, vertices are associated with each cluster.

This survey lays the foundation for recognizing the research gaps, and identifying the areas on which work needs to be done. These research gaps, along with the research objectives, contribute in the formulation of the research problem addressed in this thesis.

Table 2.4: Comparison of properties of clustering algorithms

Algorithm	Complex correlations	Axis parallel	Relying on locality assumption	Deterministic	Overlapping clusters	Overlapping subspaces	hierarchical structure	Noise robust	Arbitrary no. of clusters	Integrate dimensionality reduction	Independent of cluster shape	Graph analysis	Top down approach	Bottom up approach	Hybrid approach
Standard clustering approaches															
Agglomerative Hierarchical clustering [194]				✓			✓		✓		✓			✓	
K-Means Clustering [195]				✓							✓			✓	
BIRCH [196]						✓	✓	✓	✓						✓
CURE [197]						✓	✓		✓		✓			✓	
Expectation-Maximization [73]	✓		✓	✓	✓	✓		✓			✓				✓
Graph & network data clustering															
SimRank [106]									✓		✓	✓			✓
RankClus [108]				✓				✓			✓	✓			✓
CHAMELEON [198]					✓	✓	✓	✓	✓		✓	✓			✓
NetClus [109]				✓					✓		✓	✓			✓
PathSim [110]						✓			✓		✓	✓			✓
Subspace clustering															
CLIQUE [154]		✓		✓	✓	✓		✓	✓		✓			✓	
ENCLUS [155]		✓		✓	✓	✓		✓	✓		✓			✓	
MAFIA [156]		✓		✓	✓	✓		✓	✓		✓			✓	
nCluster [157]		✓		✓	✓	✓		✓	✓		✓			✓	
SUBCLUE [158]		✓		✓	✓	✓		✓	✓		✓			✓	
DBSCAN [103]		✓		✓	✓	✓		✓	✓		✓			✓	
DUSC [159]		✓		✓	✓			✓	✓		✓			✓	
Correlation clustering															
HICO [162]	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓			✓	
ORCLUS [163]	✓	✓	✓			✓				✓	✓			✓	
4C [166]	✓	✓	✓	✓		✓		✓	✓	✓	✓			✓	
COPAC [167]	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓			✓
ERIC [168]	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓			✓	
CURLER [169]	✓	✓	✓	✓		✓		✓	✓	✓	✓			✓	
CASH [171]	✓	✓		✓			✓	✓	✓	✓	✓			✓	
LGBACC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓
Projected clustering															
PROCLUS [177]		✓	✓			✓					✓		✓		
(LAC) [178]		✓	✓			✓					✓				✓
FINDIT [179]		✓	✓			✓					✓				✓
SSPC [180]		✓	✓			✓					✓		✓		
PRDeCon [181]		✓	✓	✓		✓		✓	✓		✓		✓		
CLTree [157]		✓	✓	✓		✓	✓	✓	✓		✓			✓	
Hybrid clustering															
DOC [182]		✓			✓	✓					✓		✓		
MINECLUS [183] [184]		✓		✓	✓	✓	✓				✓		✓		
COSA [185]		✓	✓	✓		✓		✓			✓		✓		
DiSH [186]		✓		✓		✓	✓	✓			✓			✓	
OPTICS [187]		✓		✓		✓	✓	✓			✓		✓		
FIRES [190]		✓		✓	✓	✓		✓	✓		✓			✓	
SCHISM [191]		✓		✓		✓		✓	✓		✓			✓	
P3C [192] [193]		✓		✓	✓	✓		✓	✓		✓			✓	

Chapter 3

Problem Formulation

The exhaustive literature survey and comparative study on big and high-dimensional clustering bring to light the importance of clustering in the field of data mining. The discussion about the importance of correlations in data, and the section on correlation clustering algorithms, highlight the importance of research in this area. Finding correlations in data not only helps to define relationships between data points, but helps to uncover relationships that were not known to exist. There is a need to understand the concept of correlations in datasets. Further, this understanding needs to be applied to the concept of data clustering. To complete the understanding, the problem of correlation clustering, as defined in the literature, needs to be comprehended along with getting a broad view of the solutions proposed to address this problem.

Though numerous correlation clustering approaches are proposed by different researchers, this area is still evolving. The problem has been approached by different employing methodologies resulting in different solutions for the same problem. The simplest solution can be finding correlations between data points, using the mathematical concept of correlation coefficients. The available coefficients are not suitable for making predictions in very large datasets: there is a need to define a correlation coefficient that is independent of sample size;

that measures the degree of monotonicity; is insensitive to outliers; and that demonstrates intuitively.

Another way to define correlations in data is to find clusters in a given subspace. To do so, there is a need to combine the dimensionality reduction techniques with those of data clustering. Many correlation clustering techniques have used this kind of methodology to effectively find correlations in the given data. The problem is that none of the approaches merges these two concepts perfectly. Given all this information, we list a few research gaps and the resultant research objectives that have been addressed in this thesis.

3.1 Research Gaps

- It is observed clustering of big data majorly takes place by implementing the point assignment algorithms or the hierarchical algorithms. The most common problem in point assignment algorithms remains the detection of the centroids of the clusters. Mostly, these points are randomly selected. Hierarchical clustering, on the other hand, cannot represent distinct clusters with similar expression patterns. Also, as clusters grow in size, the actual expression patterns become less relevant.
- There is a need for a hybrid approach that eliminates the disadvantages of both. Certain hybrid approaches, which carry out hierarchical clustering first to decide location and the number of clusters in the first round, and to run the point assignment clustering in another round. This type of mechanism not only complicates the solution, but also increases computational time considerably, given the huge size of the data.
- Big Data analysis relies hugely on the formation of correlations among significant data points, which focus on the “what” part of the data rather than the “why” part. In clustering algorithms there is minimal-to-no-usage of establishment of significant

correlations.

- Recommender Systems based on effective hybrid algorithms, which uses correlations, need to be developed and implemented. Further, literature validating such systems also needs to be made available for further research and improvements.

3.2 Research objectives

- Study and analyze the big data clustering approaches from hierarchical and point assignment viewpoints
- Propose an improved hybrid algorithm for big data clustering using correlations
- Develop a recommendation system model using the proposed algorithm
- Investigate the developed model by encoding it and populating it with large data sets.

3.3 Methodology for achieving the objectives

The workflow followed to achieve the objectives is given in Figure 3.1. In an effort to accomplish the set objectives, we followed a certain set of steps. The methodology applied to achieve each one of the objectives is given below:

Objective 1:

- Identified the approaches used for data clustering, and focused on the big data clustering approaches
- Performed a survey on large-scale and high-dimensional data clustering
- Surveyed the proposed solutions for the correlation clustering problem

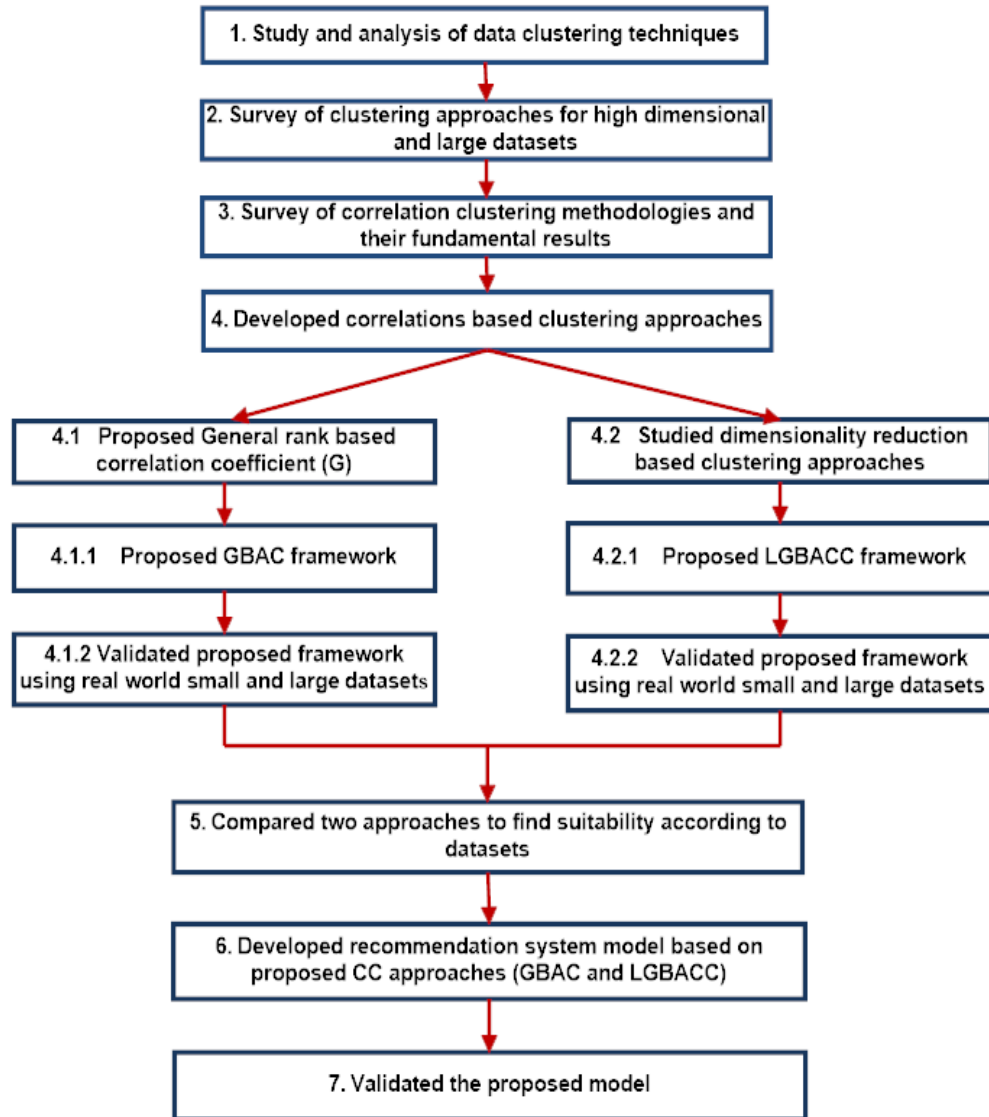


Figure 3.1: Flow of the work done

- Performed comparative study of the surveyed big data and high-dimensional clustering approaches, graph and network data clustering, subspace clustering, projected clustering, hybrid clustering and correlation clustering algorithms

Publications (accepted/communicated):

- Divya Pandove, Shivani Goel and Rinkle Rani, “Systematic Review of Clustering High-Dimensional and Large Data Sets”, ACM Transactions on Knowledge Discovery in Data, SCIE-Indexed [Minor revision submitted]

- Divya Pandove, Shivani Goel and Rinkle Rani, “Correlation Clustering Methodologies and their Fundamental Results”, Expert Systems, SCIE-Indexed [In Press]
- Divya Pandove, Shivani Goel, “A Comprehensive Study on Clustering Approaches for Big Data Mining”, International Conference on Electronics & Communication Systems, IEEE, pp. 1333-1338, February 2015

Objective 2:

- Derived a General (rank based) Correlation Coefficient (G): This metric is defined in the form of an algorithm. It is a non-parameterized quantity, which defines the relationship between two variables, using monotonic functions. This means that the proposed metric does not access linear relationship between the variables; rather, it looks at their monotonic relationship. The inputted variables are ranked before applying any other mathematical operations. This metric is evaluated on the criteria defining a predictive metric and is found to fulfill all the conditions.
- Developed GBAC (G Based Agglomerative Clustering) framework: This framework uses G to find correlations between data points represented as a data matrix, and generates a correlation matrix. This matrix is then manipulated by converting it into a dissimilarity matrix. This final matrix is then used to generate correlation clusters using the agglomerative clustering technique (AGNES). The clusters obtained are tested on the basis of the quality using parameters like Agglomerative coefficient and Fowlkes-Mallow (FM) index. Visualization plots are also generated as part of the final output to depict the clusters generated.
- Developed LGBACC (Local Graph Based Correlation Clustering) framework: This approach has been designed in a way that the fundamental concepts of data reduction

technique (PCA) have been merged with those of the graph theory and agglomerative clustering. The final clusters are generated using the agglomerative clustering approach (AGNES). The quality of the clusters is checked on the same parameters as those of GBAC. This approach takes lesser execution time as compared with other agglomerative clustering techniques because of the reduced number of data objects, and produces quality clusters.

Publications (accepted/communicated):

- Divya Pandove, Shivani Goel and Rinkle Rani, “An Intuitive General Rank based Correlation Coefficient”, *Frontiers of Information Technology & Electronic Engineering-SCIE Indexed*, doi of ‘10.1631/FITEE.1601459’ [In Press]
- Divya Pandove, Shivani Goel and Rinkle Rani, “General Correlation Coefficient based Agglomerative Clustering”, *Cluster Computing-SCIE Indexed* [Minor revision submitted]
- Divya Pandove, Rinkle Rani and Shivani Goel, “Local Graph based Correlation Clustering (LGBACC)”, *Knowledge-Based Systems-SCIE Indexed* [Minor revision submitted]

Objective 3:

- Developed a recommender system model based on the proposed correlation clustering approaches. This model integrates the correlated points obtained as a result of running LGBACC or GBAC with existing recommendation algorithms.
- Used the model to make recommendations by training and testing the proposed model, and then validating the results by running a k-fold cross validation model.

- Tested the model by running it on a real-time, large and high-dimensional dataset (MovieLens - 1GB) and compared its performance with the existing recommendation algorithms.

Publications (accepted/communicated): Divya Pandove, Rinkle Rani and Shivani Goel, “A Novel Correlation based Recommendation System for Big Data using Sparkr”, SCIE Indexed, IEEE transactions on Parallel and Distributed Computing-SCIE Indexed [Communicated]

Objective 4:

- Tested G by using real and simulated datasets, and compared it to existing correlation coefficients like Spearman’s rank (ρ) and Kendall rank correlation coefficient (τ)
- Tested GBAC by using real data sets, and evaluated cluster quality by using existing cluster quality parameters.
- Tested LGBACC by using real data sets, and evaluated cluster quality by using the same parameters as used for evaluating GBAC.
- Proved the scalability of the algorithms by running them on very large scale datasets and compared them to existing agglomerative clustering algorithms.

The detailed layout of experimental results and discussion is given in Figure 3.2.

Publications (accepted/communicated):

- Divya Pandove, Rinkle Rani and Shivani Goel, “A Novel Correlation based Recommendation System for Big Data using Sparkr”, SCIE Indexed, IEEE transactions on Parallel and Distributed Computing-SCIE Indexed [Communicated]

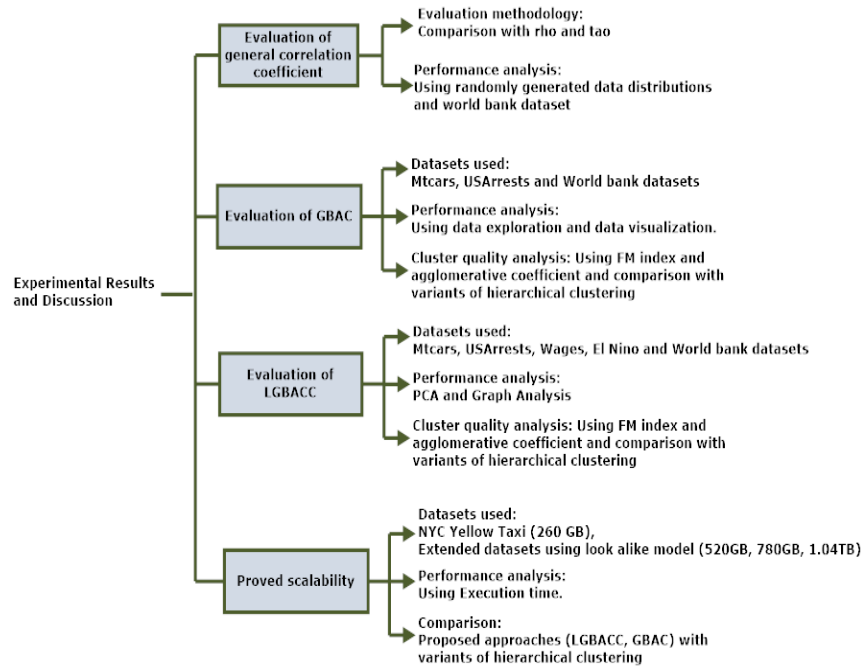


Figure 3.2: Layout of experimental results and discussion

- Divya Pandove, Shivani Goel , “Prototyping and in-Depth Analysis of Big Data Benchmarking”, 14th IEEE International Conference on Ubiquitous Computing and Communications, Liverpool, pp. 1222-1229, October 2015

The Figure 3.3 shows the mapping of objectives with publications (accepted/communicated).

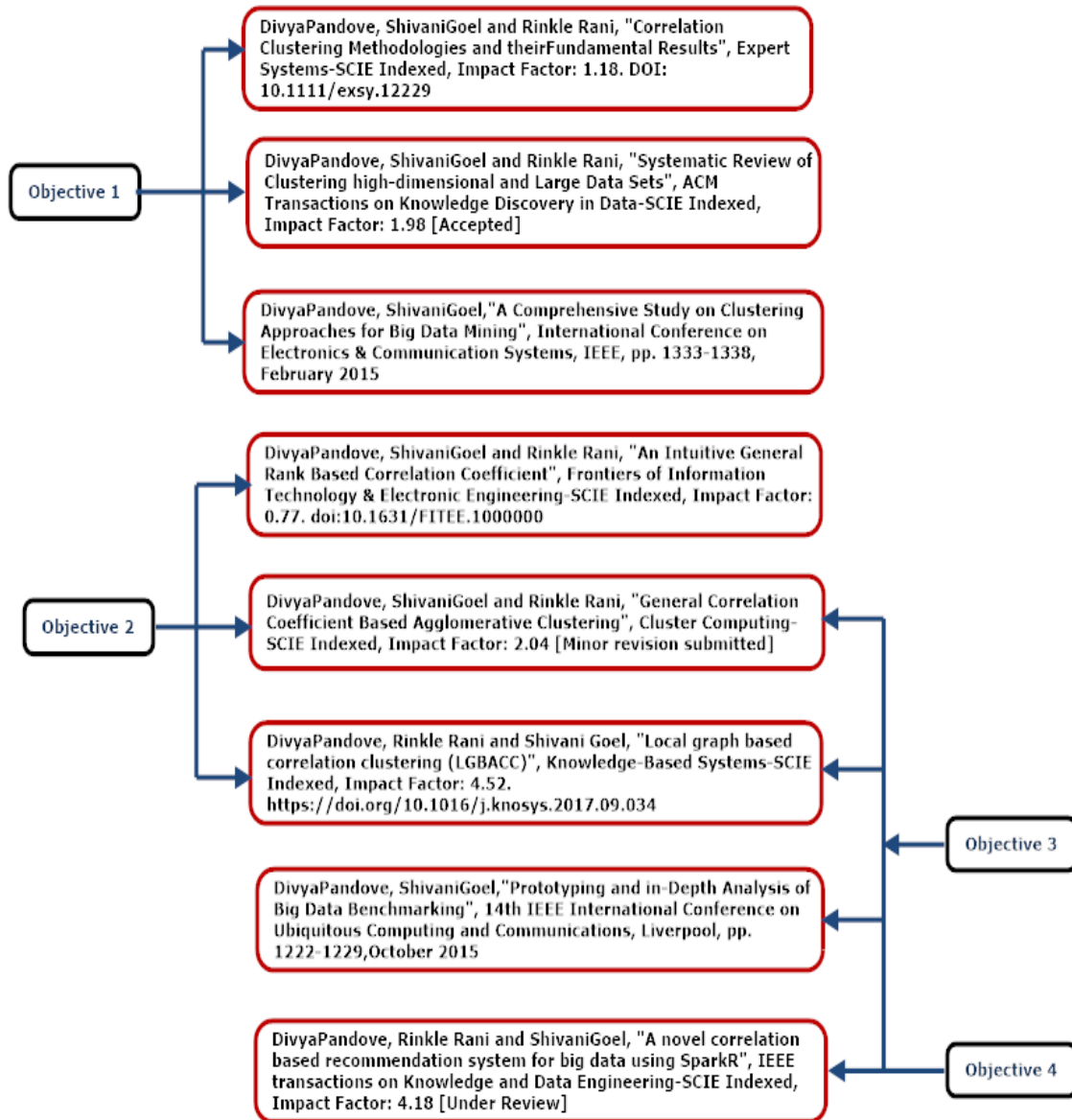


Figure 3.3: Mapping of objectives with publications

Chapter 4

Preliminaries

In this chapter, we enlist the concepts that have been used to describe the correlation clustering frameworks proposed in the following chapter. These concepts are the building blocks of the proposed frameworks. As the two approaches are evaluated using the same quality parameters, there are a few of concepts that overlap. There are also quite a few different concepts that are unique to a particular approach. For the ease of understanding, all of these preliminaries are listed at one place.

- **Correlation and dissimilarity matrix:** Once the data is ingested in the computing environment, a correlation coefficient measure is used to compute correlations between paired values and to generate a correlation matrix. The values in this matrix lie between -1 and 1. This correlation matrix acts as an input for the distance or dissimilarity matrix. Dissimilarities are always positive and are represented as $d(i, j)$. They are small when i and j are close to each other and become large when i and j are very different. It is assumed that dissimilarities are always symmetric and dissimilarity of an object to itself is zero. In this thesis, dissimilarities are calculated from the

correlation matrix using the formula [220]:

$$d(f, g) = 1 - |R(f, g)| \quad (4.1)$$

where, $R(f, g)$ is the correlation coefficient value.

- **Fowlkes-Mallows (FM) index:** It is an external evaluation method, used to determine the quality of a cluster. It is a measure of similarity used to make a comparison between the proposed clustering and a benchmark classification (in our case, hierarchical clustering). A higher value of the FM index indicates a greater similarity between the clusters, and the benchmark classifications. FM index can be calculated as follows [221]:

We consider two hierarchical clusterings for n objects that can be labeled as A_1 and A_2 . The trees A_1 and A_2 , which are obtained as a result of this clustering can be cut to produce $k = 2, \dots, n - 1$ clusters for each tree. For each value of k we obtain corresponding values of m given as:

$$M = [m_{i,j}]$$

Here, $i = 1, \dots, k$ and $j = 1, \dots, k$

$m_{i,j}$ gives the common objects between i^{th} cluster of A_1 and j^{th} cluster of A_2 .

The FM index for a given value of k can be defined as:

$$B_k = \frac{T_k}{\sqrt{P_k Q_k}} \quad (4.2)$$

Here, $T_k = \sum_{i=1}^k \sum_{j=1}^k m_{i,j}^2 - n$

$$P_k = \sum_{i=1}^k (\sum_{j=1}^k m_{i,j})^2 - n$$

$$Q_k = \sum_{j=1}^k (\sum_{i=1}^k m_{i,j})^2 - n$$

B_k can be calculated for each value of k . To show similarity between two clusterings, B_k is plotted against k as for each value of k , $0 \leq K \leq n$. When testing unrelated clusterings using this index, it is found that as the number of data points keeps on increasing, the index keeps on decreasing until it approaches zero. This property makes this index more accurate for evaluating unrelated clusterings. If noise is added to existing data sets and their similarities are compared, it is found that the value of the index is found to decrease as the noise increases. Hence, this is a reliable index for measuring similarity between the clusters [222].

- **Agglomerative coefficient (AC):** It is calculated from the banner produced as a graphical output of the algorithm AGNES. For each object i , the line containing its label is identified and its length $l(i)$ is measured on the 0-1 scale above and below the banner. The agglomerative coefficient of the data set is defined as [220]:

$$AC = \frac{1}{n} \sum_{i=1}^n l(i) \quad (4.3)$$

The AC is a dimensionless quantity having a value between 0 and 1, as $l(i)$ lies between 0 and 1. It is the average width of the banner and it remains the same even when all the original dissimilarities are multiplied by a constant factor. AC defines the strength of the clustering structure that has been obtained by group average linkage method. If $d(i)$ denotes the dissimilarity of object i to the first cluster, it is merged with other clusters, divided by the dissimilarity of the merger, in the last step of AGNES. Then Agglomerative Coefficient is given by average of all $1 - d(i)$.

- **Principal component analysis:** PCA is a dimensionality reduction technique that aims at constructing data vectors that indicate data variance in the given data in the best possible way. It can be described as [160] [161]: If an input matrix is $X = [x_1 \dots x_N]^T$, the linear mapping $Y = XV$ projects x_i to a low dimensional subspace. Here, the resultant matrix is Y and the projection matrix is V . This approach estimates projection matrix and aims to minimize the summation of error squares that occur in approximation of the input matrix. PCA proceeds by mapping the input patterns into the feature space and aims at solving the Eigenvalue problem by using feature value's covariance matrix. It is suitable for Gaussian Distributions since it is dependent on second order relationships formed in the covariance matrix [223]. PCA is a mature technique that relies on the construction of a wide range of similarity measures that help to capture local correlations of attributes. This then helps to find arbitrarily oriented subspace clusters. The only drawback of this approach is the locality assumption. PCA is also very sensitive to outliers which means that if there are noise points in the local neighborhood then the subspace determination may be incorrect [224]. A flowchart depicting the process of PCA is given in Figure 4.1. Once the data is imported into the computing environment it is standardized and the covariance matrix is computed. Subsequently, Eigenvalues and Eigenvectors are calculated based on the covariance matrix. These values are then arranged in the decreasing order and the components lower down the list are ignored in order to obtain principal components. These components are next arranged to form a feature vector matrix. This matrix is ultimately used to obtain final data by using the formula: *Final data = Row Feature Vector X Row Data Adjust*. Here, Row Feature Vector is the matrix with the Eigenvectors in the columns transposed so that the Eigenvectors are in the rows with the most significant Eigenvector at the top and is the mean-adjusted data transposed, i.e., the data items are in each column with

each row holding a separate dimension. The final data set is derived by taking data items in columns and dimensions along the rows.

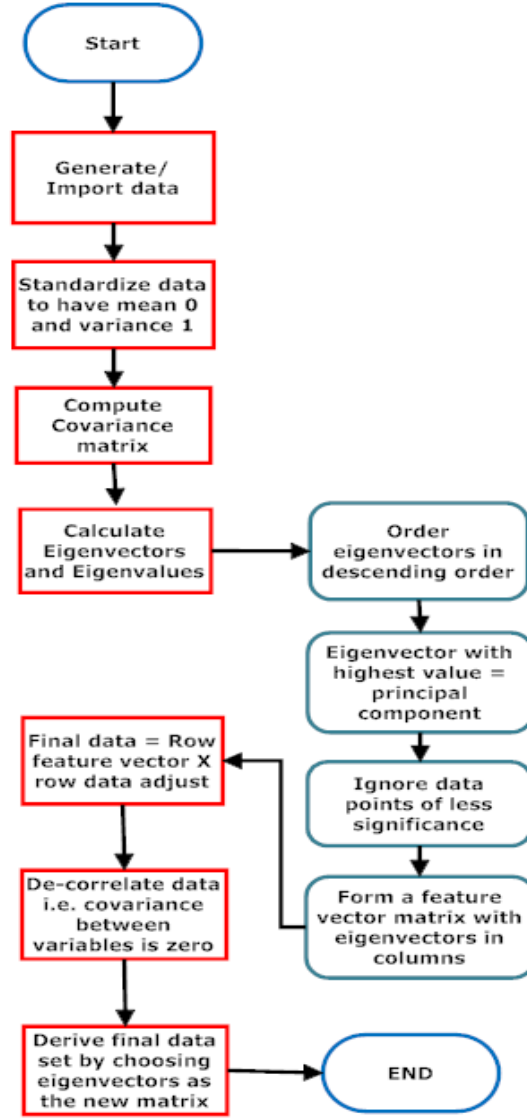


Figure 4.1: A flowchart depicting the process of PCA

- **Local covariance matrix:** Local covariance matrix generated in the framework of the proposed scheme is not based on a range query predicate because it is based on hierarchical clustering. There is no predefined query radius. The concept of local correlation matrix is based on covariance matrix. Covariance matrix is a matrix the elements of which are defined by the covariance between elements of a random vector.

This means that the elements at i^{th} and j^{th} positions in a covariance matrix represent covariance between i^{th} and j^{th} positions of a random vector. A random vector has random scalar variables having multiple dimensions [225]. If L is a correlation cluster in data base D then, $L \subseteq D$. We define the database as having n feature vectors and d dimensions. Let C be the centroid of all the points in L . The covariance matrix of L , Σ_L is $d \times d$ dimensional and can be defined as [224]:

$$\Sigma_L = \frac{1}{|L|} \cdot \sum_{C \in L} (C - \bar{C}) \cdot (C - \bar{C})^T \quad (4.4)$$

Covariance matrix in different contexts can be defined as:

Definition 1. (Covariance matrix generated by PCA)

If A is a given set of points, where $\bar{a}_i$ is the mean of all points in B , then the covariance matrix generated by PCA for a given point is given as [226]:

$$B = [b_{ij}], \text{ with } b_{ij} = \sum_{a \in A} (a_i - \bar{a}_i) \cdot (a_j - \bar{a}_j)$$

It decomposes into an orthogonal matrix V (Eigenvector matrix) and diagonal matrix E (Eigenvalue matrix) such that $B = V \cdot E \cdot V^T$

Definition 2. (Covariance matrix of a point P)

Let $P \in D$, where D is the dataset. The matrix $B_P = [b_{ij}]$ is given by [224]:

$$b_{ij} = \sum_{a \in N_{\epsilon}^P} (a_i - \bar{a}_i) \cdot (a_j - \bar{a}_j), (1, \leq i, j \leq d) \quad (4.5)$$

Here, $\bar{a}_i$ is the mean of all points $A \in N_{\epsilon}^P$ in attribute i , is called the covariance matrix

of point P. B_P as determined by PCA for a given point is given by:

$$B_P = V_P \cdot E_P \cdot V_P^T$$

Here, V_P are the Eigenvectors and E_P are the Eigenvalues of point P.

Definition 3. (Local covariance matrix)

Local covariance matrix (Σ_P) is the covariance matrix of the k nearest neighbors of point P. Given that $k \in N$, $k \leq |D|$. The local covariance matrix Σ_P of a point $P \in D$ w.r.t k is formed by r nearest neighbors of P. Let $\bar{Y}$ be the centroid of $NN_r(P)$, then [224]:

$$\Sigma_P = \frac{1}{|NN_r(P)|} \cdot \sum_{x \in NN_r(P)} (Y - \bar{Y}) \cdot (Y - \bar{Y})^T$$

- **Similarity matrix:** A similarity matrix may be defined as a symmetric matrix B which, represents qualitative assessment of relative similarity between every pair of points in a data set. The measure of similarity is represented as $B_{ij} \geq 0$, where B is a symmetric matrix. The similarity between observations x_i and x_j is denoted in a similarity matrix as $B_{ij} = b(x_i, x_j)$. Common similarity measures are: Euclidean distance: $s(x_i, x_j) = -||x_i - x_j||^2$ and Gaussian kernel: $s(x_i, x_j) = \exp(-a||x_i - x_j||^2)$. Gaussian kernel gives $B_{ij} \approx 1$, and when two points are far apart $B_{ij} \approx 0$ [227]. The correlation similarity matrix is used to compute correlation distance from one point to another in a data set defined as follows [228]:

Definition 4. (Correlation similarity matrix of a point)

Let us consider a point P, such that $P \in D$. V_P and E_P are the the Eigenvectors and Eigenvalues of the point P respectively. Let C be a constant with $C \geq 1$. Now in order

to calculate a new Eigen value matrix with $\hat{E}_P$ having diagonal entries $\hat{e}_i$.

$$\hat{e}_i \begin{cases} 1 & \text{if } \Omega(e_i) > \delta \\ C & \text{if } \Omega(e_i) \leq \delta \end{cases} \quad (4.6)$$

Here, Ω represents normalization of the Eigenvalues to convert into $[0, 1]$. The correlation similarity matrix is represented as: $\hat{B}_P = V_P \cdot \hat{E}_P \cdot V_P^T$. The correlation similarity matrix measure associated with point P is denoted by:

$$cdist_P(P, O) = \sqrt{(P - O)^T \cdot \hat{B}_P (P - O)}$$

- **Affinity matrix:** Another concept used in the proposed framework is the affinity matrix. It is made up of positive values and is a type of the symmetric matrix. It represents affinity scores between the two objects. It is computed by applying k-nearest neighbor in order to build a matrix of closest data points. Affinity matrix replaces clustering by a graph partition problem. Here, graph components are interpreted as clusters. The graph construction must be partitioned in a way that the edges connecting different clusters should have low weights. Even the edges within the same clusters must also have high values [229]. We consider three nearest neighbors for each point including the point as well. Given the similarity matrix as an input the output of the clustering contains at most three vertices of the graph G' . Here, G' is a complete graph formed out of a larger graph H . If we apply optimal clustering on H then, each cluster will contain at most 3 vertices from G [16]. Given, two data points x_i and x_j , Affinity $A_{i,j}$, that is positive and symmetric and depends on the Euclidean distance $\|x_i - x_j\|$

between the two data points, is defined as:

$$A_{ij} \simeq \exp(-\alpha \|x_i - x_j\|^2) \quad (4.7)$$

Where, α is a constant.

- **Graph Laplacian transform:** The graph Laplacian of graph G , L_G has Eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ where, $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ and Eigenvectors $\{v_1, v_2, \dots, v_n\}$. Eigenvalues reveal global graph properties which is not apparent from edge structure. If 0 is the Eigenvector of L with K different Eigenvectors, i.e., $0 = \lambda_1 = \lambda_2, \dots, = \lambda_k$, then G has K connected components. If the graph is connected, $\lambda_1 \geq 0$ and λ_2 is the algebraic connectivity of G . The greater λ_2 , the more connected G is [230]. The graph Laplacian of G is represented in Figure 4.2. In this framework, simple laplacian is applied as, it

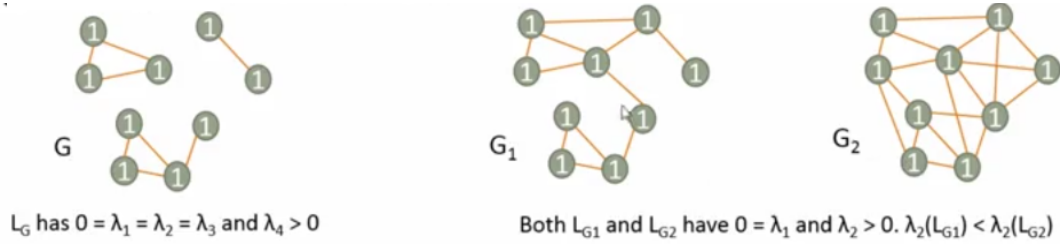


Figure 4.2: Graph Laplacian of G

can be defined as: $L = D - A$. Here, A is the Affinity matrix and D is the normalized factor of A .

Chapter 5

Proposed Correlation Clustering

Approaches

This chapter explores the role of correlation analysis in data clustering. Correlation analysis is an effective mechanism for studying patterns in data and making predictions. Many experiments in data science are based on this concept. The focus of this chapter is on studying correlations in data and forming clusters based on them. We propose two approaches that use different concepts of correlation analysis to cluster data points. Both of these approaches (GBAC and LGBACC) are presented in this chapter. GBAC is based on a General (rank based) correlation coefficient (G), which has been proposed and proved. Further, the framework for GBAC has been given and its phases have been explained. It is followed by an algorithm giving the steps of implementing the proposed framework. Similarly, the framework for LGBACC has been proposed and explained along with the corresponding algorithm.

5.1 General Correlation Coefficient based Agglomerative Clustering (GBAC)

In order to bring out meaningful patterns in data a technique should be developed that analyzes data points irrespective of their size and dimensionality. The most intuitive way to do so is to find correlations in data. This involves performing computations on thousands of data points. These data points have to be in the form of paired observations obtained from any number of sources such as data buckets, sensors, time series or even data simulations. A non-causal approach should be followed for such data analysis. Causality can be defined as “when one factor is shown to contribute to the development of a variable and its removal affects the frequency of that variable [231].” There is a need to refine the existing correlation determining factors and to come up with an approach that would factor in the different problems encountered in high-dimensional data. We focus on developing a metric that is in the form of an algorithm and is predictive in nature. A predictive metric should fulfill five main criteria [232]:

1. It should be independent of sample size to enable comparison across datasets of various sizes.
2. It must lie between -1 and 1, with 0 meaning no correlation. This is similar to the existing correlation measures in order to obtain compatibility.
3. It should be general such that it should measure the degree of monotonicity (X grows with Y) rather than linearity ($X = aY$). This means that it should be more general than the traditional correlation measures but must not be as general as the distance correlation that is equal to 0 if and only if X and Y are independent.
4. It should have no sensitivity to outliers so that it is robust

5. It should be intuitive so that it translates how the human brain perceives correlations

The criteria for a predictive metric are represented in Figure 5.1:

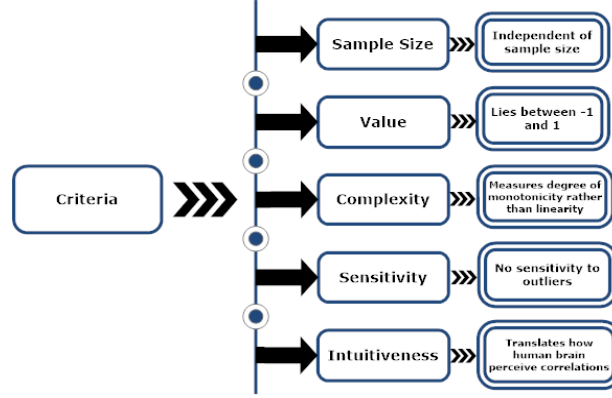


Figure 5.1: Criteria for a predictive metric

Keeping these parameters in mind, we propose the General correlation coefficient (G) based on ranking.

5.1.1 General (rank based) correlation coefficient (G)

We start by considering Spearman's rank correlation coefficient which is given by [233]:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (5.1)$$

where, $d_i = x_i - y_i$ is the difference between ranks of observations. This coefficient lies between -1 and 1, which satisfies the second criterion of the five main criteria mentioned previously. Now, we propose the General (rank based) correlation coefficient (G) given in algorithm 5.2. The hierarchy of need and motivation for G is given in Figure 5.2. The diagram shows that the need for G comes from the shortcomings of traditional metrics, like Spearman's rank correlation coefficient. These metrics, combined with the domains and datasets in which, work has been done, brings to light the research gap in this field. The

motivation for G comes from a need to develop a general metric for correlation analysis that meets all the criteria for a predictive metric and can be used in varied domains.

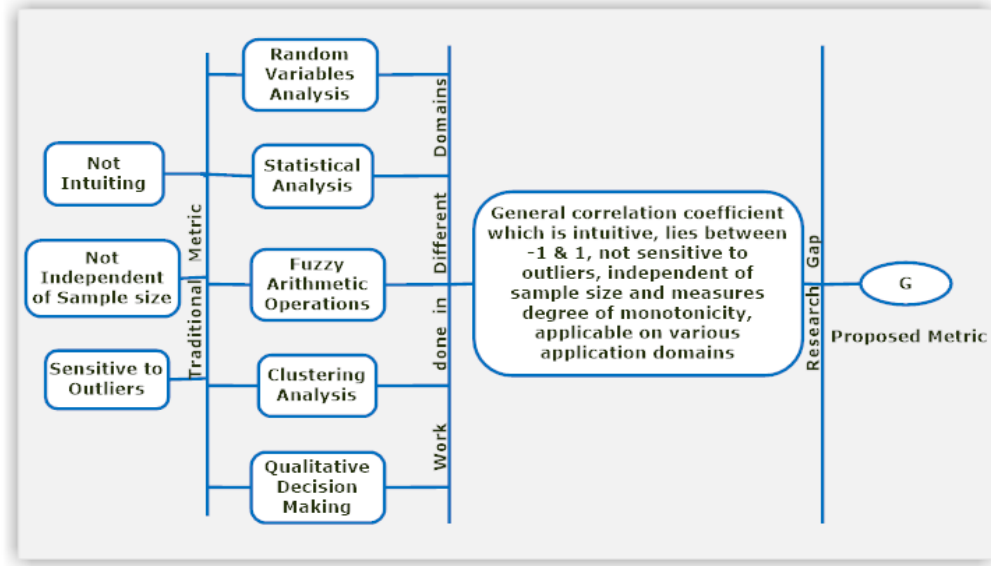


Figure 5.2: The hierarchy of need and motivation for G

This algorithm gives steps to calculate G. This is a non-parameterized correlation coefficient. This means that the final value of G, is not based on parameterized families of probability distributions. Non-parameterized statistics make no assumptions about the probability distribution of the variables being assessed. While there are fixed number of parameters in parametric models, in non-parametric models the number of parameters grows with the amount of training data [234]. This algorithm describes how well a relationship between two variables can be evaluated, by using a monotonic function. This means that the final value of G is obtained by applying a function between ordered sets of paired variables. This function preserves the given order, and fulfills the condition of a predictive metric given by [232]. The proposed metric does not access linear relationships between the variables. By working with the ranked values of the variables, it accesses their monotonic relationship. This metric is appropriate for both continuous and discrete variables, including ordinal variables. The values of paired variables are stored in arrays X and Y. The first step is to perform ranking on

both these arrays. The difference of the ranked values is then squared to minimize the affect of the sign of the ranks. The mean of this value is then calculated to give a normalized value of the ranked data. The general correlation coefficient is calculated by the formula:

$$G = 1 - \frac{2.25 * \overline{d^2}}{D} \quad (5.2)$$

Here, 2.25 is the balancing constant of the given equation. We arrived at this value by testing constants ranging from 0 to 6. This constant balances the numerator ($\overline{d^2}$) and the denominator D. The numerator is the sum of ranks of the values of two arrays X and Y, divided by n. We start by considering two arrays X and Y, taking values between 1 to n. As G is a rank-based correlation coefficient, we rank all the values of these arrays. We subtract the ranks of array Y from array X and calculate the square of the sum of these differences (similar to Spearman's rank correlation coefficient). Now, we divide this sum by n, to obtain the mean of this sum and represent it by ($\overline{d^2}$). Now, we derive the denominator 'D', which is the heart and soul of G. The denominator is represented in terms of quotients and the remainder obtained by Euclidean division of n. The denominator (D), like the denominator of Spearman's rank correlation coefficient, is in terms of n, but in a different form.

We begin by taking two equations a(x) and b(x), that express relationship between x (any permutation of integers between 0 to n-1) and i (no.of iterations).

$$a(x) = \text{sum} \|x(i) - i\| \quad (5.3)$$

$$b(x) = \text{sum} \|x(i) - (n - 1 - i)\| \quad (5.4)$$

Algorithm 5.2 General (rank based) correlation coefficient calculation

Require: Two arrays X and Y containing values for n observations

Ensure: General correlation coefficient for paired observations

```

for i=1 to n do
     $R_x = \text{Rank}[X]$ 
     $R_y = \text{Rank}[Y]$ 
     $d = R_x[i] - R_y[i]$ 
     $\text{sum}(d^2)$ 
end for
 $\overline{d^2} = \frac{\text{sum}(d^2)}{n}$ 
 $q = n \% 4, r = n \% 4$ 
 $D(n) = \frac{6q^2 + 4qr + r(r-1)}{2}$  (from eq 5.11)
if  $D(n) \% 2 == 0$  then
     $D = \frac{6q^2 + 4qr + r(r-1)}{2}$ 
else
     $D = \frac{6q^2 + 4qr + r(r-1)}{2} - 1$ 
end if
 $G = 1 - \frac{2.25 * \overline{d^2}}{D}$ 

```

The minima of these two equations is given by $t(x)$ as:

$$t(x) = \min(a(x), b(x)) \quad (5.5)$$

Here, x is any permutation of integers from 0 to $n - 1$. Now, we compute the maximum value of $t(x)$ over all permutations x of the integers 0 to $n - 1$. This value is represented as $d(n)$

Table 5.1: Symbol table for derivation of denominator D

Symbol	Meaning
x	Any permutation of integers from 0 to $n - 1$
$a(x)$	$\text{sum} \ x(i) - i\ $
$b(x)$	$\text{sum} \ x(i) - (n - 1 - i)\ $
n	Equation given by: $n = 4q + r$
$c(x)$	sum of a(x) and b(x)
c_n	Maximum value of $c(x)$
$f(i, j)$	$\ j - i\ + \ j - (n - 1 - i)\ $
p	particular value lying between 0 and $n - 1$
D	Denominator given by: $D(n) = 6q^2 + 4qr + \frac{r(r-1)}{2}$

and can be calculated by performing Euclidean division of n . Let q be the quotient and r be the remainder for the Euclidean division [235] of n by 4. Then, n is represented as:

$$n = 4q + r, \quad (0 \leq r < 4) \quad (5.6)$$

Since $d(n)$ is dependent on $t(x)$ and x varies between all permutations of x between 0 and $n - 1$, we calculate the upper and lower bounds of this equation.

- **Upper bound:** To calculate the upper bound on $d(n)$, it is noted that $t(x) \leq \frac{1}{2}c(x)$ where, $c(x) = a(x) + b(x)$. We represent the maximum value of $c(x)$ as c_n , computed for all the permutations x of integers ranging from 0 to $n-1$. $d(n)$ can be represented in terms of c_n as:

$$d(n) \leq \frac{1}{2}(c_n) \quad (5.7)$$

Adding the equations 5.3 and 5.4, we get

$$c(x) = \sum_i f(i, x_i) \quad (5.8)$$

where, $f(i, j) = \|j - i\| + \|j - (n - 1 - i)\|$ Considering the function f , we define its values as:

$$f(i, j) = \begin{cases} 2j - (n - 1) & j \geq i; j \geq n - 1 - i \\ 2i - (n - 1) & j \leq i; j \geq n - 1 - i \\ -2i + (n - 1) & j \geq i; j \leq n - 1 - i \\ -2j + (n - 1) & j \leq i; j \leq n - 1 - i \end{cases}$$

Now, we compute the values of $f(i,j)$ for various conditions for a particular value p lying between 0 and $n-1$. The values of i and j are varied from p to $n-1-p$ such that one of the following conditions is true:

- $p \leq i \leq n-1-p$ and $j = p$
- $p \leq i \leq n-1-p$ and $j = n-1-p$
- $p \leq j \leq n-1-p$ and $i = p$
- $p \leq j \leq n-1-p$ and $i = n-1-p$

These conditions are based on threshold values for the computation.

Next step is to compute the maximum value of $c(x)$ over all the permutations of x from 0 to $n-1$. As q is the quotient of n , when divided by 4, and r is the remainder, the corresponding value of c_n (max value of c_x) in terms of q , r , and p is derived as:

$$\begin{aligned}
 c_n &= \sum_{p=0}^q (n-1-2p) + r(n-1-q) \quad (\text{From equation 5.8}) \quad (5.9) \\
 &= q(n-1) - 2 \sum_{p=0}^q (p) + r(n-1-q) \\
 &= q(n-1) - 2 \frac{q(q-1)}{2} + r(n-1-q) \\
 &= qn - q - q^2 + q + r(n-1-q) \\
 &= q(4q+r) - q^2 + r(4q+r-1-q) \\
 &= 4q^2 + qr - q^2 + 4rq + r^2 - r - rq \\
 &= 3q^2 + 4qr + r(r-1)
 \end{aligned}$$

Since, we are considering the divisibility factor of 4, let us consider the cases when n

is even by substituting q by $2k$. The equation reduces to:

$$c_n = 12k^2 + 8kr + r(r-1) \quad (5.10)$$

As, $D(n) = c_n/2$, therefore $D(n)$ is:

$$D(n) = 6k^2 + 4kr + r(r-1)/2 \quad (5.11)$$

As the sum of even terms is even, $D(n)$ is even. For odd terms, $D(n) = D(n) - 1$.

- **Lower bound:** To calculate the lower bound, we consider the case when the remainder r is 0, as lower bound will depend on the value of r . We will use, $d(n) \geq t(x)$, where x is a permutation. Taking $r = 0$, we get $n = 4m$. The permutation x^0 can be defined as:

$$x^0 = \begin{cases} x^0(i) = 2m + i, & \text{if } 0 \leq i < m \\ x^0(i) = 2m - 1 - i, & \text{if } m \leq i < 2m \\ x^0(i) = 6m - 1 - i, & \text{if } 2m \leq i < 3m \\ x^0(i) = -2m + i, & \text{if } 3m \leq i < n \end{cases}$$

We obtain value of $t(x)$ for permutation x^0 as:

$$t(x^0) = 6k^2 \quad (5.12)$$

Therefore, $d(n) \geq 6k^2$. As $n=4q$,

$$d(4q) \geq 6k^2 \quad (5.13)$$

From equations 5.11 and 5.13, $d(4q) = 6k^2$ or $d(4q) = 6q^2$. Therefore,

$$d(q) = \frac{3}{8}q^2 \quad (5.14)$$

Similarly, $D(n) = \frac{3}{8}n^2$ if, n is a multiple of 4.

The symbols used in this derivation are given in Table 5.1. The value of D for a particular value of n is constant. A table containing values of D up till 1000 values of n is given in the Appendix B. G , like spearman's rank correlation coefficient, does not make any assumptions about the parameters of the population from which samples are drawn. Being an algorithm based measure it is easier to apply as compared to other rank based correlation coefficients. Since, ranking is used instead of full set of observations there is some loss of information but G is more suited to make predictions. It satisfies the five criteria of a predictive metric, i.e, independent of sample size, lies between -1 and 1, measures monotonicity, not sensitive to outliers, and intuitive. Like every other rank based correlation measure G also works on paired values.

5.1.2 Proposed framework for GBAC

We propose a framework for the G Based Agglomerative Clustering (GBAC) given in Figure 5.3. The framework has three main phases and is independent of any programming language. This means that the data can be manipulated in the same way regardless of the programming language and computer platform. The three phases of the model are discussed as follows:

1. Data synthesis phase: In this phase, the data sets are collected and they can be unstructured, semi-structured, structured, real or synthetic. Once the crude data sets are selected they are imported in the computing environment being used for analysis. This

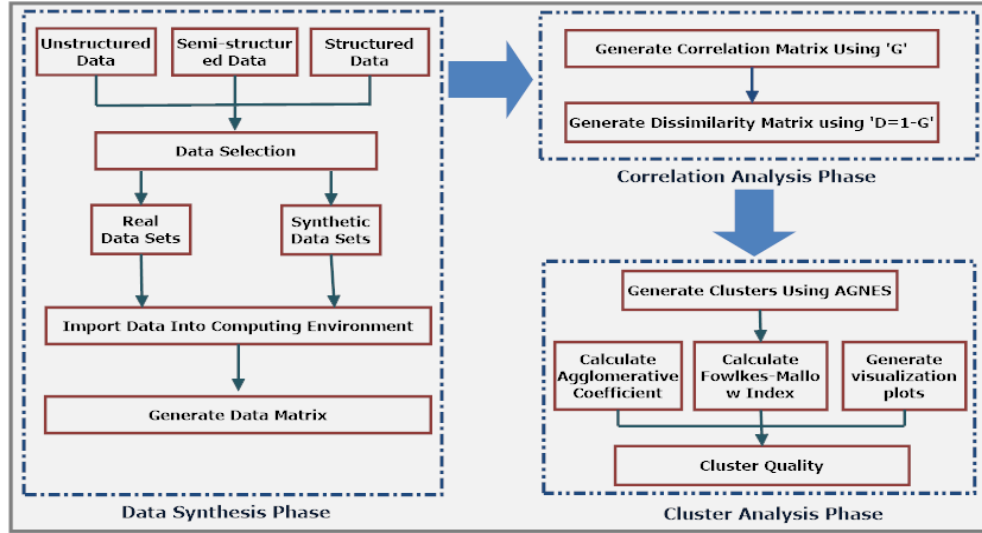


Figure 5.3: Proposed framework for General correlation coefficient based agglomerative clustering

data is cleaned and arranged in the form of a data matrix. This data matrix model contains the chosen attributes of the data sets and their corresponding values and is the output of the data synthesis phase.

2. Correlation analysis phase: The data matrix model obtained from the previous phase is given as input for correlation analysis. Pairs of attributes are selected from the matrix and correlations are found between them using G . These values are stored in a matrix and we call it the correlation matrix. This matrix has values of -1 & 1 and correlation between an attribute with itself is 1 . This matrix is then converted into the dissimilarity matrix by subtracting each element of the matrix from 1 . The dissimilarity matrix model is generated in order to get a distance parameter known to clustering algorithms. This model is the output of the correlation analysis phase.
3. Cluster analysis phase: In this phase clusters are generated based on the dissimilarity matrix model. The algorithm AGNES is applied on this matrix and visualization plots along with agglomerative coefficient are obtained. The quality of clusters is tested us-

ing the Fowlkes-Mallows (FM) index. This testing model is the final output of GBAC framework. The algorithm describing the steps of this framework is given in Algorithm 5.3.

Algorithm 5.3 G based Correlation Clustering algorithm (GBAC)

Require: Data Matrices $A[i][i]$, $B[i][i]$, length of A,B = n, Input data is represented as a data matrix

Ensure: High Quality Clusters

$Rank_i$ is calculated as: $R_i = A[i][1]$, $S_i = B[i][1]$

Correlation coefficient is calculated as:

$$G = 1 - \frac{2.25 * d^2}{D}$$

Calculate t-statistic:

if ($G==1$) or ($G==-1$) **then**

$$t = r * \infty$$

else

$$t = G \times \sqrt{\frac{N-2}{1-G^2}}$$

end if

Calculate P value: $P = 2 \times (1 - cdf^*(abs(t), N - 2))$

Generate correlation matrix $C[4][n]$ with dimensions: $A[i][1], B[i][1], G_i, P_i$

Calculate dissimilarity matrix using $D = 1 - G$

Generate clusters using AGNES

Check cluster quality using:

a) Agglomerative coefficient (AG) = $\frac{1}{n} \sum_{i=1}^n l(i)$

b) Fowlkes-Mallow index $B_k = \frac{T_k}{\sqrt{P_k Q_k}}$

c) Visualization plots

*cdf: Cumulative distributive function

5.2 Local Graph based Correlation Clustering (LGBACC)

In high-dimensional data, clusters often exist in the form of complex hierarchical relationships. In order to explore these relationships there is a need to integrate dimensionality reduction techniques with data mining approaches and graph theory. The correlations in data points emerge more clearly if this integration is flawless. We propose an approach called Local Graph based Correlation Clustering (LGBACC). This approach merges hierarchical

clustering with PCA to uncover complex hierarchical relationships and uses graph models to visualize the results. We propose a framework of this approach that is divided into four phases. Each phase is flawlessly integrated with the next phase. Visualization of data after each phase is an important output and is knitted into the fabric of the framework. The focus of this technique remains on obtaining high quality clusters.

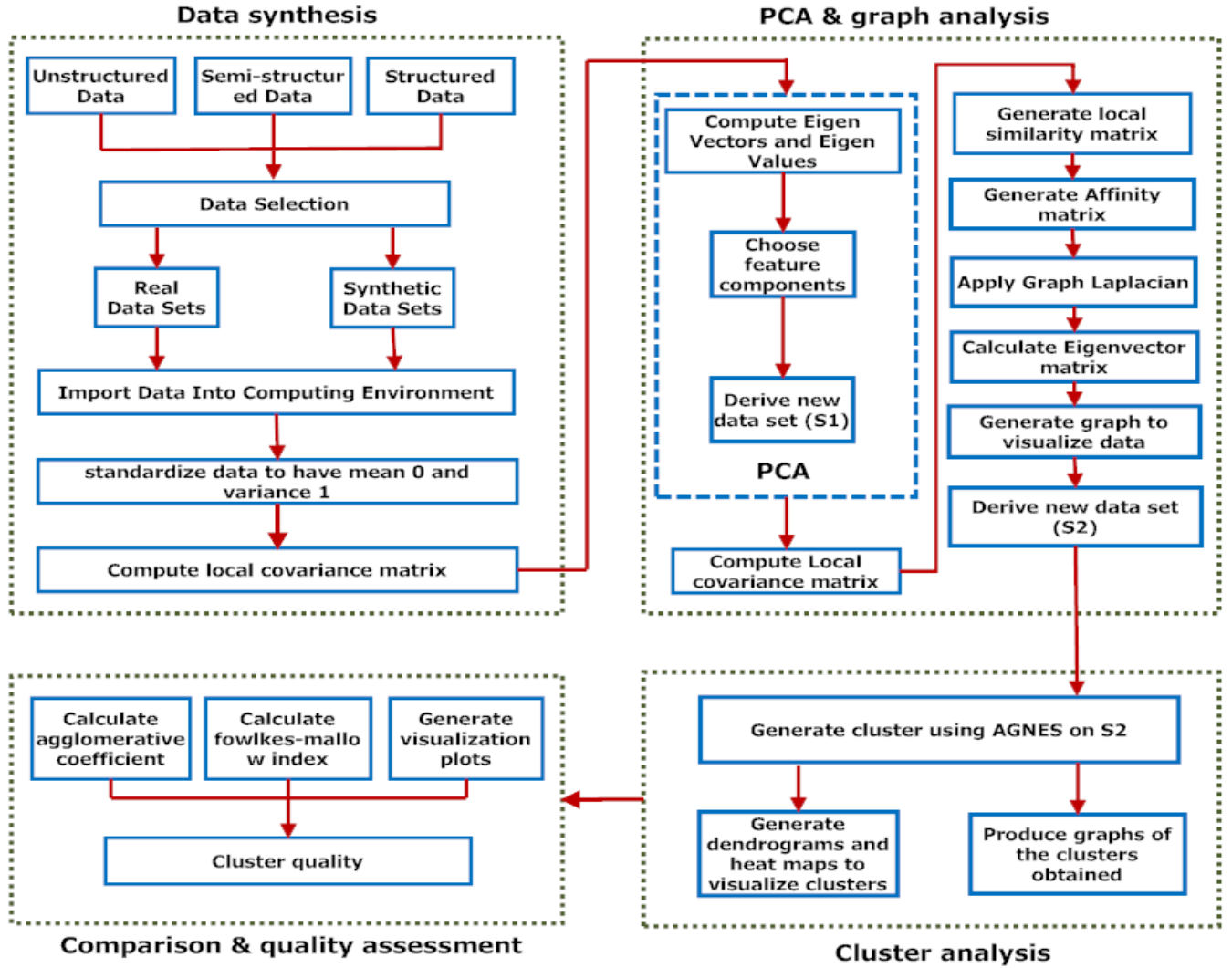


Figure 5.4: Proposed framework for local graph based correlation clustering

5.2.1 Proposed framework for LGBACC

A framework has been proposed for the local graph based correlation clustering given in Figure 5.4. The proposed scheme has four main phases. It is developed independent of any programming language. This framework integrates the concepts of PCA with graph theory and hierarchical clustering to produce high-quality clusters. All the different phases are well integrated with each other the output of one phase flawlessly flows into the next phase. In this section, we propose the framework of LGBACC and explain its different phases in detail. the steps of the approach are given in Algorithm 5.4.

- **Data synthesis phase:** In this phase the data sets are selected and can contain unstructured, semi-structured or structured data. These data sets can be real or synthetic. Once the crude data sets are selected, they are imported in the computing environment being used for analysis. This data is standardized and arranged in the form of a data matrix. This matrix is then used to calculate the local covariance matrix (defined in the previous chapter).
- **PCA and graph analysis phase:** The local covariance matrix generated in the previous phase acts as input for PCA and graph analysis. This phase is the heart and soul of this framework. It combines the underlying elements of PCA and graph theory to analyze the local covariance matrix. The output of this phase is a processed new dataset.
 - **PCA :** In this phase the covariance matrix acts as an input to compute Eigenvalues and Eigenvectors. These values help to determine principle components. Once the feature components are chosen, they are selected to form a new dataset S_1 . This is a reduced dataset, having only principle components, hence is easier to manipulate. This dataset is then used to generate a local covariance matrix for

Algorithm 5.4 Local Graph based Correlation Clustering (LBACC) Algorithm

Require: Data Matrices $A[i][i]$, $B[i][i]$, length of A,B = n, Input data is represented as a data matrix

Ensure: High Quality Clusters

Standardize data (mean=0, variance=1)

a) $\bar{a} = \text{Mean}(A[i][1]) = 0$, $\bar{b} = \text{Mean}(B[i][1]) = 0$

b) $\sigma^2(A[i][1]) = \sigma^2(B[i][1]) = 1$

Compute local covariance matrix using:

$$\text{Cov}(A[i][1], B[i][1]) = \frac{\sum_{i=1}^n (A[i][1] - \bar{a})(B[i][1] - \bar{b})}{n-1}$$

Apply PCA on local covariance matrix

a) Calculate Eigenvectors and Eigenvalues using:

$$B = [b_{ij}], \text{ with } b_{ij} = \sum_{a \in A} (a_i - \bar{a}_i) \cdot (a_j - \bar{a}_j)$$

b) Select principal components

c) Derive new data set S1

d) Compute local covariance matrix using:

$$\text{Cov}(A[i][1], B[i][1]) = \frac{\sum_{i=1}^n (A[i][1] - \bar{a})(B[i][1] - \bar{b})}{n-1}$$

Apply graph theory analysis as:

a) Generate local similarity matrix $S_{i,j} = s(a_i, b_i)$ from S1, represented as:

$$\hat{B}_P = V_P \cdot \hat{E}_P \cdot V_P^T$$

b) Generate affinity matrix using:

$$A_{ij} \simeq \exp(-\alpha \|x_i - x_j\|^2)$$

c) Apply Simple Graph Laplacian as:

$$L = D - A$$

d) Calculate Eigenvector matrix:

e) Derive new dataset S2

Generate clusters using AGNES

Check cluster quality using:

a) Agglomerative coefficient (AG) = $\frac{1}{n} \sum_{i=1}^n l(i)$

b) Fowlkes-Mallow index $B_k = \frac{T_k}{\sqrt{P_k Q_k}}$

Exit

graph analysis.

- Graph analysis: The local covariance matrix is used to generate the correlation similarity matrix (S) . The next step is to compute affinity matrix (A) based on S1. ‘A’ has to be symmetric and contain positive values. This is done by applying

the k-nearest neighbor filter in order to obtain a graph representation depicting the closest points in the dataset. Since the matrix is symmetric, if the element A_{ij} is selected, A_{ji} will also be selected as a nearest neighbor. Next we calculate an unnormalized graph Laplacian given by $U = D - A$, and then normalize it by using a simple Laplacian. A simple Laplacian is given by $H = D^{-1}A$. This can be seen as a random walk, where $D^{-1}A$ is a transition matrix. This process produces groups of nodes such that the random walk rarely transitions from one group to another. The last computation of this phase is to produce an Eigenvector matrix. This is then used to generate a graph to visualize data before clustering and to generate a new dataset S_2 .

- **Cluster analysis phase:** In this phase, the processed dataset (S_2) is used to generate clusters using the algorithm AGNES. These clusters are then visualized using dendrograms and graphs. Since the original dataset has been processed to include graph friendly data, it is easier to visualize clusters in the form of small graphs. Here, because of reduced dimensionality, it is easier to see relationships between different attributes and distinction between clusters. This phase combines the qualities of agglomerative data clustering with data visualization.
- **Comparison and quality assessment phase:** In this phase the quality of the clusters generated using LGBACC is compared with the existing hierarchical clustering algorithms. This is done by using indices like Agglomerative Coefficient, FM index and comparative (Bk) plots.

5.3 Conclusion

Correlation clustering has been defined by using two different approaches. Both of the approaches have been defined using the preliminaries described in the previous chapter. The correlation coefficient used to define the first algorithm GBAC, has been derived and its algorithm has been proposed. This metric is found to fulfill all the conditions of a predictive metric. The framework of GBAC has been defined and all of its phases have been explained. The corresponding algorithm has also been proposed. The other approach, LGBACC, is different as it employs dimensionality reduction and graph theory, but is similar to GBAC as it uses agglomerative clustering for cluster formulation. The parameters for checking the quality of the two approaches are also the same. This gives a sense of symmetry to both of the proposed approaches. The experimental evaluations and result analysis for G, GBAC and LGBACC are given in the next chapter.

Chapter 6

Experiments and Implementation Details

In this chapter, the details of the experiments performed to test the proposed techniques are given. These results have been compared to the existing techniques along with a detailed analysis of underlying metrics of respective techniques.

6.1 Datasets

We enlist the real time datasets that have been used to perform experiments on the proposed approaches.

- **World Bank:** This dataset is taken from worldbank.org/indicator and contains indicators of human development. It contains about 20 indicators for 235 counties of the world, having 2355 data values. A mind map containing attributes of the World Bank development indicators is given in Figure 6.1. This dataset is a compilation of relevant and high-quality information about the quality of people's lives. The data are internationally comparable statistics about development. The dataset has been used by many researchers including [236], [237], and [238].
- **USArrests dataset:** This standard dataset contains statistics for crimes of assault,

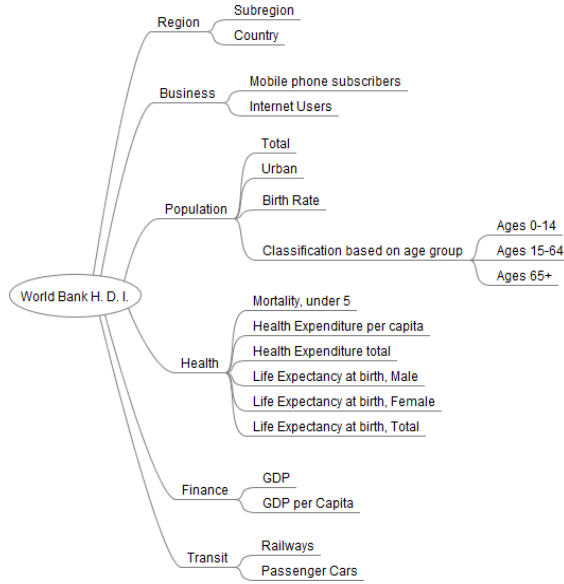


Figure 6.1: Mind map representing attributes of World Bank dataset

murders, and rapes in each of the 50 US states and arrests per 100,000 residents. It also contains percentage of population living in urban areas. This dataset has been used for research purpose by many authors such as [239] and [240]. We have used it to compare our results with the results of standard hierarchical clusterings. A scatterplot depicting this dataset is given in Figure 6.2.

- **Mtcars dataset:** This dataset contains motor trend car road tests and comprises of fuel consumption and 10 aspects of automobile design and performance for 32 automobiles. This dataset has also been used by many authors such as [241],[242] and [243] to prove their hypothesis. A scatterplot depicting this dataset is given in Figure. 6.3.

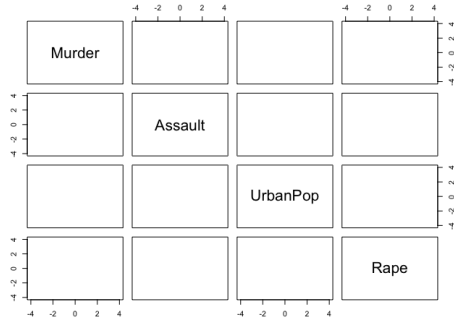


Figure 6.2: Scatterplot for USArrests dataset

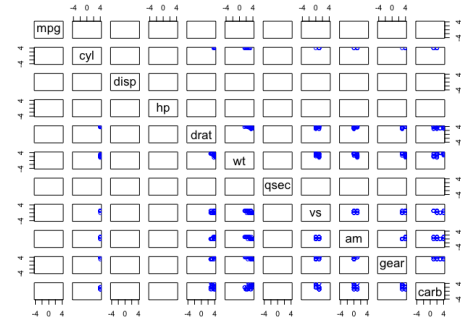


Figure 6.3: Scatterplot for Mtcars dataset

- Wages for 85 CPS:** This database contains supplement census information, collected in between the census years. This data consists of information regarding the wages and other information of 534 workers. Many inferences can be drawn out of this data such as what all characteristics are wages dependent on and whether there is a gender gap in the wages etc. The attributes of the dataset are education, south, sex, experience, union, wage, age, race, occupation, sector and marriage. Some of these variables have discrete numeric values and the others have binary values. This dataset has been used to validate dimensionality reduction clustering techniques such as: [168], [171], [167], etc. It can be found at http://lib.stat.cmu.edu/datasets/CPS_85_Wages and is depicted using a scatterplot in Figure 6.4.
- El Nino:** This is a benchmark dataset taken from <https://archive.ics.uci.edu/ml/datasets/El+Nino>. It is collected using a number of bouys placed in equatorial pacific. It contains oceanographic and surface meteorological readings. This data can be used to study weather conditions throughout the world, climatic variations, reliability of the buoy collecting data, etc. It has been used by researchers such as [244], [162], etc. The El Nino dataset can be visualized using a scatterplot in Figure 6.5. A brief summary of the described datasets used is given in Table 6.1.

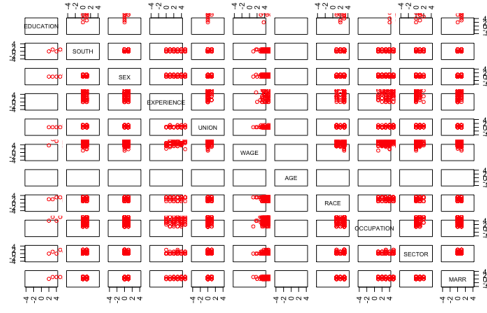


Figure 6.4: Scatterplot for Wages

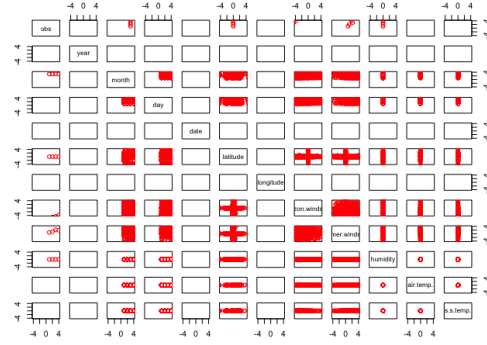


Figure 6.5: Scatterplot for El Nino

Table 6.1: Summary of datasets

Dataset	No. of dimensions	No. of data points	Application area
US Arrests	4	200	Crime statistics
Mtcars	11	352	Automobile design and performance
Wages	11	5874	Statistical information on workers
El Nino	12	178080	Oceanographic readings
World Bank	20	553425	Indicators of human development

6.2 Evaluation of General (Rank based) Correlation Coefficient

The proposed General (Rank based) correlation coefficient (G) is a symmetric metric. This means its value will not change, even if the two array values are swapped. The value of G is obtained by considering a pair of values or attributes, and finding correlation between their values. The objective of the experimentation is to compute G , and compare it to the existing correlation coefficients. We have selected two existing correlation coefficients: Spearman's rank correlation coefficient (ρ) and Kendall rank correlation coefficient (τ). Both are non parametric tests for correlation analysis. Both of these are based on ranking of data points and are briefly discussed as follows:

- Spearman's rank correlation coefficient which given by [233]:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (6.1)$$

The value of the correlation coefficient lies between -1 and 1 and assesses monotonic relationships between the rank values of those two variables.

- Kendall rank correlation coefficient is used to measure ordinal association between two measured quantities. It is also a ranked correlation coefficient, but unlike spearman's rank correlation coefficient, it is not affected by how far from each other the ranks are but only by, how equal the ranks are between observations. Hence, it is appropriate for measuring discrete variables but, it is not defined for continuous variables. To define Kendall we consider two samples X and Y each of size n. $\frac{n(n-1)}{2}$ represents the number of possible pairing between X and Y. Now the pairs are ordered first by x value then by y value. A pair of values is concordant when, both x and y are ordered and $X[i] > Y[i]$. If n_c represents the number of concordant pairs, and n_d represents number of discordant pairs then Kendall rank correlation coefficient (τ) is given by [245]:

$$\tau = \frac{n_c - n_d}{n(n-1)/2} \quad (6.2)$$

6.2.1 Performance analysis

We select the World bank dataset as, its attributes are suitable for evaluation of the performance of G on the predictive metric criteria. These observations are real time but are smaller in number. Performance of G also needs to be tested on larger number of observations. For this purpose, we have evaluated G on a synthetic, large dataset as well. The data has been

generated using different random number distributions such as uniform, binomial, exponential, poisson and normal. The results obtained are tabular, visual and comparative in nature. Experiments have been performed on a Dell workstation T- 5600 with 2.5 GHz processor and DDR3 8GB, 1600 MHz RAM. It has a 1GB network card with GPU computing. R programming language has been used to perform all types of analysis, data visualizations and simulations.

- Simulated datasets: We have generated data using random number distributions to test G at $n=15000, 45000, 75000$ and $1,00,000$ data points. The aim of this simulation is to test the behavior of G at different values of n , and compare it to ρ and τ . Various random distributions have been used to generate different patterns of data. The results of comparisons on all the above mentioned random number distributions are given in Table 6.2. It is observed that G remains comparable to ρ but fares better than τ . The result obtained at values of $n = 15,000; 45,000; 75,000;$ and $100,000$ have been plotted for different distributions. The aim is to visually compare G with ρ and τ , and see how it performs on the predictive metric criteria. It is found that for binomial and exponential distributions G and ρ coincide while both of them are different from τ . It can be said that G demonstrates properties similar to ρ , which means that G assesses the degree monotonicity between the variables, and not a linear relationship. In normal, uniform and poisson distributions it differs a little from both τ and ρ . This shows that G is a balanced ranking correlation metric that is appropriate, both for discrete and continuous variables. It can also be observed in all the distributions, that G is independent of sample size as it continues to give consistent results irrespective of the sample size. On all values of n , for all distributions, G remains comparable to ρ and performs better than τ . It performs better than τ as it does not have any limitations of τ and can also be defined for continuous variables, that can be shown

by the comparative graphs in Figure 6.6, Figure 6.7, Figure 6.8, Figure 6.9 and Figure 6.10. These graphs are based on the values given in Table 6.2. The blue line represents value of ρ , green line that of τ and red the value of G . The purpose of these graphs is to visually demonstrate the differences, and similarities in the values of G , ρ and τ , for different distributions.

Table 6.2: Simulations on random number distributions to compare correlation coefficients

Simulation Method	Data Generation		Rho(ρ)	Tau(τ)	G
	No. of data points	Parameters			
Uniform random variables	15000	lower value:1 Upper value:15000	-0.0059	-0.004	-0.0055
	45000	lower value:1 Upper value:45000	-0.002	-0.0013	-0.002
	75000	lower value:1 Upper value:75000	-0.003	-0.002	-0.003
	100,000	lower value:1 Upper Vaue:10000	0.003	0.002	0.003
Exponential Random variables	15000	Mean:0.2	0.0088	0.0058	0.0088
	45000	Mean:0.2	0.0027	0.00184	0.0027
	75000	Mean:0.2	-0.0033	-0.0022	-0.0034
	100,000	Mean:0.2	-0.0042	-0.0028	-0.0042
Normal random variables	15000	Mean:15 Standard Deviation:2	-0.0047	-0.0031	-0.0049
	45000	Mean:15 Standard deviation:2	-0.005269	-0.0035	-0.00549
	75000	Mean:15 Standard deviation:2	0.00247	0.00165	0.00312
	100,000	Mean:15 Standard deviation:2	0.00114	0.00076	0.00211
Poisson random variables	15000	Mean:10	0.0017	0.0013	0.0111
	45000	Mean:10	0.00737	0.00533	0.01667
	75000	Mean:10	0.0048	0.0035	0.0155
	100,000	Mean:10	-0.0058	-0.00421	-0.00245
Binomial random variables	15000	parameter:10 probability of success:0.3	0.0015	0.0012	0.0441
	45000	parameter:10 probability of success:0.3	-0.0039	-0.003148	0.03916
	75000	parameter:10 probability of success:0.3	-0.00185	-0.00147	0.04065
	100000	parameter:10 probability of success:0.3	-0.0015	-0.00121	0.04104

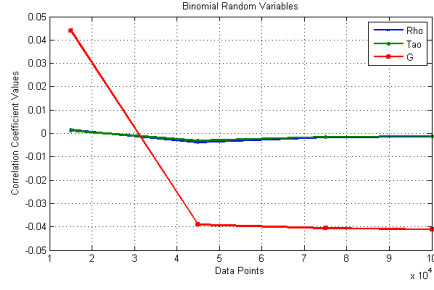


Figure 6.6: Comparison of G with ρ and τ for Binomial distribution

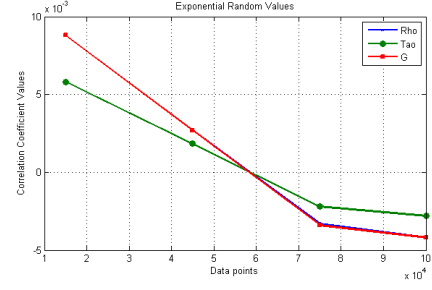


Figure 6.7: Comparison of G with ρ and τ for Exponential distribution

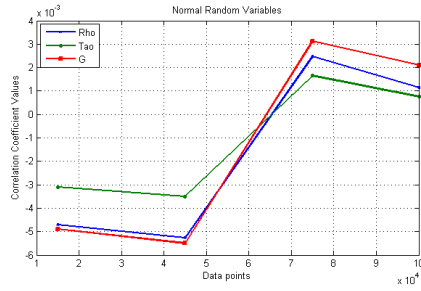


Figure 6.8: Comparison of G with ρ and τ for Normal distribution

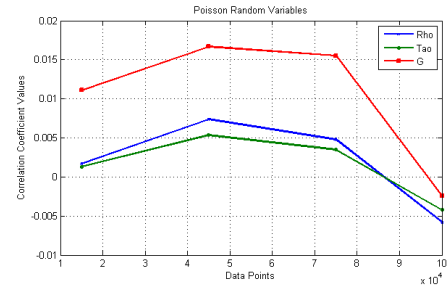


Figure 6.9: Comparison of G with ρ and τ for Poisson distribution

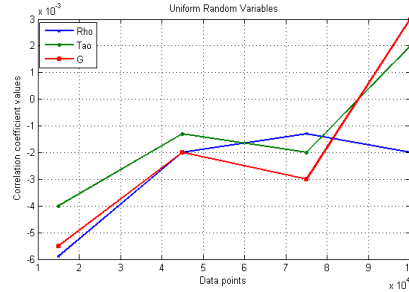


Figure 6.10: Comparison of G with ρ and τ for Uniform distribution

- **Real datasets:** We have selected 13 attributes indicating human development given by the World Bank. These attributes are: GDP per capita, GDP, passenger cars, urban population, internet users, mobile subscribers, mortality (under-5), birth rate, total population, urban population, life expectancy and health exp. per capita. These attributes have then been used to compute G and compare it with ρ and τ . The results of this comparison have been given in Table 6.3 and have been graphically demonstrated

in Figure 6.11. These real world attributes reaffirm the results obtained on the synthetic data of random variable distributions. In addition to that, it also indicates that G is more intuitive than ρ and τ . It has higher value of correlation where, according to human understanding, it should be higher and vice versa. ρ and G are comparable in some instances, but G has better intuitive power. There is no metric to test the accuracy of intuitiveness. It can only be gauged by understanding the dependance relationships between the attributes, according to the conventional wisdom and comparing it with the values obtained from the correlation coefficients. For instance common sense says that there should be a high correlation between GDP per capita and GDP, the value of G at 8.284 shows a high correlation. Similarly higher the GDP per capita, more number of passenger cars can be purchased, this is also shown by a high correlation value of 0.98 given by G . The correlation between mortality under-5 and birth rate is neither high nor low and G at 0.5974 demonstrates exactly that. Value of G for the rest of the attributes also demonstrate high power of intuitiveness. It is matched by ρ at placed but τ does not perform well when it comes to power of intuitiveness. Both the real and synthetic data sets indicate that G fulfills the criteria of monotonicity, intuitiveness and independence from sample size. The value of G , given its construction shall always lie between -1 and 1.

Table 6.3: Comparison of correlation coefficients on World Bank dataset

No. of Data Points	Attributes	Rho	Tau	G
2155	GDP per capita v/s GDP	0.6841	0.2798	0.8284
775	GDP per capita v/s Passenger cars	0.7567	0.07312	0.98
775	Passenger cars v/s urban population	0.40167	0.2781	0.4032
1836	GDP per capita v/s Internet users	0.8396	0.12322	0.839
1834	Mobile subscribers v/s Internet Users	0.367	0.256	0.3672
2038	Mortality, under-5 v/s birth rate	0.5974	0.4277	0.5977
2299	Total population v/s Urban population	0.5065	0.358	0.5069
1967	Life expectancy v/s Health exp. per capita	0.8491	0.675	0.8897

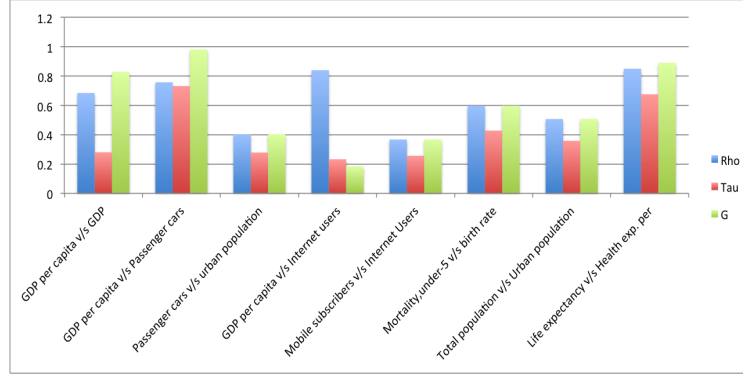


Figure 6.11: Graphical representation of comparison of correlation coefficients on the World Bank dataset

6.3 Evaluation of GBAC framework

The general correlation coefficient (G) proposed and validated in the previous section has been used to produce a correlation matrix, which forms the basis of clustering using AGNES algorithm. We have selected Mtcars (Motor trend car road tests) and USArrests (Violent crime rates by US State) data sets from the “free for research dataset” repository in R. These datasets have been used to test the quality of clustering using G, as they are standard datasets and have been used by many researchers to validate their hypothesis regarding clustering of data points. In addition to these data sets, we have also used a World Bank dataset to test the behavior of G in high dimensional data spaces. 23 attributes have been selected for this dataset to produce a 23×23 correlation matrix and a 23×23 dissimilarity matrix. The descriptive statistics of dissimilarity calculated in these datasets have also been given in order to get a summary of the dissimilarity matrix. These descriptive statistics are [246]:

- Minima: It gives the minimum data value of a particular data set. It may or may not be unique.
- 1st Quartile: It gives the 1st quarter mark i.e. approximately 25% of the data values are less than or equal to the first quartile.

- Median: The mid of the data is given by median. It is calculated by picking out the middle value when data points are sorted in increasing or decreasing order. It is very effective while dealing with data set containing outliers.
- Mean: It is a measure of centre of data and is calculated by the sum of data values divided by the number of data values.
- 3rd Quartile: It denotes the top one quarter of data i.e. approximately 25% data values are greater than or equal to the third quartile.
- Maxima: It gives the maximum data value of a particular data set. It may or may not be unique.

We perform clustering on the dissimilarity matrix and obtain Agglomerative Coefficient and Fowlkes-Mallows index. Along with the FM index, we have also calculated E_{FM} and V_{FM} for the relevant expectancy and variance under the null hypothesis of no relation. In order to demonstrate the results and datasets more clearly, we have used some plots and visual representations. The datasets have been represented using heat maps. A heat map contains elements of a data matrix, where individual values contained in the matrix are represented as colors. Correlation plots have been used to demonstrate the values of correlation coefficient G . It also describes in which region of values ranging from -1 to 1 does a particular value falls, distinguished by different colors. A banner has been used to show the output of AGNES and to calculate the Agglomerative Coefficient. Dendrograms has also been included to illustrate the arrangement of the clusters produced by hierarchical clustering. Bk plots have been used to calculate FM index for a series of k cuts for two dendrograms. A Bk plot is simply a scatter plot of B_k versus k . This plot helps in identifying the similarity between two dendrograms at different levels of k (number of clusters). It contains a dot and dash line, giving performance of a benchmark classification, a solid red line giving performance of G

based hierarchical clustering, and a dotted line demonstrating the performance of standard hierarchical clustering.

6.3.1 Performance analysis

We discuss the performance of GBAC framework first on the standard datasets then on the real time high dimensional World Bank dataset.

- USArrests dataset: Figure 6.12 illustrates the heat map for USArrests data set and Figure 6.13 demonstrates the correlation plot containing correlation matrix obtained by applying G to the USArrests dataset. The AGNES banner and Dendrogram for the GBAC are given in Figure 6.14 and Figure 6.15 respectively. Table 6.4 gives the description statistics of the dissimilarity matrix of this data set. On analysing the bk plots we can conclude that the performance of G based agglomerative clustering demonstrated by the solid red line is better than all the traditional hierarchical clusterings with single, average and complete linkages as shown in Figure 6.16. The dotted line represents the standard clusterings and the dot and dashed line shows the performance of the benchmark clustering. GBAC performs closer to the benchmark clustering as compared to the traditional hierarchical clustering approaches.

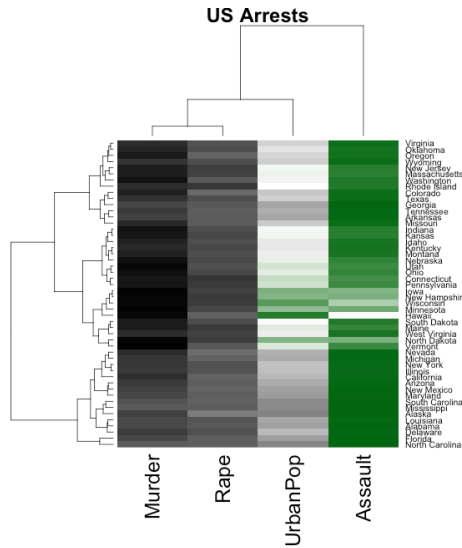


Figure 6.12: Heat map for USArrests dataset

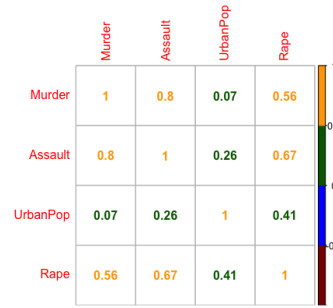


Figure 6.13: Correlation plot using G

Table 6.4: Descriptive statistics for USArrests dissimilarity matrix

Min.	1st Qu.	Median	Mean	3rd Qu.	Max
0.262	0.9432	1.22	1.42	1.718	4.047

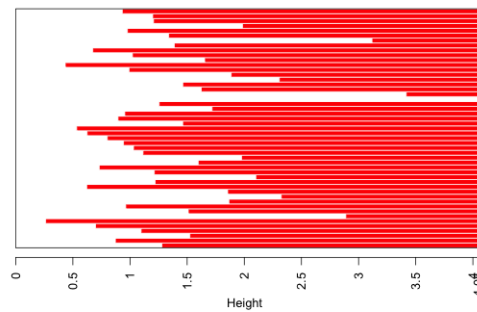


Figure 6.14: Banner for AGNES on USArrests dataset

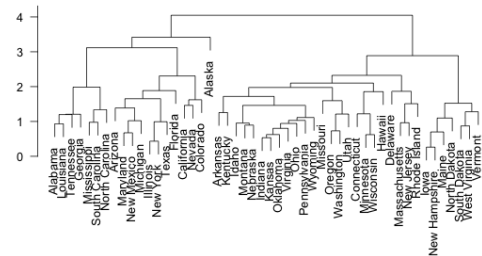


Figure 6.15: Dendrogram for USArrests dataset

- Mtcars dataset: Figure 6.17 illustrates the heat map for mtcars dataset and Figure 6.18 demonstrates the correlation plot containing correlation matrix obtained by applying G to this dataset. The AGNES banner and Dendrogram for the GBAC are given in Figure 6.19 and Figure 6.20 respectively. Table 6.5 gives the description statistics of the dissimilarity matrix of this data set. The results demonstrated by bk plots using

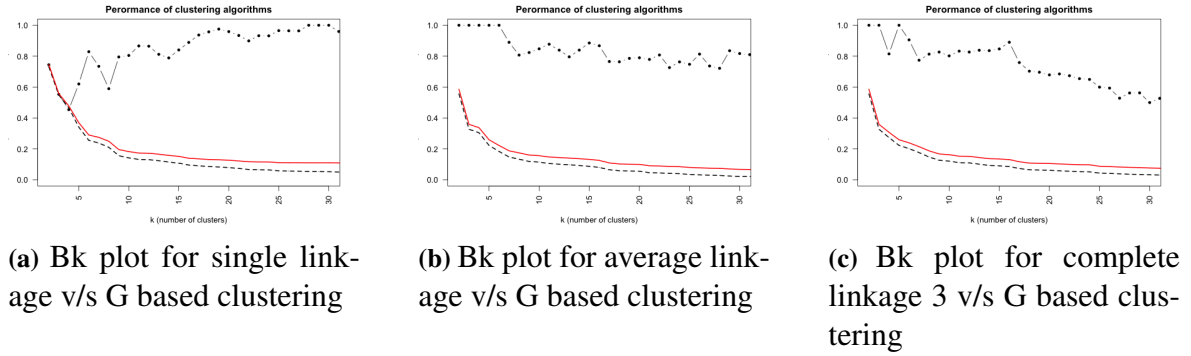


Figure 6.16: Bk plots of Hierarchical clustering based on G v/s standard hierarchical clustering for USArrests dataset

this data set are similar to the results obtained by USArrests data set and can be seen in Figure6.21. Again it is observed that GBAC performs better than the traditional hierarchical clustering approaches.

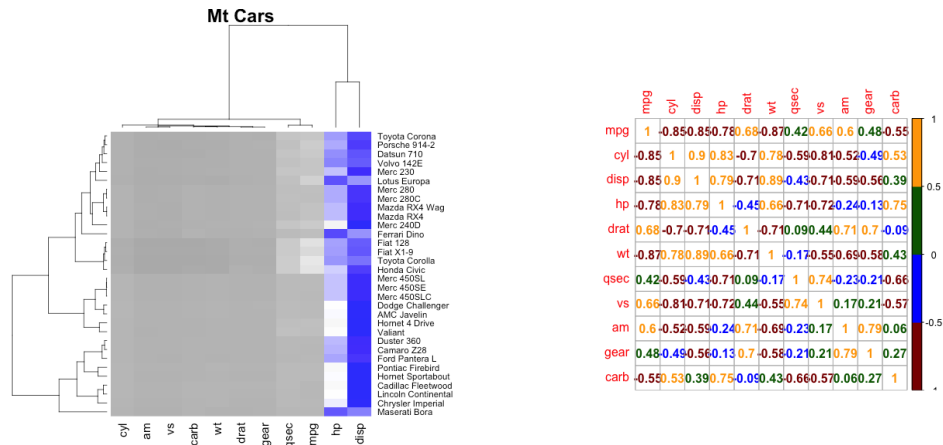


Figure 6.17: Heat map for Mt cars data set

Figure 6.18: Correlation plot using G

Table 6.5: Descriptive statistics for dissimilarity matrix of Mtcars dataset

Min.	1st Qu.	Median	Mean	3rd Qu.	Max
0.371	1.079	2.192	2.276	3.298	6.385

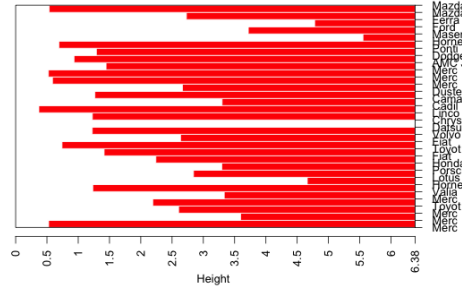


Figure 6.19: Banner for AGNES on Mtcars

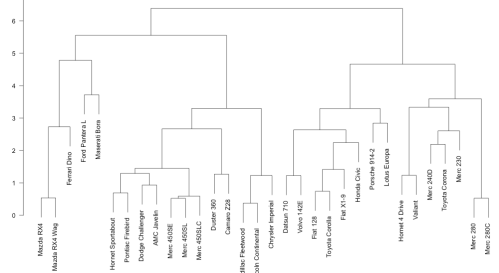
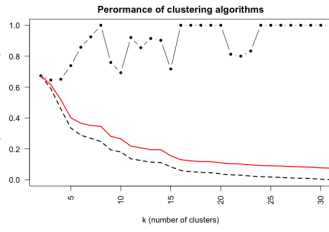
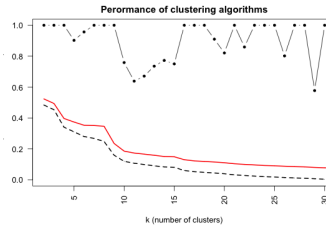


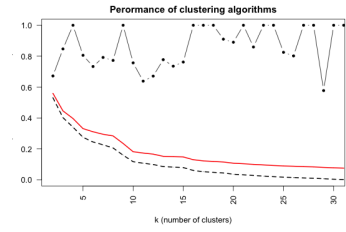
Figure 6.20: Dendrogram for data set Mtcars



(a) Bk plot for single linkage v/s G based clustering



(b) Bk plot for average linkage v/s G based clustering



(c) Bk plot for complete linkage v/s G based clustering

Figure 6.21: Bk plots of Hierarchical clustering based on G v/s standard hierarchical clustering for Mtcars dataset

- World Bank dataset: For the purpose of analysis, we have selected 23 attributes that are listed in Table 6.7 along with the correlation and dissimilarity matrices of these attributes. Table 6.6 gives the description statistics of the dissimilarity matrix of this data set. The AGNES banner and dendrogram for the G based agglomerative clustering for this data set are given in Figure 6.22 and Figure 6.23 respectively.

Table 6.6: Descriptive statistics for dissimilarity matrix of World Bank dataset

Min.	1st Qu.	Median	Mean	3rd Qu.	Max
1.010	2.562	3.518	3.936	4.781	22.530

Table 6.7: Correlation matrix and dissimilarity matrix with attributes of World Bank dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
A	1																						
B	0.8244	1																					
C	0.731	0.812	1																				
D	0.8264	0.8359	0.7715	1																			
E	0.9117	0.9249	0.8831	0.9496	1																		
F	0.516	0.61317	0.6114	0.5154	0.5892	1																	
G	-0.468	-0.4223	-0.4722	-0.4611	-0.4973	-0.06806	1																
H	-0.8471	0.189	-0.7758	-0.8211	-0.8795	-0.4173	0.5981	1															
I	-0.7329	-0.4569	-0.5291	-0.72	-0.7646	-0.291	0.5	0.7989	1														
J	0.2165	-0.1878	-0.1714	0.1741	0.1983	0.1875	-0.19612	-0.1514	-0.1514	1													
K	-0.3669	-0.4539	-0.52627	-0.3993	-0.4617	-0.1099	0.4503	0.5242	0.5122	-0.0722	1												
L	-0.2335	-0.1923	-0.1749	-0.2514	-0.2352	0.1326	0.2028	0.3371	0.33714	0.10485	0.4916	1											
M	0.8167	-0.766	-0.7719	-0.8162	-0.8611	-0.4783	0.4898	0.8595	0.8595	-0.09904	0.48372	0.30005	1										
N	0.6976	0.71796	0.6824	0.5781	0.7126	0.605	-0.30354	-0.5839	-0.5839	0.26148	-0.4324	-0.0478	-0.652	1									
O	-0.7997	-0.7349	-0.7417	-0.765	-0.8163	-0.4763	0.47075	0.8226	0.82264	-0.1245	0.48192	0.27616	0.9739	-0.6887	1								
P	0.4284	0.3154	0.39172	0.29439	0.3771	0.241	-0.1791	-0.3794	-0.3794	-0.06196	-0.2417	-0.18444	-0.3858	0.5216	-0.3908	1							
Q	0.18515	0.14299	0.21657	0.1218	0.1414	0.361	0.03827	-0.1759	-0.1759	-0.07435	-0.03576	-0.1311	-0.2169	0.14478	-0.2323	0.2077	1						
R	-0.5611	-0.5974	-0.64641	-0.6137	-0.648	-0.3039	0.37388	0.5976	0.5976	-0.07435	-0.03576	-0.1311	-0.2169	0.14478	-0.2323	0.2077	-0.1158	1					
S	-0.3237	-0.3712	-0.3522	-0.3344	-0.3771	-0.236	0.1832	0.3045	0.3045	-0.04346	0.26745	0.07823	0.2723	-0.3891	0.2532	-0.2461	0.02989	0.5214	1				
T	-0.7241	-0.6984	-0.71267	-0.7418	-0.782	-0.3622	0.4851	0.7796	0.7796	-0.05286	0.4634	0.2828	0.75245	-0.5206	0.7266	-0.4209	-0.1967	0.6397	0.3711	1			
U	0.19	0.2756	0.1534	0.1328	0.1842	0.3577	-0.02451	-0.08096	-0.08096	0.2194	0.01821	0.1739	-0.07786	0.2652	-0.08984	0.0183	-0.06686	0.04528	-0.1011	-0.05766	1		
V	0.7613	0.7831	0.7267	0.9161	0.875	0.5431	-0.4104	-0.7486	-0.7486	0.12717	-0.3191	-0.1932	-0.7258	0.6992	-0.6626	0.3671	0.148	-0.5747	-0.31268	-0.6739	0.155	1	
W	0.18364	0.1518	0.1351	-0.17127	0.11365	0.1789	0.09607	0.04506	0.04506	0.1547	0.06097	0.2781	0.0086	0.1686	0.0352	0.12477	-0.1325	0.0441	-0.04618	0.11373	0.2803	-0.04342	1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
A	0	0.1756	0.269	0.1736	0.0883	0.484	1.468	1.8471	1.7329	0.7835	1.3669	1.2335	0.1833	0.3024	1.7997	0.5716	0.81485	1.5611	1.3237	1.7241	0.81	0.2387	0.81636
B	0.1756	0	0.188	0.1641	0.0751	0.38683	1.4223	0.811	1.4569	1.1878	1.4539	1.1923	1.766	0.28204	1.7349	0.6846	0.85701	1.5974	1.3712	1.6984	0.7244	0.2169	0.8482
C	0.269	0.188	0	0.2285	0.1169	0.3886	1.4722	1.7758	1.5291	1.1714	1.52627	1.1749	1.7719	0.3176	1.7417	0.60828	0.78343	1.64641	1.3522	1.71267	0.8466	0.2733	0.8649
D	0.1736	0.1641	0.2285	0	0.0504	0.4846	1.4611	1.8211	1.72	0.8259	1.3993	1.2514	1.8162	0.4219	1.765	0.70561	0.8782	1.6137	1.3344	1.7418	0.8672	0.0839	1.17127
E	0.0883	0.0751	0.1169	0.0504	0	0.4108	1.4973	1.8795	1.7646	0.8017	1.4617	1.2352	1.8611	0.2874	1.8163	0.6229	0.8586	1.648	1.3771	1.782	0.8158	0.125	0.88635
F	0.484	0.38683	0.3886	0.4846	0.4108	0	1.06806	1.4173	1.291	0.8125	1.1099	0.8674	1.4783	0.395	1.4763	0.759	0.639	1.3039	1.236	1.3622	0.6423	0.4569	0.8211
G	1.468	1.4223	1.4722	1.4611	1.4973	1.06806	0	0.4019	0.4637	1.19612	0.5497	0.7972	0.5102	1.30354	0.52925	1.1791	0.96173	0.62612	0.8168	0.5149	1.02451	1.4104	0.90933
H	1.8471	0.811	1.7758	1.8211	1.8795	1.4173	0.4019	0	0.2011	1.1514	0.4758	0.6629	0.1405	1.5839	0.1774	1.3794	1.1759	0.4024	0.6955	0.2204	1.08096	1.7486	0.95494
I	1.7329	1.4569	1.5291	1.72	1.7646	1.291	0.4637	0.2011	0	1.1514	0.4878	0.66286	0.1405	1.5839	0.17736	1.3794	1.1759	0.4024	0.6955	0.2204	1.08096	1.7486	0.95494
J	0.7835	1.1878	1.1714	0.8259	0.8017	0.8125	1.19612	1.1514	1.1514	0	1.0722	0.89515	1.09904	0.73852	1.1245	1.06196	1.07435	1.0776	1.04346	1.05286	0.7806	0.87283	0.8453
K	1.3669	1.4539	1.52627	1.3993	1.4617	1.1099	0.5497	0.4758	0.4878	1.0722	0	0.50831	0.51628	1.4324	0.51808	1.2417	1.03576	0.56387	0.73255	0.5366	0.98179	1.3191	0.93603
L	1.2335	1.1923	1.1749	1.2514	1.2352	0.8674	0.7972	0.6629	0.66286	0.89515	0.5084	0	0.69995	1.0478	0.72384	1.18444	1.1311	0.7381	0.92177	0.7172	0.8261	1.1932	0.7219
M	0.1833	1.766	1.7719	1.8162	1.8611	1.4783	0.5102	0.1405	0.1405	1.09904	0.51628	0.69995	0	1.652	0.0261	1.3858	1.2169	0.4167	0.7277	0.24755	1.07786	1.7258	0.9314
N	0.3024	0.28204	0.3176	0.4219	0.2874	0.395	1.30354	1.5839	1.5839	0.73852	1.4324	1.0478	1.652	0	1.6887	0.4784	0.85522	1.4804	1.3891	1.5206	0.7348	0.3908	0.8314
O	1.7997	1.7349	1.7417	1.765	1.8163	1.4763	0.52925	0.1774	0.17736	1.1245	0.51808	0.72384	0.0261	1.6887	0	1.3908	1.2323	0.4515	0.7468	0.2734	1.08984	1.6626	0.9648
P	0.5716	0.6846	0.60828	0.70561	0.6229	0.759	1.1791	1.3794	1.3794	1.06196	1.2417	1.18444	1.3858	0.4784	1.3908	0	0.7923	1.4173	1.2461	1.4209	0.9817	0.6329	0.87523
Q	0.81485	0.85701	0.78343	0.8782	0.8586	0.639	0.96173	1.1759	1.1759	1.07435	1.03576	1.1311	1.2169	0.85522	1.2323	0.7923	0	1.1158	0.97011	1.1967	1.06686	0.852	1.1325
R	1.5611	1.5974	1.64641	1.6137	1.648	1.3039	0.62612	0.4024	0.4024	1.0776	0.56387	0.7381	0.4167	1.4804	0.4515	1.4173	1.1158	0	0.6786	0.3603	0.95472	1.5747	0.9559
S	1.3237	1.3712	1.3522	1.3344	1.3771	1.236	0.8168	0.6955	0.6955	1.04346	0.73255	0.92177	0.7277	1.3891	0.7468	1.2461	0.97011	0.6786	0	0.6289	1.1011	1.31268	1.04618
T	1.7241	1.6984	1.71267	1.7418	1.782	1.3622	0.5149	0.2204	0.2204	1.05286	0.5366	0.7172	0.24755	1.5206	0.2734	1.4209	1.1967	0.3603	0.6289	0	1.05766	1.6739	0.88627
U	0.81	0.7244	0.8466	0.8672	0.8158	0.6423	1.02451	1.08096	1.08096	0.7806	0.98179	0.8261	1.07786	0.7348	1.08984	0.9817	1.96686	0.95472	1.1011	1.05766	0	0.845	0.7197
V	0.2387	0.2169	0.2733	0.0839	0.125	0.4569	1.4104	1.7486	1.7486	0.87283	1.3191	1.1932	1.7258	0.3908	1.6626	0.6329	0.852	1.5747	1.31268	1.6739	0.845	0	1.04342
W	0.81636	0.8482	0.8649	1.17127	0.88635	0.8211	0.90933	0.95494	0.95494	0.8453	0.93603	0.7219	0.9314	0.8314	0.9648	0.87523	1.1325	0.9559	1.04618	0.88627	0.7197	1.04342	0

On analyzing the Bk plots for this high dimensional data set given in Figure 6.24, we can see that GBAC framework fares better than all the three traditional hierarchical clusterings. This means that for a particular value of K, in high dimensional data, the proposed clustering scheme performs better than the traditional clustering techniques.

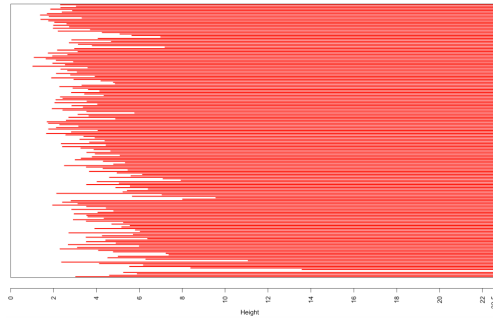


Figure 6.22: Banner for AGNES on World Bank dataset

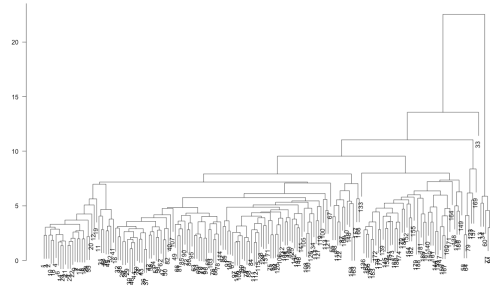


Figure 6.23: Dendrogram for World Bank dataset

6.3.2 Cluster quality analysis

The Agglomerative Coefficients of GBAC, for all the three datasets is given in Table 6.8. For the standard datasets it is 0.7376 and 0.7826, which shows high quality of clustering. It is even higher for the high-dimensional data set at a value of 0.8581.

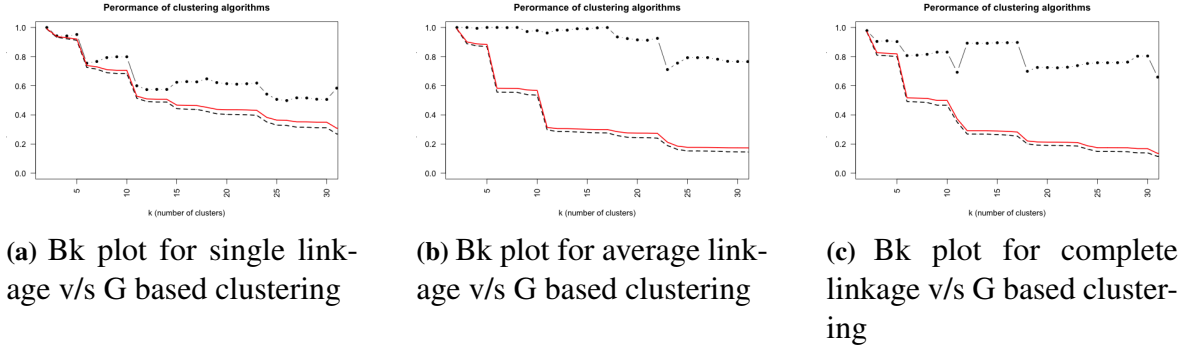


Figure 6.24: Bk plots of Hierarchical clustering based on G v/s standard hierarchical clustering for World Bank dataset

This shows that GBAC clusters are of high quality with limited to no outliers. This shows that G handles outliers efficiently and is not sensitive to them. The FM index

Table 6.8: Agglomerative Coefficient table

Data Set	Attributes		Agglomerative Coefficient
US Arrests	Crime Criteria	US States	0.7376088
	4	50	
Mt Cars	Car attributes	Car family	0.7826078
	11	32	
World Bank	Parameters	Countries	0.8581071
	23	188	

for given values of k (no. of clusters) are given in Table 6.9. These values show that for both standard and high dimensional data sets, the FM index for G based clustering is 1. This means that G shows perfect clustering. This is comparable to average and single linkage hierarchical clustering and is better than the complete linkage hierarchical clustering. It should be noted that these perfect values are obtained by inputting the perfect values of k obtained from the dendrograms of all the three datasets. The E_{FM} (Expectation of FM index) and V_{FM} (Variance of the FM index) are also mentioned.

The execution time and memory requirements taken by GBAC on all the three datasets used above are given in Table 6.10. The user time relates to the execution of the code, the system time relates to CPU time, and the elapsed time is the difference in times since you

Table 6.9: FM Index table

Data Set	Value of k	Clustering Methodology	FM Index		
			Index	V_FM	E_FM
Standard Data Sets					
US Arrests	3	Single	1	0.000269904	0.882449
		Average	1	0.3044898	0.000402221
		GBAC	1	0.000465811	0.2457143
		Complete	0.8147512	0.2735279	0.000425977
Mt Cars	4	Single	1	0.001284565	0.7681452
		Average	1	0.000603349	0.453629
		GBAC	1	0.000603349	0.453629
		Complete	0.8468545	0.000704428	0.4023428
Real data Set					
World Bank development indicators	5	Single	1	9.37E-06	0.9577881
		Average	1	7.57E-05	0.8686995
		GBAC.	1	7.57E-05	0.8686995
		Complete	1	2.15E-04	0.7371146

Table 6.10: Execution time and memory requirement of datasets

Data set	Execution time (in seconds)			Memory requirement
	User	System	Elapsed	
USArrests	0.317	0.041	0.372	62.1 MB
Mtcars	0.403	0.248	0.693	85.4 MB
World Bank	0.195	0.029	0.237	85.4 MB

started the stopwatch (and will be equal to the sum of user and system times if the chunk of code was run altogether).

6.4 Evaluation of LGBACC framework

The proposed approach combines dimensionality reduction with cluster analysis. The focus of this approach is to improve quality of the clusters obtained. The framework proposed in the previous chapter has been implemented and tested using real world datasets. The datasets have been selected from the “free for research dataset repository” in the R programming language, and some other real world data sets used by researchers working in the field of correlation clustering and dimensionality reduction. Graphical results have been provided

for each and every phase of the framework. The quality of the clusters have been tested using indices like Agglomerative Coefficient and FM index. In order to demonstrate the results and data sets more clearly, we have used some plots and visual representations.

6.4.1 Performance analysis

The results obtained, after running the steps of the LGBACC framework described in the previous chapter, on the real time data sets are as follows:

PCA and Graph analysis phase

After the data has been imported into the computing environment, the next step is to perform PCA for dimensionality reduction, and visualize the data set using graph theory. The analysis performed in this phase can be seen visually. Tools such as bar graphs, pie charts and graphs have been used to visualize the result of this phase. Bar plots have been used to compute principal components and represent the percentage of information retained by each dimension. Individual variables have been represented using variables factor map.

The statistics for each dimension of the PCA are indicative of the information contained in data. The parameters used to depict such are eigenvalue, variance percent and cumulative variance percent. The statistics for all the datasets are given from Table 6.11 - 6.15. The visualization of PCA has been done using bar graphs and pie charts and the results for all datasets have been given in Figures 6.25 - 6.28. The percentage of information retained by each dimension and individual variables have been visualized clearly using these tools.

The analysis of data visualized using graphs have been presented from Figure 6.29 - 6.33. These graphs have been obtained by generating local similarity matrix and affinity matrix on the local covariance matrix from the data synthesis phase. The most important step of this phase is to apply graph Laplacian on the affinity matrix in order to calculate Eigen vector

Table 6.11: PCA statistics for Wages dataset

	Eigenvalue	Variance %	Cumulative variance %
Dim.1	2.29E+00	2.09E+01	20.86061
Dim.2	1.68E+00	1.53E+01	36.15663
Dim.3	1.60E+00	1.45E+01	50.66143
Dim.4	1.07E+00	9.74E+00	60.40585
Dim.5	9.62E-01	8.75E+00	69.15367
Dim.6	8.68E-01	7.89E+00	77.04632
Dim.7	7.79E-01	7.08E+00	84.12465
Dim.8	7.35E-01	6.68E+00	90.80292
Dim.9	5.56E-01	5.05E+00	95.85298
Dim.10	4.56E-01	4.15E+00	99.9991
Dim.11	9.94E-05	9.03E-04	100

matrix. The graphs generated clearly show the relationships between different attributes and data points of each data set. The purpose of this phase is to clearly visualize data using the plots generated after performing PCA. Both these tools are used to better the quality of the clusters produced in the next phase.

Table 6.12: PCA statistics for El Nino dataset

	Eigenvalue	Variance %	Cumulative variance %
Dim.1	3.05E+00	3.05E+01	30.53924
Dim.2	1.46E+00	1.46E+01	45.09338
Dim.3	1.26E+00	1.26E+01	57.6809
Dim.4	1.01E+00	1.01E+01	67.78218
Dim.5	9.97E-01	9.97E+00	77.75592
Dim.6	8.25E-01	8.25E+00	86.01047
Dim.7	6.34E-01	6.34E+00	92.34746
Dim.8	5.07E-01	5.07E+00	97.41928
Dim.9	2.58E-01	2.58E+00	100
Dim.10	1.77E-24	1.77E-23	100

Table 6.13: PCA statistics for Mtcars dataset

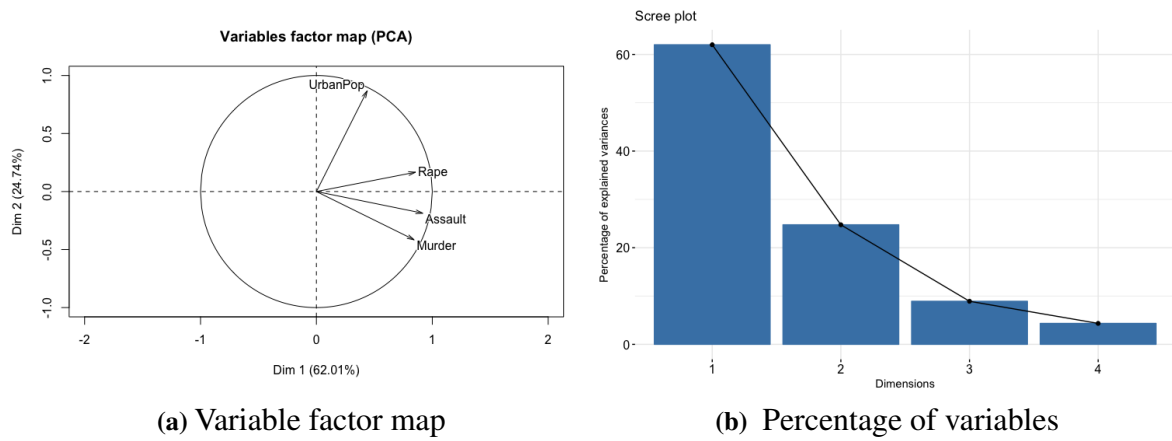
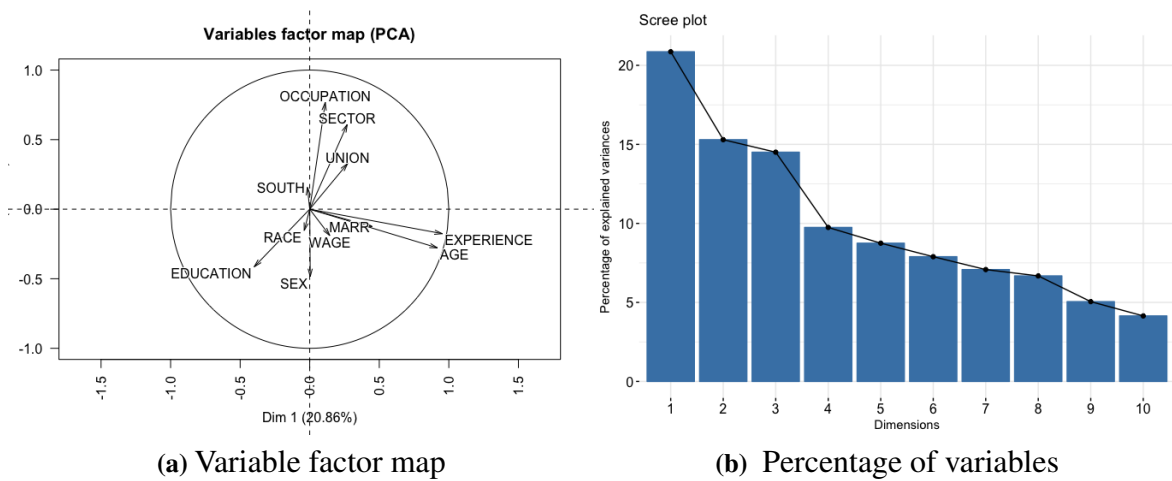
	Eigenvalue	Variance %	Cumulative variance %
Dim.1	6.60840025	60.0763659	60.07637
Dim.2	2.65046789	24.0951627	84.17153
Dim.3	0.62719727	5.7017934	89.87332
Dim.4	0.26959744	2.4508858	92.32421
Dim.5	0.2234511	2.0313737	94.35558
Dim.6	0.21159612	1.9236011	96.27918
Dim.7	0.13526199	1.2296544	97.50884
Dim.8	0.12290143	1.1172858	98.62612
Dim.9	0.07704665	0.7004241	99.32655
Dim.10	0.05203544	0.4730495	99.7996
Dim.11	0.02204441	0.2004037	100

Table 6.14: PCA statistics for World Bank dataset

	Eigenvalue	Variance %	Cumulative variance %
Dim.1	5.58742723	31.04126242	31.04126
Dim.2	3.78608178	21.03378765	52.07505
Dim.3	3.54103882	19.67243789	71.74749
Dim.4	1.53508985	8.528276952	80.27576
Dim.5	0.88648678	4.92492655	85.20069
Dim.6	0.6584518	3.658065539	88.85876
Dim.7	0.51826728	2.879262652	91.73802
Dim.8	0.34726257	1.929236503	93.66726
Dim.9	0.32230164	1.79056466	95.45782
Dim.10	0.21246632	1.180368437	96.63819
Dim.11	0.19062247	1.059013709	97.6972
Dim.12	0.15970561	0.887253371	98.58446
Dim.13	0.11744001	0.652444515	99.2369
Dim.14	0.08742656	0.485703088	99.7226
Dim.15	0.03352692	0.186260666	99.90886
Dim.16	0.01307144	0.072619123	99.98148
Dim.17	0.00316945	0.017608057	99.99909
Dim.18	0.00016348	0.000908222	100

Table 6.15: PCA statistics for USArrests dataset

	Eigenvalue	Variance %	Cumulative variance %
Dim.1	2.4802416	62.006039	62.00604
Dim.2	0.9897652	24.744129	86.75017
Dim.3	0.3565632	8.91408	95.66425
Dim.4	0.1734301	4.335752	100

**Figure 6.25:** PCA for USArrests dataset**Figure 6.26:** PCA for Wages dataset

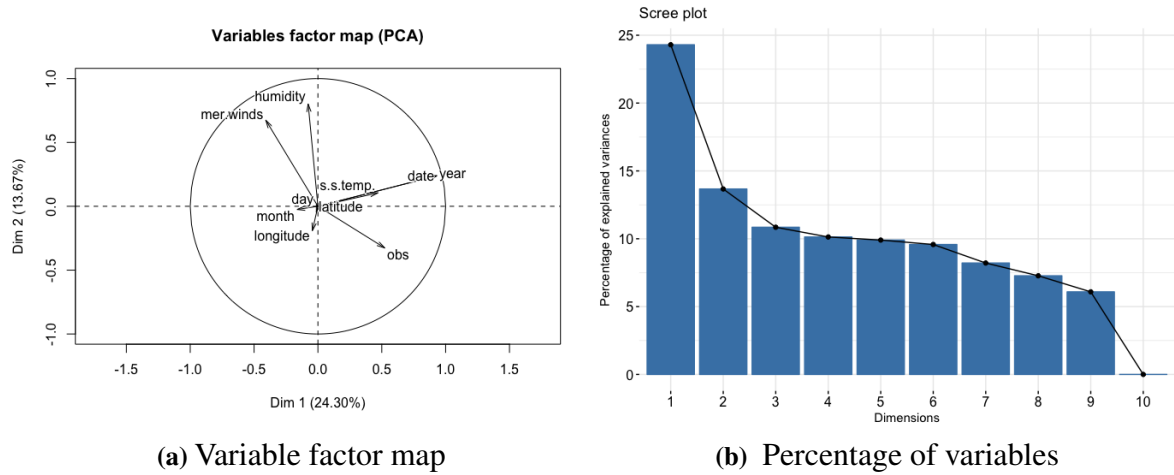


Figure 6.27: PCA for El Nino dataset

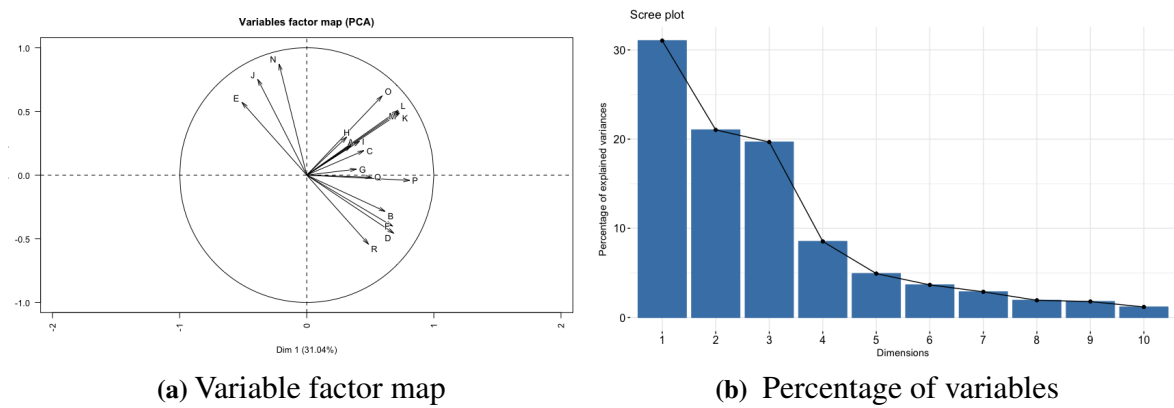


Figure 6.28: PCA for World Bank dataset

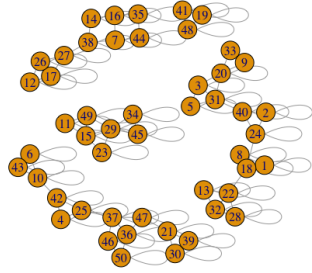


Figure 6.29: Graph for USArrests dataset

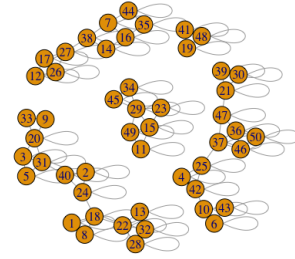


Figure 6.30: Graph for Mtcars dataset

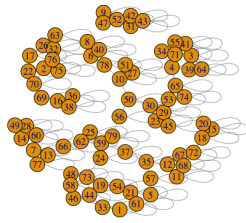


Figure 6.31: Graph for Wages dataset

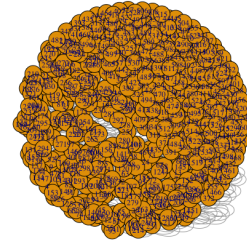


Figure 6.32: Graph for El Nino dataset

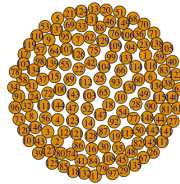


Figure 6.33: Graph for World Bank dataset

Cluster analysis

In this phase the reduced dataset is taken and data clusters are generated. The visualization of the clusters generated from this newly obtained dataset have been represented in Figure 6.34 to Figure 6.38 (for all the above mentioned datasets). Clustering on dataset S_2 obtained in the previous phase has been performed using the agglomerative hierarchical clustering

algorithm AGNES. The results have been represented in the form of factor maps, 3D model of clusters, dendrograms and cluster plots. Factor maps are formed by generating individual points colored according to the cluster they belong to. The axes in the map makes the placement of data points, in a given dimension, clear. It can be observed from the factor maps of all the datasets that they work well for low dimension as well as high dimension datasets. Lower dimensional datasets having lesser number of data points are individually visible as members of the clusters formed. In higher dimensional datasets with large number of data points, factor maps give a clear picture of density of points in a given cluster.

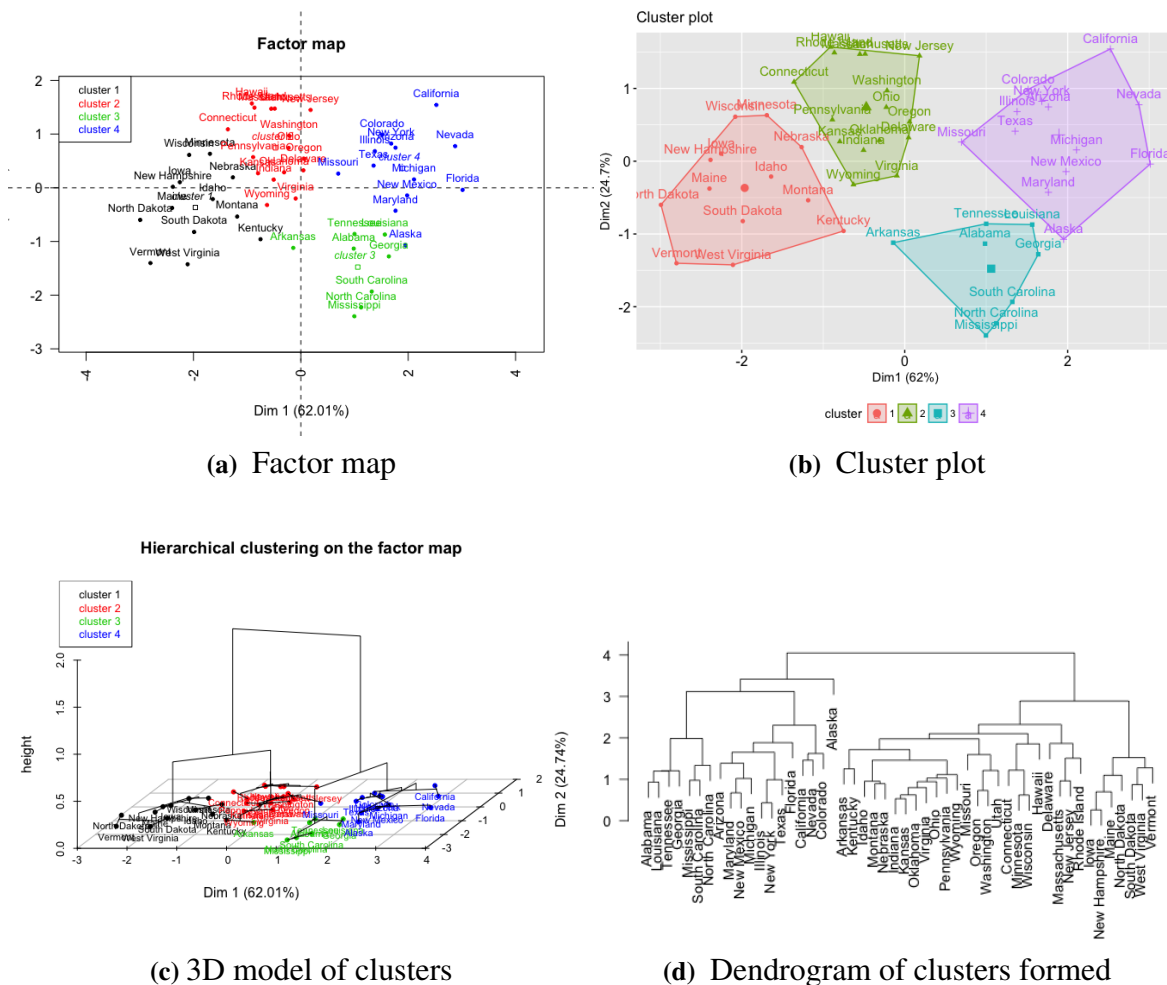


Figure 6.34: Cluster analysis of USArrests dataset

The 3D map plots the factor map in the form of a tree. This tool has been used in order

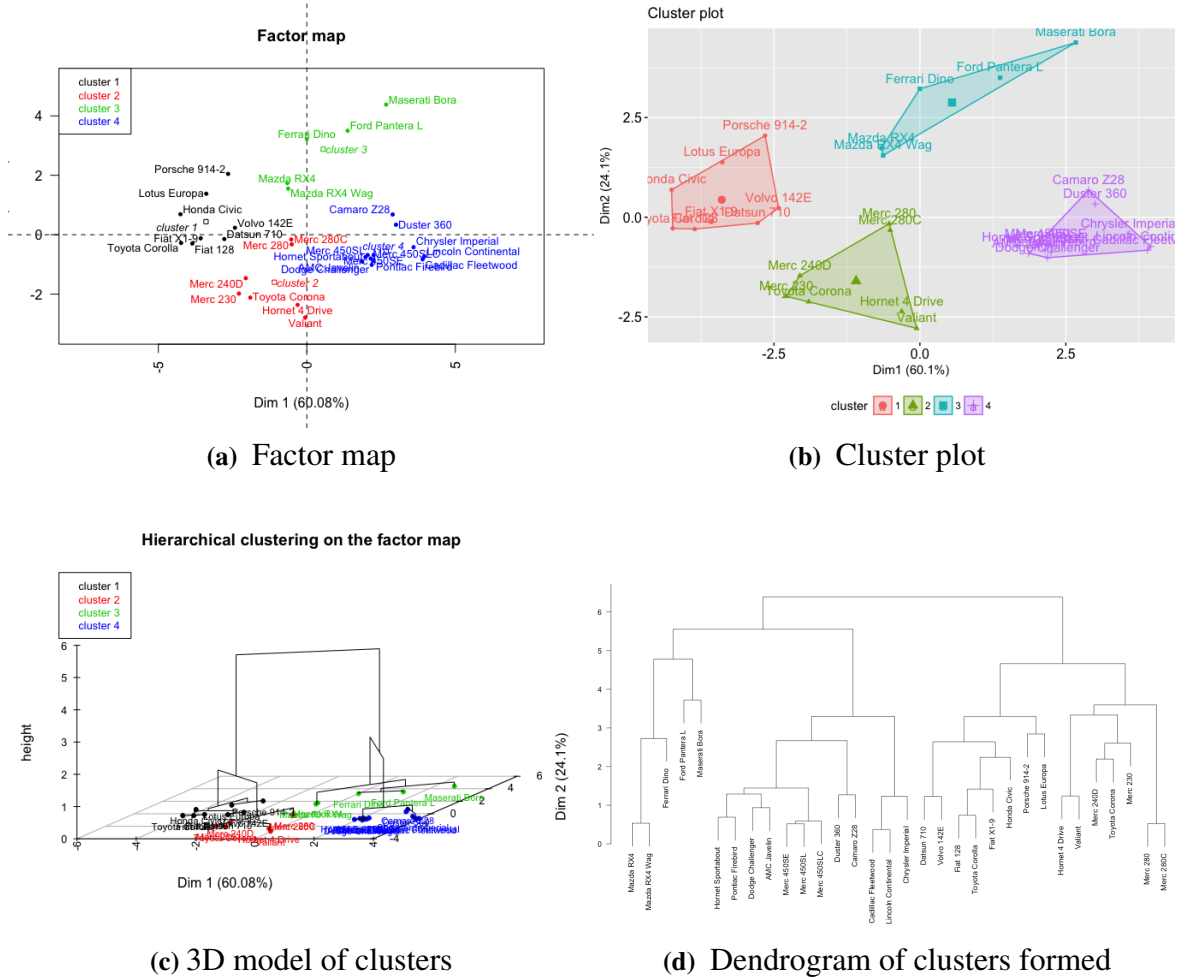


Figure 6.35: Cluster analysis of Mtcars dataset

to clearly visualize the distribution of data points in the given clusters in 3-dimensional data space. Again it works for both low and high dimensional datasets. Cluster plots give a 2-dimensional view of the clustering. Dendrogram is a classic output of any hierarchical clustering algorithm. Dendrograms in the form of a tree give clear hierarchy of the data points of all the datasets and their division into clusters. The second part of this phase is to generate graphs from the dataset S_2 in order to visualize the clusters formed from the reduced data set in the form of relationships between data points and attributes. The graphs have been presented in Figure 6.39 - 6.43. All these graphs clearly show the relationships between attributes and division into clusters, except for the graph presented for the data set El

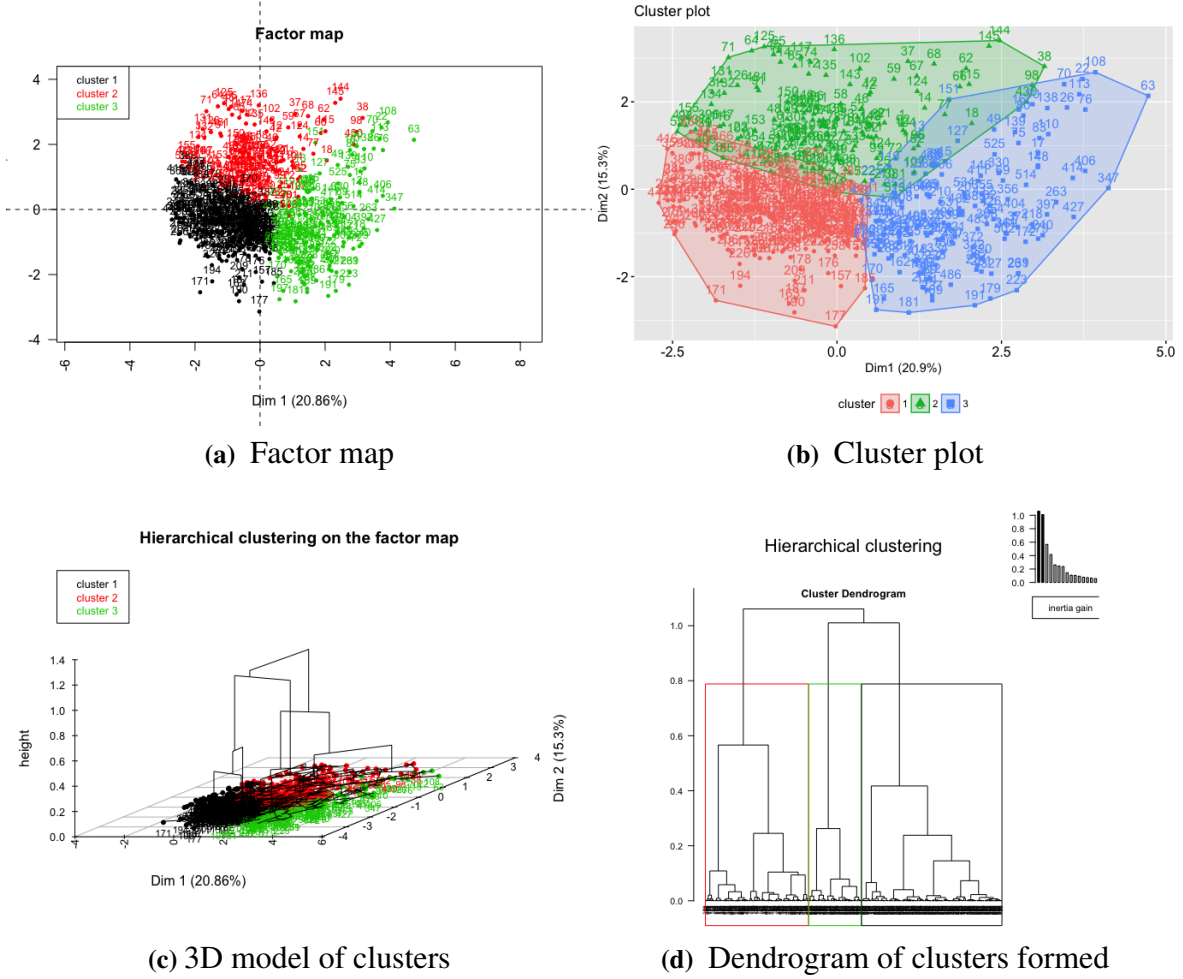
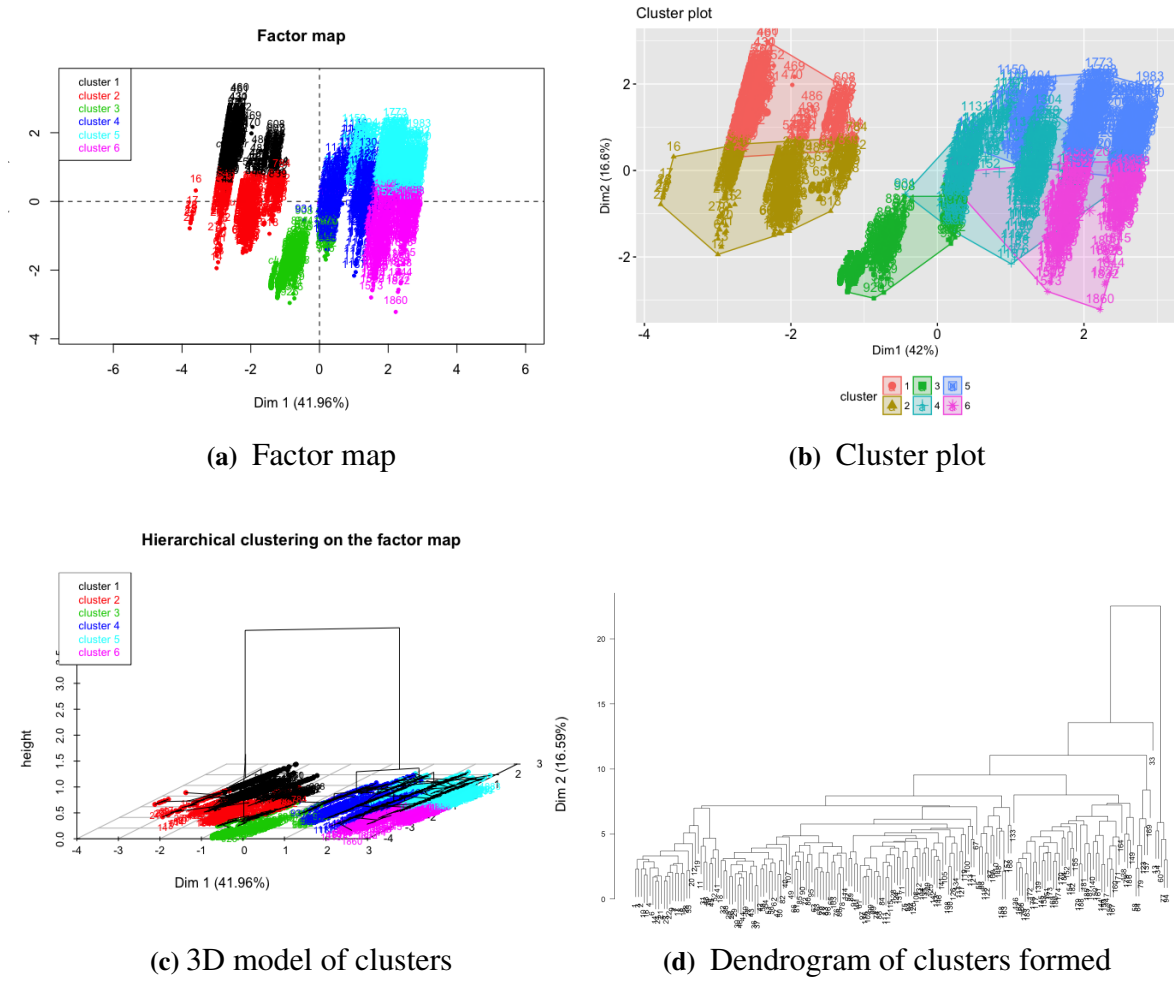


Figure 6.36: Cluster analysis of Wages dataset

Nino in Figure 6.42. This graph is cluttered and the relationships and clusters are not clearly visible. This means that 2-D graphs can be used to represent relationships in low and high dimensional datasets with less number of data points. In case of datasets with large number of data points (in this case more than 1,70,000), we need to find better tools to visualize relationships.

Cluster quality assessment

As we have applied algorithm AGNES on all the datasets in order to obtain hierarchy of clusters, we calculate Agglomerative Coefficient to indicate the strength of the clustering

**Figure 6.37:** Cluster analysis of El Nino dataset

structure. Table 6.16 contains Agglomerative Coefficients and Descriptive statistics of all the datasets.

Table 6.16: Descriptive statistics and Agglomerative Coefficient

Dataset	Agglomerative Coefficient	Min.	1st. Quad	Median	Mean	3rd Quad	Max.
USArrests	0.7574659	0.262	0.9432	1.279	1.447	1.823	4.538
Mtcars	0.7934925	0.371	1.079	2.192	2.292	3.298	6.721
Wages	0.8532717	0.0649	1.09	1.95	2.205	3.038	10.85
El Nino	0.9400893	0.1391	0.3115	0.508	0.6714	0.8094	6.527
World Bank	0.9907157	0	0.2495	0.5382	1.026	1.045	55.88

If we look at the value of Agglomerative Coefficients obtained for LGBACC, it is indicative of the fact that this approach produces high quality clusters irrespective of the dimen-

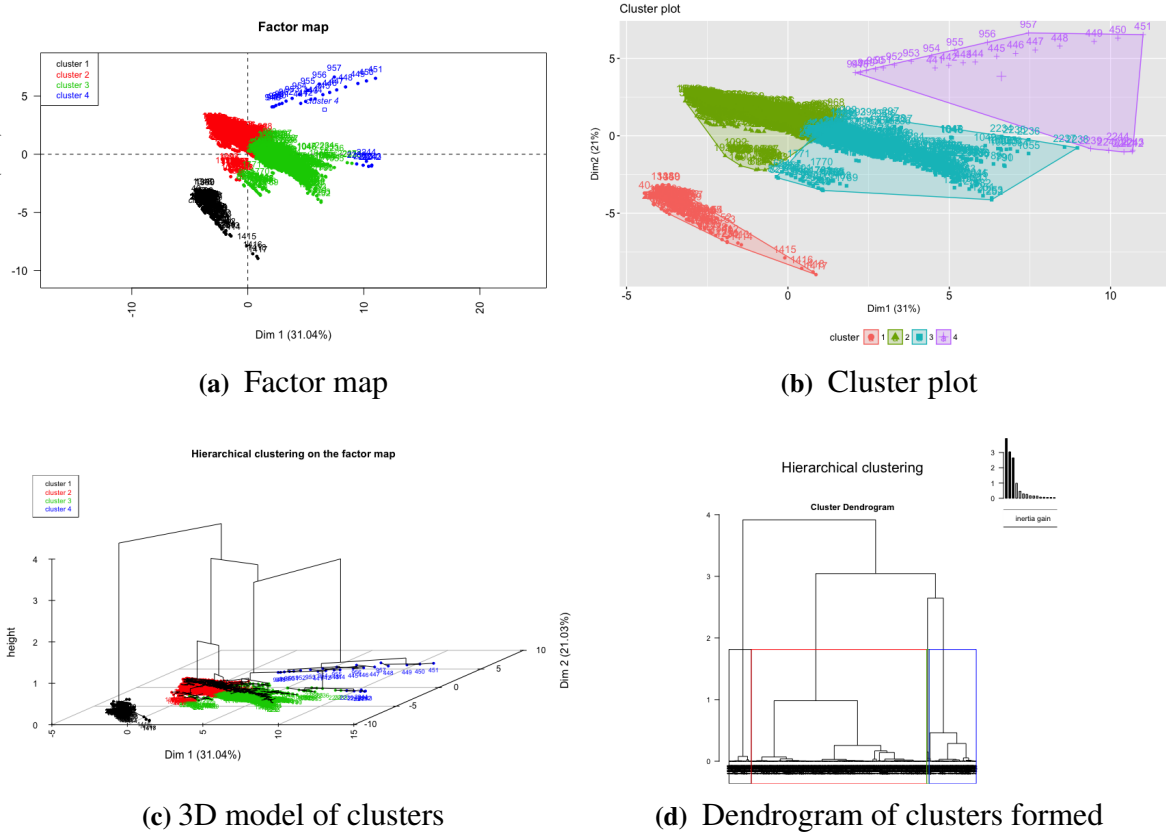


Figure 6.38: Cluster analysis of World Bank dataset

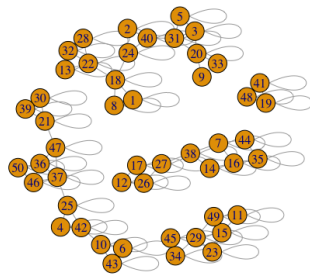


Figure 6.39: Graph representing clusters formed in USArrests dataset

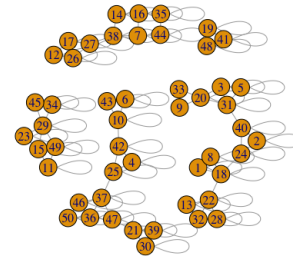


Figure 6.40: Graph representing clusters formed in Mtcars dataset

sionality of the dataset. The value of the coefficient remains high and almost touches 1 in a few cases. Another measure that we use to access the quality of the clusters generated by the approach is the FM index. The values of the index for a given values of k (no. of clusters)

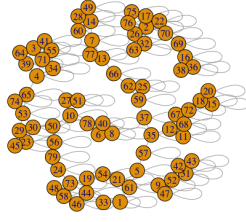


Figure 6.41: Graph representing clusters formed in Wages dataset

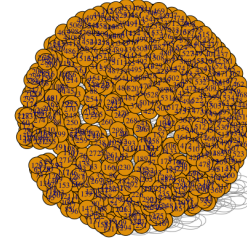


Figure 6.42: Graph representing clusters formed in El Nino dataset

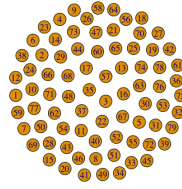


Figure 6.43: Graph representing clusters formed in World Bank dataset

are given in Table 6.17. These values show that for both low and high dimensional data sets, the FM index for LGBACC is 1 or nearing 1. This means that LGBACC shows near perfect clustering. This is comparable to average linkage hierarchical clustering and is better than the complete and single linkage methodologies. The E_{FM} (Expectation of FM index) and V_{FM} (Variance of the FM index) are also mentioned.

The execution time and memory requirements for datasets used above are given in Table 6.18. It is observed that LGBACC is not computationally expensive. The execution time and memory requirement increases slightly with an increase in the size of the data.

Bk plots have been used to calculate the FM index for a series of k cuts for two dendrograms. A Bk plot is simply a scatter plot of Bk versus k. This plot helps in identifying the similarity between two dendrograms in different levels of k (number of clusters). It contains a dot and dash line giving performance of a benchmark classification, a solid red line giv-

Table 6.17: Fowlkes-Mallow index table

Data Set	Value of K	Hierarchical clustering Methodology	Fowlkes Mallows Index		
			Index	E_FM	V_FM
USArrests	4	LGBACC	0.9523533	0.2340068	0.000479088
		Single	0.7688429	0.6784646	0.000477478
		Average	0.8555136	0.2604952	0.000441122
		Complete	0.8555136	0.2604952	0.000441122
Mtcars	4	LGBACC	0.9545903	0.3252535	0.001204995
		Single	0.766965	0.4731679	0.002596811
		Average	0.95459	0.3252535	0.001450695
		Complete	0.8555136	0.2604952	0.000441122
Wages	3	LGBACC	1	0.5081547	0.5081547
		Single	1	0.985082	4.12E-07
		Average	1	0.5438652	1.30E-05
		Complete	1	0.6692947	5.52E-05
El Nino	6	LGBACC	1	0.2052496	1.96E-06
		Single	1	0.1783011	6.59E-07
		Average	1	0.2074882	2.02E-06
		Complete	0.8723849	0.2074882	2.02E-06
World Bank	3	LGBACC	0.990538	0.9697704	3.38E-07
		Single	0.9940347	0.9847843	8.20E-08
		Average	0.9978417	0.9768331	2.20E-07
		Complete	1	0.9789206	0.9789206

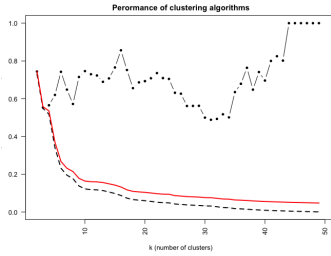
ing performance of LGBACC and a dotted line demonstrating the performance of standard hierarchical clustering. On analyzing the Bk plots we can conclude that the performance of LGBACC demonstrated by the solid red line is better than all the traditional hierarchical clusterings with single, average and complete linkages as shown in Figure 6.44 - 6.48. The dotted line represents the standard clustering and the dot and dashed line shows the performance of the benchmark clustering.

6.5 Evaluation of scalability for proposed approaches

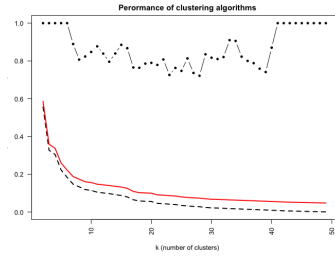
To test the working of the proposed algorithms on the basis of the stated parameters, we have performed experiments on four datasets, using SparkR (R on Spark). Spark R is an R

Table 6.18: Execution time and memory requirement of datasets

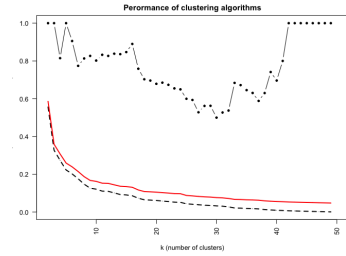
Datasets	Execution time (in seconds)			Memory requirement
	User	System	Elapsed	
US Arrests	6.829	0.361	8.134	215 MB
MtCars	6.133	0.527	6.937	276 MB
Wages	5.288	0.502	6.116	267 MB
El Nino	5.314	0.429	6.011	267 MB
World Bank	6.634	0.621	7.72	320 MB



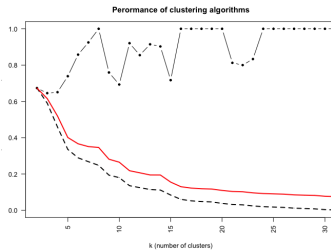
(a) Bk plot for single linkage v/s LGBACC



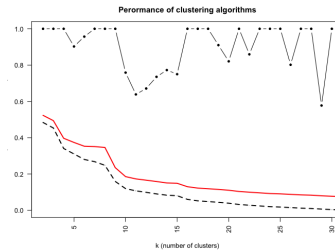
(b) Bk plot for average linkage v/s LGBACC



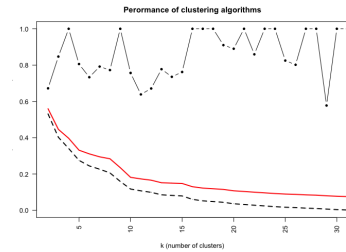
(c) Bk plot for complete linkage v/s LGBACC

Figure 6.44: Bk plots of traditional hierarchical clustering algorithms v/s LGBACC for US-Arrests dataset

(a) Bk plot for single linkage v/s LGBACC



(b) Bk plot for average linkage v/s LGBACC



(c) Bk plot for complete linkage v/s LGBACC

Figure 6.45: Bk plots of traditional hierarchical clustering algorithms v/s LGBACC for Mt-cars dataset

language package that provides a front end to use Apache Spark from R. It provides a distributed data frame implementation that supports different operations but on large datasets. It also supports distributed machine learning using MLib. We have used Amazon simple storage (S3) buckets to store our large datasets and piped the data into the SparkR environ-

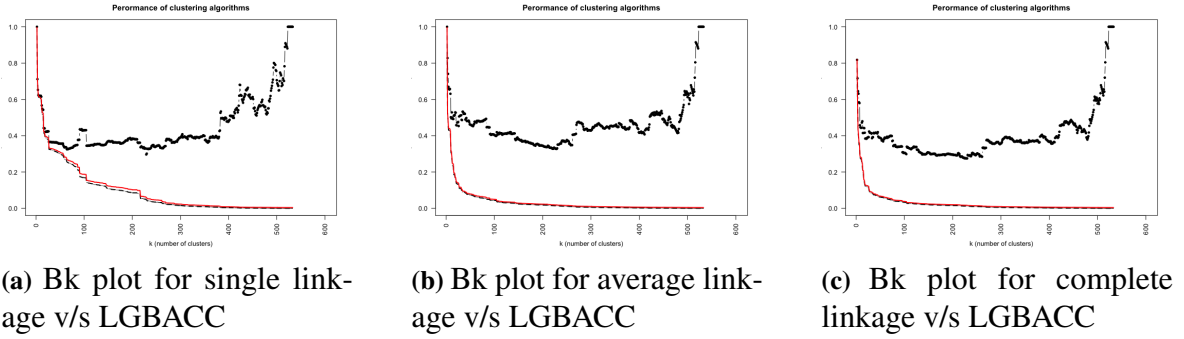


Figure 6.46: Bk plots of traditional hierarchical clustering algorithms v/s LGBACC for Wages dataset

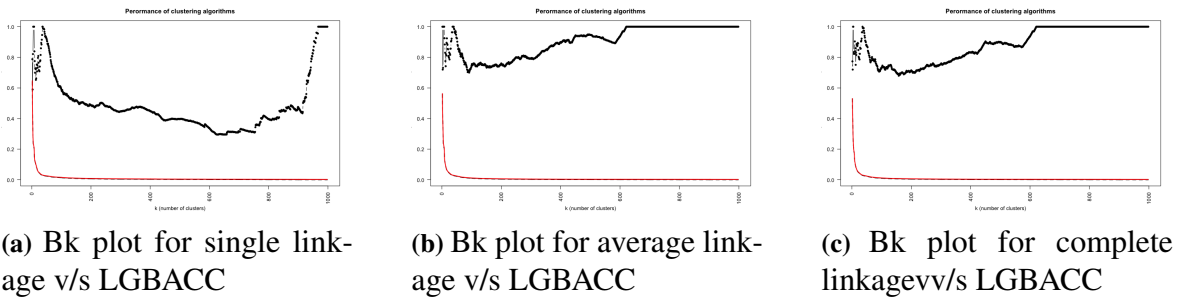


Figure 6.47: Bk plots of traditional hierarchical clustering algorithms v/s LGBACC for El Nino dataset

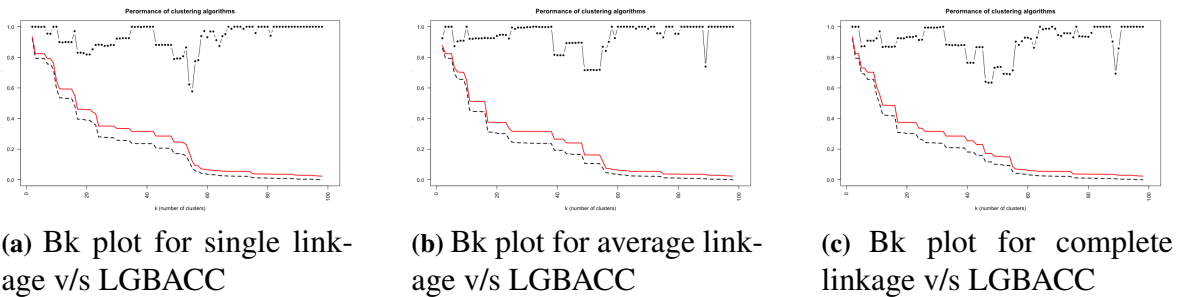


Figure 6.48: Bk plots of traditional hierarchical clustering algorithms v/s LGBACC for World Bank dataset

ment using the datasource API. The SparkR architecture used for experimentation is given in Figure 6.49.

We have used the New York yellow cabs dataset taken from http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml. This is a very large dataset with approximately 1.3 bil-

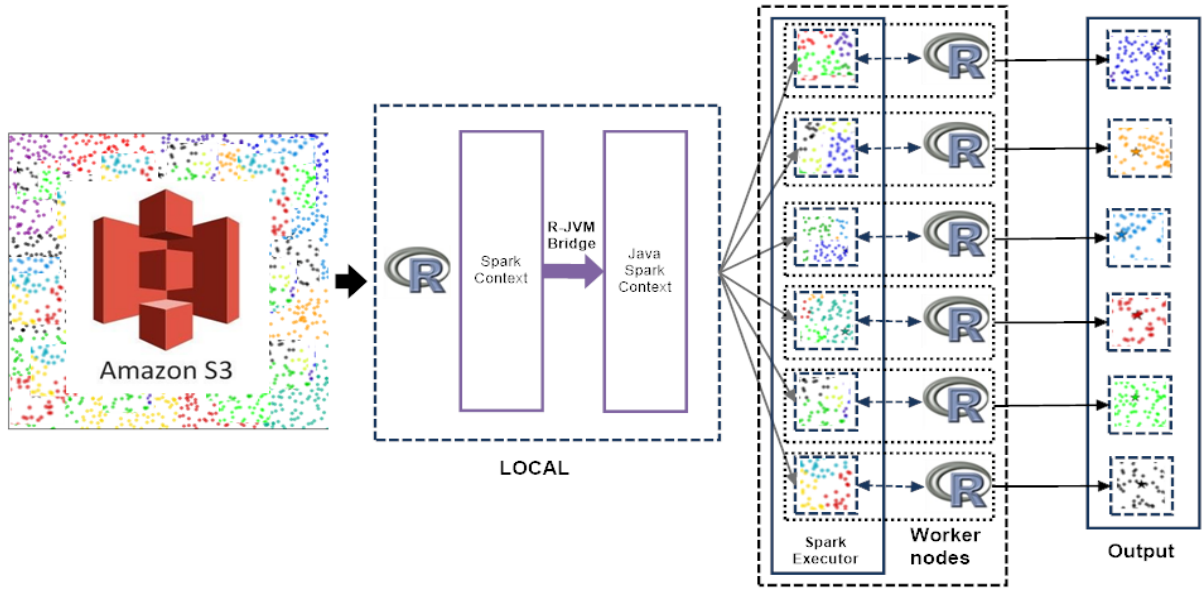


Figure 6.49: SparkR architecture used for experimentation

Table 6.19: Comparison of cluster quality for the proposed algorithms

Algorithms	AC	Fowlkes Mallows Index		
		Index	E_FM	V_FM
LGBACC	0.9944	1	0.2340068	0.00047908
GBAC	0.7376	0.831	0.000465811	0.2457143

lion data points and takes approximately 260 GB on the disk. The data has been taken from January 2009 to December 2016. We have worked on 21 attributes combined from all the tables in the database. These attributes are: medallion number, license, vender_id, rate_code, store_and_fwd, pickup_datetime, dropoff_datetime, passenger_count, trip_time_in_secs, trip_distance, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude, payment_type, fare_amount, surcharge, mta_tax, tip_amount, tolls_amount and total amount.

We performed experiments on this large dataset to compare the two algorithms on the basis of Agglomerative Coefficient and FM index. The results are given in Table 6.19. We discover the LGBACC generates better quality clusters than GBAC. Though both of them produce pretty high quality clusters, LGBACC is a little better than GBAC.

The visualization of experiments performed on LGBAAC and GBAC clustering are in-

cluded. All these figures represent a data sample of 23 weeks of taxi activity, chosen randomly from the dataset. This has been done for clear visibility of the data points. Figure 6.50 shows a correlation plot depicting correlations between clusters formed on the basis of taxi activity on each day of the week. It can be inferred that neither a positive nor a negative correlation exists between these clusters. Figure 6.51 shows data clustering by GBAC, it is represented in the form of bar plots. Figure 6.52 is a more systematic representation of this clustering in the form of a line graph. The clusters are clearly represented with the help of different colored lines and the variation in data points is shown by the slope of the line. Figure 6.53 represents clustering performed on the first half of the LGBACC algorithm, before performing PCA. Figure 6.54 shows percentage of explained variance against the dimensions, to perform PCA on the dataset. Figure 6.55 is representation of the variance factor map for PCA. The Figures show steps of PCA in detail. These representations are of the PCA phase of the LGBACC algorithm. Figure 6.56 shows 3D representation of the hierarchical clustering performed on the dataset produced after performing PCA (S2). Figure 6.57 shows a cluster plot of the distinct quality clusters produced by LGBACC algorithm.

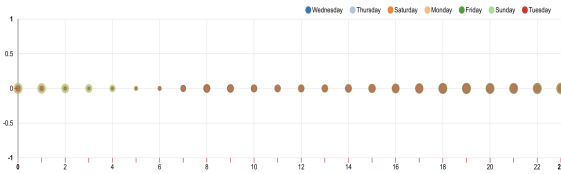


Figure 6.50: Correlation plot showing correlation between the clusters

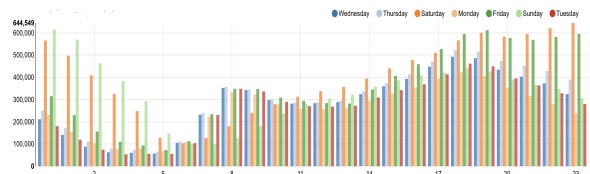


Figure 6.51: Clustering by GBAC

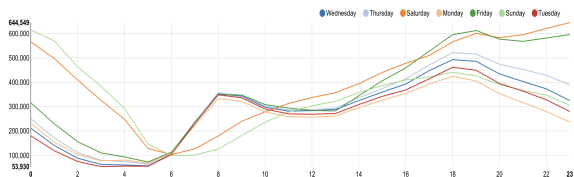


Figure 6.52: Clustering by GBAC

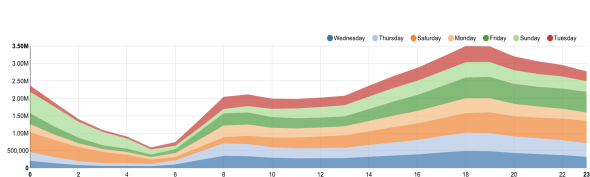


Figure 6.53: Clustering on first half of LGBACC, before performing PCA

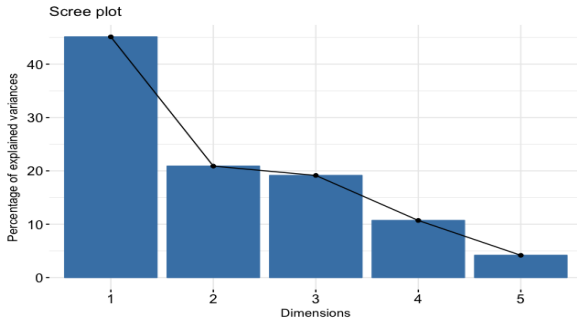


Figure 6.54: Percentage of variance v/s Dimensions

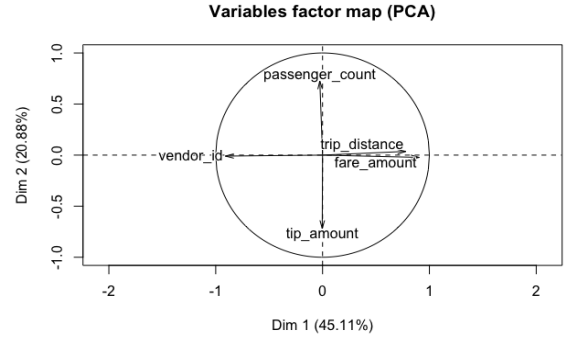


Figure 6.55: Variance factor map for PCA

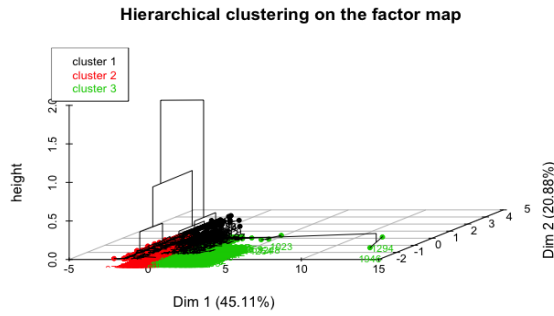


Figure 6.56: Hierarchical clustering on the factor map

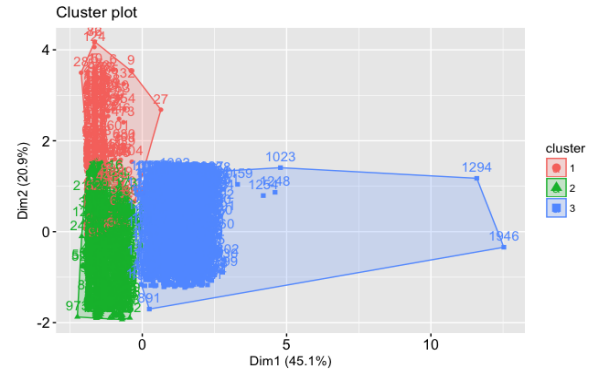


Figure 6.57: Cluster plot of the distinct clusters

Table 6.20: Description of datasets

Dataset	Size (approx.)	No. of data points (approx.)
DS1	260 GB	1.3 billion
DS2	520 GB	2.6 billion
DS3	780 GB	3.9 billion
DS4	1.04 TB	5.2 billion

We have used look-alike model to extend this dataset and generate four different datasets, to test the scalability of the proposed approaches, and compare them to the existing hierarchical clustering algorithms. The details of the extended datasets in terms of data points and size are given in Table 6.20. We have used the execution time parameter to run this analy-

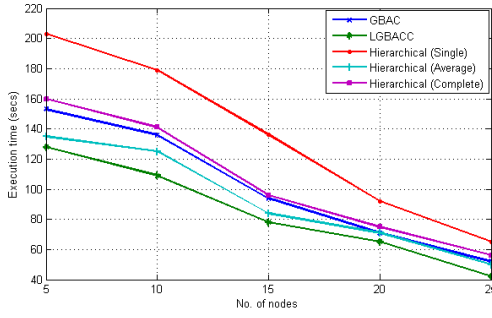


Figure 6.58: Execution time of clustering algorithms on DS1

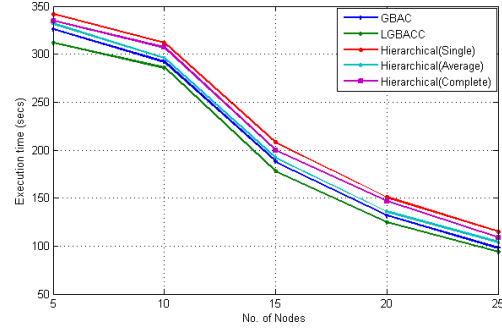


Figure 6.59: Execution time of clustering algorithms on DS2

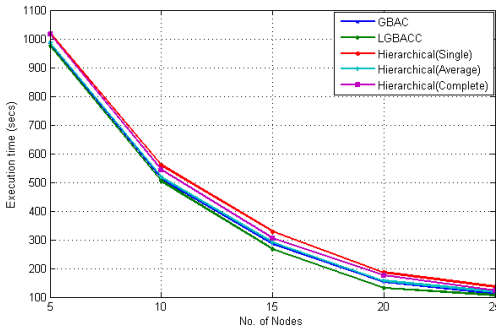


Figure 6.60: Execution time of clustering algorithms on DS3

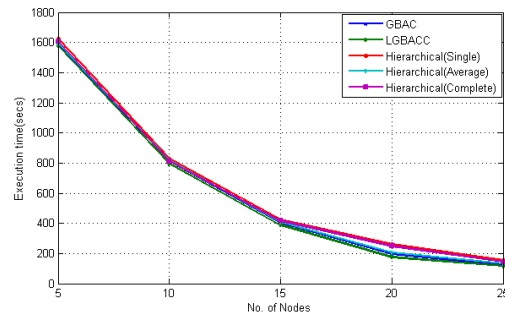


Figure 6.61: Execution time of clustering algorithms on DS4

sis. Different number of worker nodes have been used to compare the algorithms, varying between 5 and 25. The results of this analysis are given in Fig 6.58 to Figure 6.61. A linear relationship is observed between the execution times of the algorithms. It is observed that LGBACC has the fastest execution time as compared to other algorithms. For DS1, at 25 nodes, it is 19% faster than GBAC and 35.3% faster than the single link hierarchical clustering algorithm, which takes the maximum time out all the algorithms considered for this experiment. For DS3, at minimum number of nodes, 5, it is 4.2% faster than GBAC algorithm, but is 5.3 % faster, at 20 nodes, of the same dataset. For the largest dataset(DS4), LGBACC is again the fastest at all the nodes and average hierarchical clustering algorithm is a close second with GBAC trailing very closely behind. It was observed that LGBACC and average link hierarchical clustering are the fastest out of the five algorithms. LGBACC,

is a little faster than the average link, this is due to the time reduction achieved by applying PCA. GBAC is better than Single and Complete link hierarchical clustering algorithms, in terms of execution time.

6.6 Comparison between the proposed approaches

Both the proposed approaches are similar in the manner that they produce quality clusters, and overcome the problem of specifying the number of clusters in advance. There are many points of differentiation as well. The comparison of GBAC and LGBACC is given in Table 6.22 and points of differentiation between different correlation clustering approaches is given in Table 6.21.

Table 6.21: Comparison between different correlation clustering algorithms

	Complex Correlation	Simple +ve correlation	Simple -ve correlation	Dim. reduction	Hierarchical structure	Independent w.r.t attributes	Independent w.r.t objects	Noise robust
ORCLUS	y	y	y	y	-	y	-	-
4C	y	y	y	y	-	y	y	y
COPAC	y	y	y	y	-	y	y	y
ERIC	y	y	y	y	y	y	y	y
CASH	y	y	y	-	y	-	y	y
LGBACC	y	y	y	y	y	y	y	y
GBAC	y	y	y	-	y	y	y	y

Table 6.22: Comparison between proposed algorithms

	LGBACC	GBAC
Defining features	1. Works on standardized data 2. Uses PCA to reduce dimensions 3. Uses graph theory to obtain better cluster quality	1. Based on calculating mathematical correlations between data points. 2. Uses agglomerative clustering to produce clusters 3. Tests the quality of clusters produced
Suitable for data types	Qualitative and quantitative data	Numerical data, both discrete and continuous. Works best on ordinal, interval or ratio variables
Advantages	1. It is more general in nature as it works on similarity matrix data. 2. Very effective in handling high dimensional data 3. Dimensionality reduction reduces the execution time. 4. It works well on overlapping clusters	1. Does not assume number of clusters in advance 2. Produces high quality clusters 3. An ordering of the objects is obtained 4. Mathematical determination results in distinct clusters
Limitations	1. Multiple mathematical operations increase the computational cost 2. Dimensionality reduction may result in loss of veracity of data 3. It is based on locality assumption	1. Suitable only for numerical data 2. Works only on data matrix 3. Like all hierarchical clustering algorithms has high time complexity 4. Mathematical operations increase the time and resource utilization cost.

Chapter 7

Recommendation system model

In this chapter, we propose a recommender system model based on the proposed algorithms. This model merges the power of correlation clustering with that of prediction analysis to make better quality recommendations. The model is introduced by listing a set of preliminaries that are required to explain the phases of the model. The model is explained in detail, keeping in mind the input and output of each phase. Finally, the model is tested using a large dataset, on some predefined parameters. The proposed model makes recommendations based on a user's past behavior, as well as similar type of decisions made by other users. It is used to make predictions about the items or ratings that the users are interested in. They are better than the search algorithms as they help the users find the items they might not have found themselves.

7.1 Preliminaires

- Recommendation algorithms: There are many algorithms are used by the existing applications to make recommendations. We list the algorithms defined in the recommenderlab package of R language:

- User based collaborative filtering (UBCF) [247]: It analyzes the rating data collected from many individuals. The assumption is that the users with similar preferences rate items similarly. The missing ratings can be predicted by finding a cluster of similar users and aggregating the ratings to make predictions. The clusters are defined by using similarity measures that either give maximum points of similarity between the users or takes all the users above a particular threshold.
 - Item-based collaborative filtering (IBCF) [248]: This approach is based on a rating matrix. The recommendations are made based on items that can be inferred from the ratings matrix. This approach is based on the assumption that users will go for items similar to the other items they have liked.
 - User and Item-based collaborative filtering using 0-1 data [249]: This method is used in situations where less rating data is available. The usage behavior is analyzed to infer preferences. The information is presented in the form of 0's and 1's. 1 means that user has a preference for a product and 0 means not.
 - Recommendations for 0-1 data based on association rules [250]: The recommendations are made based on the dependency models for items given by some pre-defined association rules.
-
- K-fold cross validation [251]: The dataset is split into K sets (called folds) of approximately the same size. The evaluation is done k times, using one fold as a test fold and the all the other folds are used for learning. This is a robust approach for evaluating recommender algorithms, making sure that every user is in the test data, at least one. The averaging approach ensures robust results and error estimates.
 - Evaluation of predicted values: The best way to evaluate a predicted value is to use measures such as Mean Average Error (MAE), Mean Square Error (MSE) and Root

Table 7.1: Representation of confusion matrix

Actual / Predicted	Negative	Positive
Negative	a	b
Positive	c	d

Mean Square Error (RMSE) [252].

- MAE: This measure helps to evaluate a prediction by computing the deviation of a prediction from the true value. It is calculated as:

$$MAE = \frac{1}{|\kappa|} \sum_{(i,j) \in \kappa} |r_{ij} - \hat{r}_{ij}| \quad (7.1)$$

where, κ is the set of all user item pairings (i,j), $\hat{r}_{ij}$ is the predicted rating and r_{ij} is the known rating, that was not used to learn the model.

- RMSE: This is another popular measure to find out accuracy. It detects larger errors better than MAE and is suitable to situations where small prediction errors are not very important to find. It can be computed as:

$$RMSE = \sqrt{\frac{\sum_{(i,j) \in \kappa} (r_{ij} - \hat{r}_{ij})^2}{|\kappa|}} \quad (7.2)$$

$$MSE = (RMSE)^2$$

- Precision [253]: It is an information retrieval measure, it evaluates recommender performance using:

$$Precision = \frac{\text{Correctly recommended items}}{\text{Total recommended items}} \quad (7.3)$$

In terms of confusion matrix, given in Table 7.1, Precision can be expressed as:

$$Precision = \frac{d}{b+d} \quad (7.4)$$

- Recall [253]: It uses useful recommendations to define a measure for evaluating recommender performance. It is calculated as:

$$Recall = \frac{\text{Correctly recommended items}}{\text{Total useful recommendations}} \quad (7.5)$$

In terms of confusion matrix, it is represented as:

$$Recall = \frac{d}{c+d} \quad (7.6)$$

- ROC (Receiver Operating Character): It is a method to compare two classifiers at different parameter settings. The ROC curve is a method to detect system's probability (also known as sensitivity or True Positive Rate (TPR)) by the false positive rate, with regard to model parameters. The efficiency of the two systems can be computing by looking at the area under the ROC curve. Bigger area indicates better performance.

7.1.1 Model Explanation

The phases of the recommender system model presented in Figure 7.1 are explained as follows:

- Data pre-processing phase: After the dataset is imported into the computing environment, it needs to be organized in a way that it contains attributes that are required to make recommendations. Once the data is in the desired form, it is converted into a

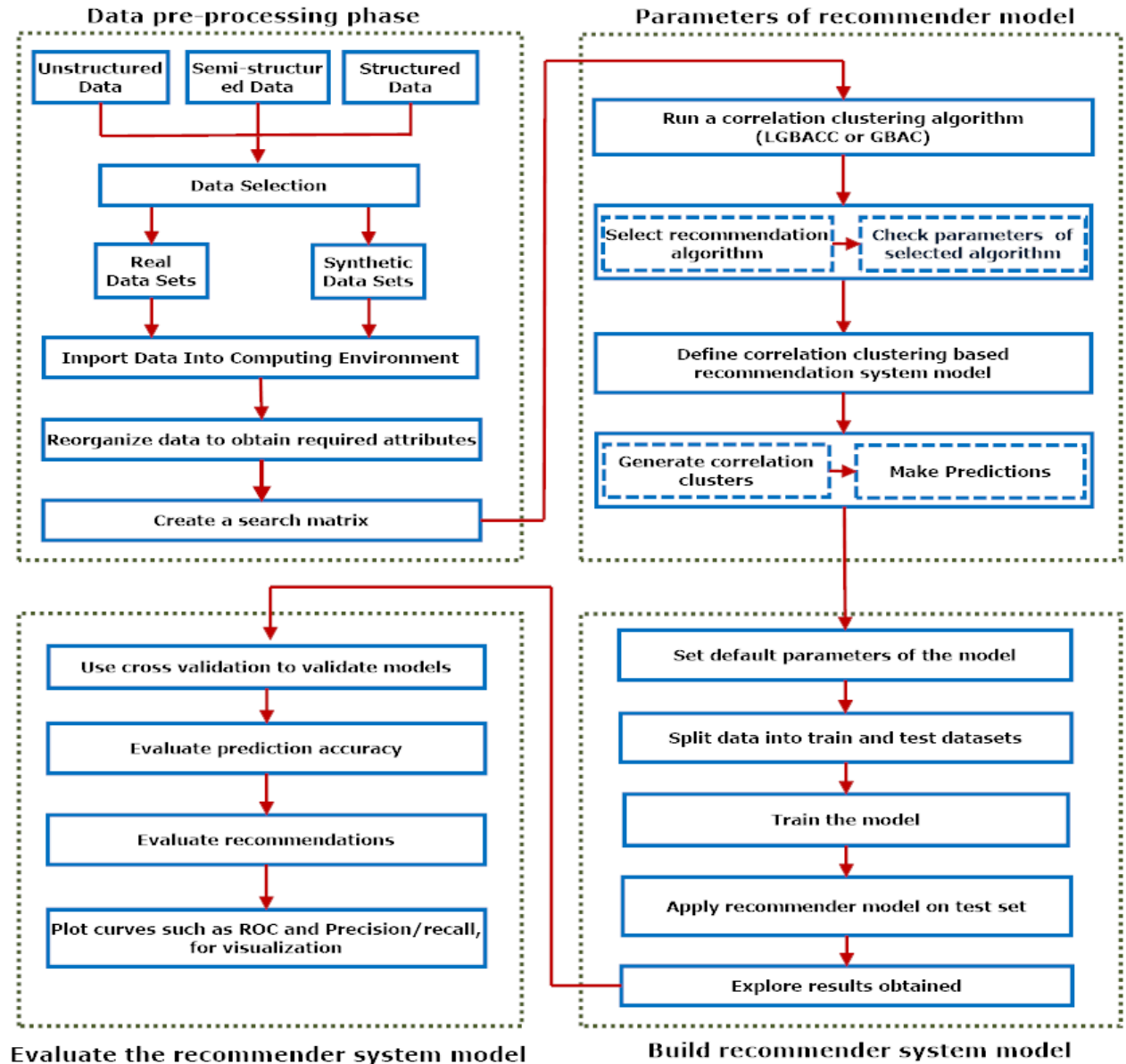


Figure 7.1: Recommender System Model

data matrix, known as the search matrix.

- **Select parameters of recommender model:** The search matrix acts as an input for running correlation clustering algorithms like LGBACC or GBAC. Once correlation clusters are obtained, a recommendation algorithm is selected depending on the kind of recommendations to be made. The next step is to define the parameters of the selected algorithm. This is done in preparation of defining correlation clustering based recom-

mendation system model. The core of this model is creating a relationship between generated correlated data points and parameters of the recommendation algorithm.

- **Build recommender system model:** Once the desired data is obtained, and the parameters of the recommendation algorithms are understood, the default parameters of the recommender model are defined. The next step is to take the clustered dataset and split it into train and test data sets. Then the data is trained and recommender model is applied on the test data set. This is the core of the proposed framework. The last step of this phase is to explore the results obtained.
- **Evaluate recommender system model:** K-fold cross validation method is used to validate the results obtained in the last phase. The next step is to evaluate the prediction accuracy by using parameters such as RSME, MSE and MAE. Then the recommendations are evaluated by using metrics like Precision and Recall. The last step is to visualize the quality of the recommendations made by plotting curves such as ROC and Precision/Recall.

7.1.2 Experimental Evaluation

We validate the recommendation system model by using a real world MovieLens dataset, taken from <http://grouplens.org/datasets/movielens/latest>. This dataset has 24,000,000 ratings and 670,000 tag applications applied to 40,000 movies by 260,000 users. The dataset is approximately 1 Gb in size and has been used by many researchers to make recommendations [254], [255], [256], [257]. The latest version of the dataset has information about users from the year 1996 to 2016.

There are four linked files, namely `links.csv`, `movies.csv`, `ratings.csv` and `tags.csv`. For

Table 7.2: Descriptive statistics of movies.csv

MovieId		Title	Genres
Min.	1	Length:40110	Length:40110
1st Qu.	32972	Class :character	Class :character
Mean	86841	Mode :character	Mode :character
3rd Qu.	134560		
Max.	165201		
Median	98457		

Table 7.3: Descriptive statistics of ratings.csv

	User ID	Move ID	Rating	Timestamp
Min.	1	1	0.500	7.897e+08
1st Qu.	63930	1015	3.000	9.784e+08
Median	129401	2424	3.500	1.132e+09
Mean	129374	13535	3.527	1.150e+09
3rd Qu.	194037	5816	4.000	1.308e+09
Max.	259137	165201	5.000	1.477e+09

testing the recommender model, only movies.csv and ratings.csv have been used. The descriptive statistics of the two files are given in Table 7.2 and Table 7.3. The search matrix is prepared by extracting a list of genres and making each genre a separate attribute. There are 18 genres, combining those with the attributes of ‘movies.csv’ the search matrix has 20 attributes. We run both LGBACC and GBAC on this matrix. The quality parameters of the two are given in Table 7.4. These algorithms provide similar data points, grouped together. We explore the data further and map the search matrix with ‘ratings.csv’ to prepare the data for making recommendations. We also normalize this data to make it cleaner for the next step of the recommendation system model. As for the recommendation algorithm, we select Item-based collaborative filtering (IBCF) model for our analysis. Looking at the requirements of the algorithm, we determine that the defining parameters of the algorithm

Table 7.4: Quality parameters for the two approaches

Algorithms	Agglomerative Coefficient	Fowlkes Mallows Index		
		Index	E_FM	V_FM
LGBAAC	0.9944	1	0.2340068	0.000479088
LGBAC	0.7376	0.831	0.000465811	0.2457143

Table 7.5: Evaluation of predicted values

		LGBACC	GBAC
Evaluation of predicted values	RMSE	0.9442798	1.2501181
	MSE	0.8916644	1.5627951
	MAE	0.7270231	0.9129824
Model time	Case 1	0.004sec	0.823sec
	Case 2	0.003sec	0.795sec
	Case 3	0.003sec	0.753sec
	Case 4	0.005sec	0.746sec
Prediction time	Case 1	0.175sec	0.033sec
	Case 2	0.166sec	0.03sec
	Case 3	0.18sec	0.031sec
	Case 4	0.17sec	0.031sec

are user ratings and preferences. These are the building blocks of the recommender system model. We need to link the correlation clusters with parameters of the recommendation algorithm. Linking correlated data points obtained to the user ratings and preferences, defines the correlation clustering based recommender system model. Next step is to set the default parameters of the IBCF model. We set the default parameter of method = cosine, and k (the number of items to compute similarities) as 30. Now, we build the model by splitting the whole data set as 80% training data and 20 % testing data. We apply the recommender system on the dataset, by training and testing the data. The results obtained are validated by running a 4-fold cross validation model. The results of this analysis in terms of RSME, MSE and MAE are given in Table 7.5. We see that LGBACC has approximately 25% better prediction capability than GBAC. Running this cross validation model for all the folds, we arrive at the conclusion that model time for LGBACC is much less than that of GBAC, but it takes significantly more prediction time.

Now we evaluate the recommendations by calculating Precision and Recall based on the confusion matrix. The confusion matrix for LGBACC and GBAC are given below:

<i>Rows</i>	TP	FP	FN	TN	precision	recall	TPR	FPR
10	4.59434	5.216981	71.13208	358.0566	0.4682692	0.07358824	0.07358824	0.01403121
20	7.490566	12.132075	68.23585	351.1415	0.3817308	0.11349283	0.11349283	0.0328015
30	10.084906	19.349057	65.64151	343.9245	0.3426282	0.14647926	0.14647926	0.05233702
40	12.537736	26.707547	63.18868	336.566	0.3194712	0.18124044	0.18124044	0.0723574
50	14.698113	34.358491	61.0283	328.9151	0.2996154	0.21481573	0.21481573	0.09344943
60	16.858491	42.009434	58.86792	321.2642	0.2863782	0.24290723	0.24290723	0.11428793

GBAC confusion matrix

<i>Rows</i>	TP	FP	FN	TN	precision	recall	TPR	FPR
10	2.066038	7.745283	73.66038	355.5283	0.2105769	0.03106647	0.03106647	0.02138576
20	4.396226	15.226415	71.33019	348.0472	0.2240385	0.06719202	0.06719202	0.04181422
30	6.490566	22.943396	69.23585	340.3302	0.2205128	0.09499043	0.09499043	0.0628056
40	8.509434	30.735849	67.21698	332.5377	0.2168269	0.12448159	0.12448159	0.08416257
50	10.566038	38.292453	65.16038	324.9811	0.2157479	0.15244732	0.15244732	0.10485478
60	12.660377	45.650943	63.06604	317.6226	0.2160726	0.17919405	0.17919405	0.1248983

LGBACC confusion matrix

We now plot the ROC and Precision-Recall curves for both LGBACC and GBAC from the confusion matrix. Looking at the Precision-Recall curve Figure 7.2 and Figure 7.3, we see that LGBACC has better Precision-Recall ratio than GBAC. But the area under the ROC curve for LGBACC is less than the other approach (Figure 7.4 and Figure 7.5).

We compare the existing recommendation models with each other by using IBCF and UBCF models. We create recommendation system models using both LGBACC and GBAC and compare them with the random recommendation system model. Precision-Recall and ROC curves for all the algorithms given in Figure 7.6 and Figure 7.7. It can be observed that

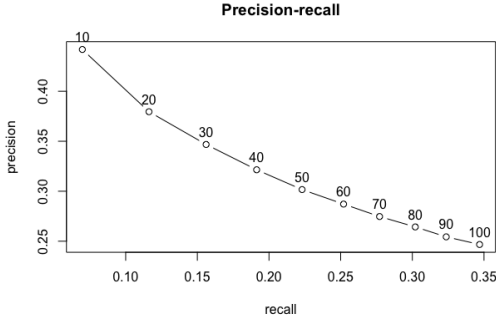


Figure 7.2: Precision-Recall curve for LGBACC

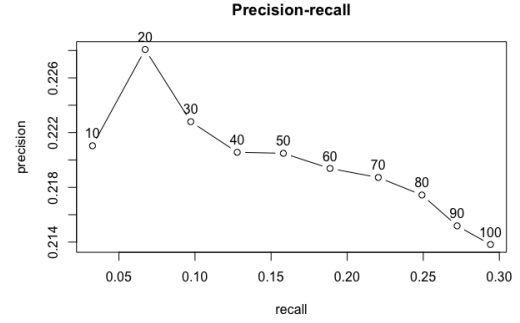


Figure 7.3: Precision-Recall curve for GBAC

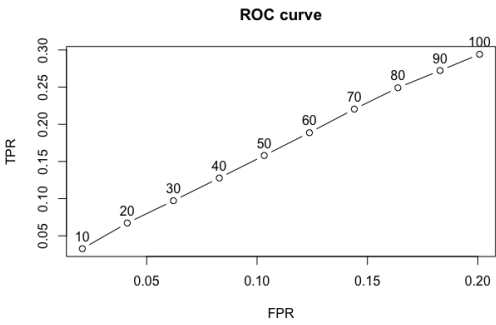


Figure 7.4: ROC curve for LGBACC

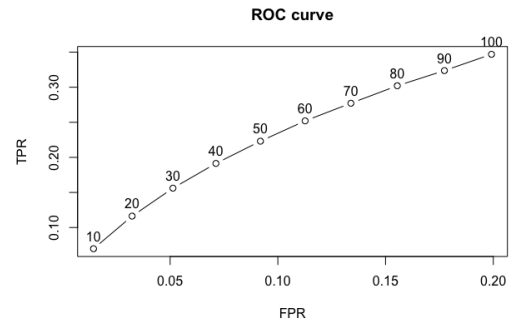


Figure 7.5: ROC for GBAC

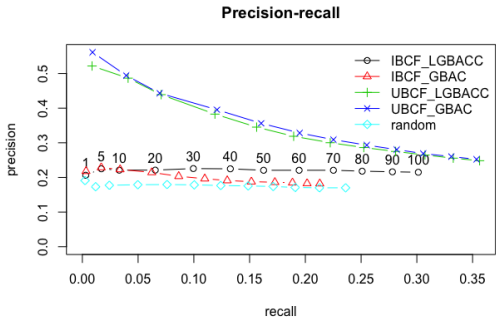


Figure 7.6: Comparison based on Precision-recall

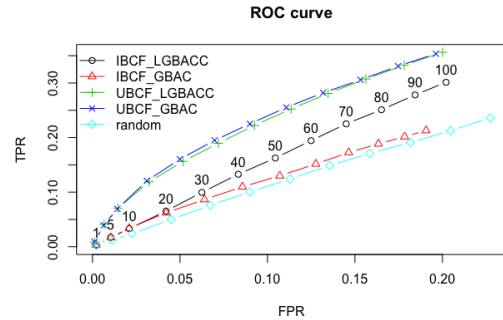


Figure 7.7: Comparison based on ROC

Random recommendation algorithm, that does not use the power of correlations performs the worst, on both the criteria. LGBACC and GBAC using UBCF, perform the best. This analysis shows that integrating correlated points with recommendation algorithms make better quality recommendations.

Chapter 8

Conclusion and future scope

Handling large-scale and high-dimensional data is a huge challenge. Data available for analysis these days is not always in a structured form. Most of it is in semi-structured or unstructured forms like tweets, logs, hashtags, blogs, images, videos, etc. Taking this data and deriving meaningful results out of it is the biggest challenge of data analysis. Existing data-mining techniques, including the clustering, need to be revisited in order to make them compatible with distributed computing environments. Platforms like Hadoop and Spark have come up to support this kind of computation. Given the huge amount of data, it is humanly impossible to go over all the data points individually. The challenge here is to develop sophisticated machine learning models based on intelligent data-mining techniques, such as clustering, that work in a distributed and parallel computing environment to analyze and process the given data. Another issue that needs to be addressed is the quality of clusters. Clustering algorithms need to be sophisticated enough to adopt better feature selection approaches. Feature selection, if performed well, can help to counter the drawbacks of dimensionality reduction.

Another challenge in dealing with high-dimensional data is overlapping clustering. False correlations formulated in different subspaces lead to formulation of overlapping clustering that in turn results in false interpretation of data. Data visualization is another point of con-

cern; and research, as most of it is done by using graph theory, and clustering can play a huge role in making it more accurate. The graph-clustering algorithms need to be improved in order to properly visualize data having a huge size and large dimensions. All these challenges are being addressed by researchers in both academia and industry, as the solutions to these problems will help understand hidden meanings in data in a much better way. Correlation clustering is an important data-mining approach for many applications.

The research objectives in this thesis are built on evolving a data-clustering approach based on finding correlations in data. There cannot be a general clustering technique that is suitable for solving all the problems and is applicable to all datasets. The choice of the clustering approach is based on the understanding of the data domain and the basic principles of the clustering approach in question. The survey presented in this thesis aims to provide a systematic study of the current approaches by segregating them into appropriate heads.

Many tools are used for the successful usage of correlation analysis in data science, and one of them is the use of correlation coefficients. In this thesis, we have proposed and validated an algorithm that gives the General (rank based) correlation coefficient (G). This metric is tested on the five criteria of a predictive metric. These are: independence from sample size; lies between -1 and 1; measures the degree of monotonicity instead of linearity; not sensitive to outliers; and demonstrates intuitiveness. This metric has been compared with the existing Spearman's rank (ρ) and Kendall (τ) correlation coefficient measures, both of which are rank-based. G has been found to be better than τ and comparable to ρ in terms of the accuracy of degree of correlation. In order to test the independence of G with regards to sample size, it has been tested on simulated data for n varying from 15000 to 100000. It is found independent of data size. Also, since it is comparable to ρ , it measures the degree of monotonicity instead of linearity. Given the construction of this metric, its value always lies between -1 and 1. The power of intuitiveness is tested by evaluating G on a real-time

dataset having human development indicators. G has been found to have better intuitive power than both ρ and τ . Much work can be done to refine the metric G by making it more concrete, such as, finding its statistical significance. It can also be expanded to deal with multi-dimensional input; currently, it takes $2 \times n$ sized arrays as input.

Further, a framework for GBAC has been proposed. This framework has three phases of data synthesis, correlation analysis and cluster analysis. All phases of this framework are visualized using plots such as heat maps, correlation plots, dendrograms and AGNES banners. The quality of the clusters produced is tested by using existing measures of agglomerative coefficients and FM index. On comparing GBAC with the existing hierarchical clustering approaches, using bk plots, G based clusters are found to be of better quality. This indicates that GBAC handles the outliers well, and is not sensitive to them. G fulfills all the five predictive metric criteria. GBAC gives a better quality of clusters and thus can be used in many domains of industry and academia. The most important application of this technique will be in the field of prediction analysis. It can be used in applications such as online predictive systems, health care information systems, political prediction systems, financial markets recommendation systems etc.

The second approach proposed in this thesis employs a search for hierarchies in correlation clusters. None of the approaches proposed earlier uses visualization and graph theory to explore hierarchies in datasets. We have proposed a novel approach that combines hierarchical clustering with dimensionality reduction methodology to reveal hierarchical relationships in the datasets. This approach focuses on improving the quality of the clusters produced. The resulting cluster hierarchy is visualized by means of graph models. We have proved theoretically and experimentally that LGBACC is a state-of-the-art clustering algorithm. The only problem with this methodology is that it makes the assumption that the members of a cluster are present on a common hyperplane. This drawback is a common denominator for

all approaches based on locality assumption.

Both of these algorithms act as the building blocks for designing a recommendation system model through correlations. This model uses correlation clustering algorithms to determine similar data points; use this measure of similarity and merge it with existing recommendation algorithms. It has been discovered that recommendations made using correlated data points are better and faster to compute. A parallel and distributed computing environment has been used to perform all types of analysis. We have used SparkR platform to process high-dimensional and large datasets.

Future Scope

At the end of the thesis, we emphasize on the potentials of the proposed methods for future research. Instead of considering the data points being distributed uniformly in one correlation hyperplane, it would be more practical to work on different distributions of data points within the same hyperplane. This indicates that the meaning of correlation clusters will change intuitively. Another interesting aspect that can be worked on is finding a uniform definition of correlation dimensionality. Dimensionality varies with the datasets, but there can be a universal definition that can be arrived at in in-depth analysis. The correlation clustering frameworks defined in this thesis can be defined in a more general way, having full-dimensional clusters as a special case. The most important area to explore is detection of non-linear structures in data. Most of the correlation clustering algorithms identify linear correlations among subsets of features in high-dimensional data. Concepts such as employing kernel PCA and Hough transformations can help to extend the research in this area to deal with non-linear correlations. The existing qualitative and quantitative models can also be extended to deal with non-linear correlations. The main focus of the future research is to make data-mining approaches more general and suitable to all kinds of data. This way, the

practical problems can be solved in a direct way, instead of using simplified representations using numerical attributes. To implement this, the complex objects need to be represented perfectly in terms of simple objects like numerical feature vectors, and they need to be incorporated into the data-mining process, keeping the domain knowledge in consideration.

Bibliography

- [1] U. Boryczka, “Finding groups in data: Cluster analysis with ants,” *Applied Soft Computing*, vol. 9, no. 1, pp. 61–70, 2009.
- [2] I. R. Rao, “Data mining and clustering techniques,” in *DRTC Workshop on Semantic Web*, vol. 8, 8- 10 December 2003.
- [3] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
- [4] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.
- [5] C. Fraley and A. E. Raftery, “How many clusters? which clustering method? answers via model-based cluster analysis,” *The computer journal*, vol. 41, no. 8, pp. 578–588, 1998.
- [6] P. Brucker, “On the complexity of clustering problems,” in *Optimization and operations research*, pp. 45–54, Springer, 1978.
- [7] W. J. Welch, “Algorithmic complexity: three np-hard problems in computational statistics,” *Journal of Statistical Computation and Simulation*, vol. 15, no. 1, pp. 17–25, 1982.

- [8] J. Edmonds, "Paths, trees, and flowers," *Canadian Journal of mathematics*, vol. 17, no. 3, pp. 449–467, 1965.
- [9] K. R. Prasad and B. E. Reddy, "Assessment of clustering tendency through progressive random sampling and graph-based clustering results," in *IEEE 3rd International Advance Computing conference (IACC)*, pp. 726–731, IEEE, 2013.
- [10] L. Galluccio, O. Michel, P. Comon, M. Kliger, and A. O. Hero, "Clustering with a new distance measure based on a dual-rooted tree," *Information Sciences*, vol. 251, pp. 96–113, 2013.
- [11] J. Kleinberg, "An impossibility theorem for clustering," *Advances in neural information processing systems*, pp. 463–470, 2003.
- [12] J. Hauke and T. Kossowski, "Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data," *Quaestiones Geographicae*, vol. 30, no. 2, pp. 87–93, 2011.
- [13] V. Didelez and I. Pigeot, "Judea pearl: Causality: Models, reasoning, and inference," *Politische Vierteljahresschrift*, vol. 42, no. 2, pp. 313–315, 2001.
- [14] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011.
- [15] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," in *IEEE Annual Symposium on Foundations of Computer Science(FOCS)*, vol. 238, 2002.
- [16] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine Learning*, vol. 56, no. 1-3, pp. 89–113, 2004.

- [17] M. R. Garey and D. S. Johnson, “Computers and intractability: a guide to the theory of np-completeness,” *San Francisco, LA: Freeman*, pp. 90–91, 1979.
- [18] A. Blum, A. Kalai, and H. Wasserman, “Noise-tolerant learning, the parity problem, and the statistical query model,” *Journal of the ACM (JACM)*, vol. 50, no. 4, pp. 506–519, 2003.
- [19] M. Kearns, “Efficient noise-tolerant learning from statistical queries,” *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 983–1006, 1998.
- [20] A. Condon and R. M. Karp, “Algorithms for graph partitioning on the planted partition model,” *Random Structures and Algorithms*, vol. 18, no. 2, pp. 116–140, 2001.
- [21] F. McSherry, “Spectral partitioning of random graphs,” in *Proceedings of 42nd IEEE Symposium on Foundations of Computer Science*, pp. 529–537, Las Vegas, Nevada, USA, 2001.
- [22] R. Shamir and D. Tsur, “Improved algorithms for the random cluster graph model,” in *Algorithm Theory: SWAT 2002*, vol. 31, pp. 230–239, Springer, 2002.
- [23] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica, “Correlation clustering in general weighted graphs,” *Theoretical Computer Science*, vol. 361, no. 2, pp. 172–187, 2006.
- [24] N. Ailon, M. Charikar, and A. Newman, “Aggregating inconsistent information: ranking and clustering,” *Journal of the ACM (JACM)*, vol. 55, no. 5, p. 23, 2008.
- [25] M. Charikar, V. Guruswami, and A. Wirth, “Clustering with qualitative information,” vol. 71, pp. 360–383, Elsevier, 2005.

- [26] I. Giotis and V. Guruswami, “Correlation clustering with a fixed number of clusters,” in *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pp. 1167–1176, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.
- [27] O. Goldreich, S. Goldwasser, and D. Ron, “Property testing and its connection to learning and approximation,” *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 653–750, 1998.
- [28] C. Mathieu and W. Schudy, “Correlation clustering with noisy input,” in *Proceedings of the twenty-first annual ACM-SIAM symposium on discrete algorithms*, pp. 712–728, Society for Industrial and Applied Mathematics, New Jersey, USA, 2010.
- [29] E. D. Demaine and N. Immorlica, “Correlation clustering with partial information,” in *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*, pp. 1–13, Springer, 2003.
- [30] T. Leighton and S. Rao, “Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms,” *Journal of the ACM (JACM)*, vol. 46, no. 6, pp. 787–832, 1999.
- [31] N. Garg, V. V. Vazirani, and M. Yannakakis, “Approximate max-flow min-(multi) cut theorems and their applications,” *SIAM Journal on Computing*, vol. 25, no. 2, pp. 235–251, 1996.
- [32] Y. Bejerano, N. Immorlica, S. Naor, and M. Smith, “Location area design in cellular networks,” in *International Conference on Mobile Computing and Networking (MOBICOM)*, San Diego, California, 2003.

- [33] P. Klein, S. A. Plotkin, and S. Rao, “Excluded minors, network decomposition, and multicommodity flow,” in *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pp. 682–690, San Diego, California, USA, ACM, 1993.
- [34] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, “The complexity of multiway cuts,” in *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pp. 241–251, Victoria, British Columbia, Canada, ACM, 1992.
- [35] N. Ailon and E. Liberty, “Correlation clustering revisited: The “true” cost of error minimization problems,” in *Automata, Languages and Programming*, pp. 24–36, Springer, 2009.
- [36] H.-P. Kriegel, P. Kröger, and A. Zimek, “Detecting clusters in moderate-to-high dimensional data: subspace clustering, pattern-based clustering, and correlation clustering,” *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1528–1529, 2008.
- [37] R. Bellman, R. E. Bellman, R. E. Bellman, and R. E. Bellman, *Adaptive control processes: a guided tour*, vol. 4. Princeton university press Princeton, 1961.
- [38] H.-P. Kriegel, P. Kröger, and A. Zimek, “Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 3, no. 1, p. 1, 2009.
- [39] C. M. Bishop, “Pattern recognition and machine learning,” *Information science and statistics. Springer*, pp. 1–15, 2006.
- [40] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, “When is nearest neighbor meaningful?,” in *7th International Conference on Database Theory*, pp. 217–235, 1012 January.

- [41] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.
- [42] C. Faloutsos and I. Kamel, "Beyond uniformity and independence: Analysis of r-trees using the concept of fractal dimension," in *Proceedings of the thirteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pp. 4–13, 24-27 May.
- [43] A. Belussi and C. Faloutsos, "Estimating the selectivity of spatial queries using the 'correlation' fractal dimension," pp. 1–26, 1995.
- [44] B.-U. Pagel, F. Korn, and C. Faloutsos, "Deflating the dimensionality curse using multiple fractal dimensions," in *Data Engineering, 2000. Proceedings. 16th International Conference on*, pp. 589–598, San Diego, CA, USA, IEEE, 3 March 2000.
- [45] F. Korn, B.-U. Pagel, and C. Faloutsos, "On the dimensionality curse and the self-similarity blessing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 1, pp. 96–111, 2001.
- [46] P. Ritala, A. Golnam, and A. Wegmann, "Coopetition-based business models: The case of amazon. com," *Industrial Marketing Management*, vol. 43, no. 2, pp. 236–249, 2014.
- [47] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, no. 1, pp. 76–80, 2003.
- [48] H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: From big data to big impact.," *MIS quarterly*, vol. 36, no. 4, pp. 1165–1188, 2012.
- [49] A. Sen, P. A. Dacin, and C. Pattichis, "Current trends in web data analysis," *Communications of the ACM*, vol. 49, no. 11, pp. 85–91, 2006.

- [50] S. D. Volpone, S. Tonidandel, D. R. Avery, and S. Castel, "Exploring the use of credit scores in selection processes: Beware of adverse impact," *Journal of Business and Psychology*, vol. 30, no. 2, pp. 357–372, 2015.
- [51] C. McGregor, "Big data in neonatal intensive care," *Computer*, vol. 46, no. 6, pp. 54–59, 2013.
- [52] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature*, vol. 457, no. 7232, pp. 1012–1014, 2009.
- [53] T. H. Davenport, P. Barth, and R. Bean, "How big data is different," *MIT Sloan Management Review*, vol. 54, no. 1, pp. 43–46, 2013.
- [54] H. Kitano, "Systems biology: a brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.
- [55] C. Murthy, S. Pal, and D. D. Majumder, "Correlation between two fuzzy membership functions," *Fuzzy Sets and Systems*, vol. 17, no. 1, pp. 23–38, 1985.
- [56] B. Chaudhuri and A. Bhattacharya, "On correlation between two fuzzy sets," *Fuzzy sets and systems*, vol. 118, no. 3, pp. 447–456, 2001.
- [57] S.-T. Liu and C. Kao, "Fuzzy measures for correlation coefficient of fuzzy numbers," *Fuzzy Sets and Systems*, vol. 128, no. 2, pp. 267–275, 2002.
- [58] D. H. Hong, "Fuzzy measures for a correlation coefficient of fuzzy numbers under t w (the weakest t-norm)-based fuzzy arithmetic operations," *Information Sciences*, vol. 176, no. 2, pp. 150–160, 2006.

- [59] A. Seth, H. Agarwal, and A. R. Singla, "Reliability estimation of services oriented systems using adaptive neuro fuzzy inference system," *Journal of Software Engineering and Applications*, vol. 7, no. 07, pp. 581–591, 2014.
- [60] W.-L. Hung, "Using statistical viewpoint in developing correlation of intuitionistic fuzzy sets," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 9, no. 04, pp. 509–516, 2001.
- [61] H. Mitchell, "A correlation coefficient for intuitionistic fuzzy sets," *International Journal of Intelligent Systems*, vol. 19, no. 5, pp. 483–490, 2004.
- [62] N. Chen, Z. Xu, and M. Xia, "Correlation coefficients of hesitant fuzzy sets and their applications to clustering analysis," *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 2197–2211, 2013.
- [63] H. Liao, Z. Xu, X.-J. Zeng, and J. M. Merigó, "Qualitative decision making with correlation coefficients of hesitant fuzzy linguistic term sets," *Knowledge-Based Systems*, vol. 76, pp. 127–138, 2015.
- [64] K. Seth, A. Sharma, and A. Seth, "Component selection efforts estimation—a fuzzy logic based approach," *International Journal of Computer Science and Security, (IJCSS)*, vol. 3, no. 3, pp. 210–215, 2009.
- [65] X. Huo and G. J. Székely, "Fast computing for distance covariance," *Technometrics*, vol. 58, no. 4, pp. 435–447, 2016.
- [66] J. Kong, B. E. Klein, R. Klein, K. E. Lee, and G. Wahba, "Using distance correlation and ss-anova to assess associations of familial relationships, lifestyle factors, diseases, and mortality," *Proceedings of the National Academy of Sciences*, vol. 109, no. 50, pp. 20352–20357, 2012.

- [67] R. Li, W. Zhong, and L. Zhu, “Feature screening via distance correlation learning,” *Journal of the American Statistical Association*, vol. 107, no. 499, pp. 1129–1139, 2012.
- [68] G. J. Székely and M. L. Rizzo, “On the uniqueness of distance covariance,” *Statistics & Probability Letters*, vol. 82, no. 12, pp. 2278–2282, 2012.
- [69] R. Lyons *et al.*, “Distance covariance in metric spaces,” *The Annals of Probability*, vol. 41, no. 5, pp. 3284–3305, 2013.
- [70] R. Xu and D. Wunsch, “Survey of clustering algorithms,” *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645–678, 2005.
- [71] C. P. Chen and C.-Y. Zhang, “Data-intensive applications, challenges, techniques and technologies: A survey on big data,” *Information Sciences*, vol. 275, pp. 314–347, 2014.
- [72] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, “The rise of ‘big data’ on cloud computing: Review and open research issues,” *Information Systems*, vol. 47, pp. 98–115, 2015.
- [73] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, “A survey of clustering algorithms for big data: Taxonomy and empirical analysis,” *IEEE transactions on emerging topics in computing*, vol. 2, no. 3, pp. 267–279, 2014.
- [74] A. B. Ayed, M. B. Halima, and A. M. Alimi, “Survey on clustering methods: Towards fuzzy clustering for big data,” in *6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pp. 331–336, Tunis, Tunisia, IEEE, 11-14 August 2014.

- [75] M. Chen, S. Mao, and Y. Liu, “Big data: A survey,” *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.
- [76] A. Amini, T. Y. Wah, and H. Saboohi, “On density-based data streams clustering algorithms: a survey,” *Journal of Computer Science and Technology*, vol. 29, no. 1, pp. 116–141, 2014.
- [77] S. Arora and I. Chana, “A survey of clustering techniques for big data analysis,” in *5th International Conference on Confluence The Next Generation Information Technology Summit (Confluence)*, pp. 59–65, Noida, India, IEEE, 25-26 September 2014.
- [78] B. Zerhari, A. A. Lahcen, and S. Mouline, “Big data clustering: Algorithms and challenges,” in *Proceedings of International Conference on Big Data, Cloud and Applications (BDCA’15)*, pp. 1–6, Tetuan, Morocco, IEEE, 25-26 May 2015.
- [79] O. Boyinbode, H. Le, and M. Takizawa, “A survey on clustering algorithms for wireless sensor networks,” *International Journal of Space-Based and Situated Computing*, vol. 1, no. 2-3, pp. 130–136, 2011.
- [80] C. C. Aggarwal and C. Zhai, “A survey of text clustering algorithms,” in *Mining text data*, pp. 77–128, Springer, 2012.
- [81] J. Brank, M. Grobelnik, and D. Mladenic, “A survey of ontology evaluation techniques,” in *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*, pp. 166–170, 27-30 November 2005.
- [82] J. Y. Yu and P. H. J. Chong, “A survey of clustering schemes for mobile ad hoc networks,” *IEEE Communications Surveys & Tutorials*, vol. 7, no. 1, pp. 32–48, 2005.
- [83] R. Garg and P. Mishra, “Some observations of sequential, parallel and distributed association rule mining algorithms,” in *IEEE International Conference on Computer*

- and Automation Engineering, ICCAE'09*, pp. 336–342, Bangkok, Thailand, IEEE, 8-10 March 2009.
- [84] R. Collobert and S. Bengio, “Svmtorch: Support vector machines for large-scale regression problems,” *The Journal of Machine Learning Research*, vol. 1, pp. 143–160, 2001.
- [85] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.
- [86] P. H. Sneath, “The application of computers to taxonomy,” *Microbiology*, vol. 17, no. 1, pp. 201–226, 1957.
- [87] T. Sørensen, “A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons,” *Biol. Skr.*, vol. 5, pp. 1–34, 1948.
- [88] F. Murtagh, “A survey of recent advances in hierarchical clustering algorithms,” *The Computer Journal*, vol. 26, no. 4, pp. 354–359, 1983.
- [89] M. Cochez and H. Mou, “Twister tries: Approximate hierarchical agglomerative clustering for average distance in linear time,” in *Proceedings of the 2015 ACM SIGMOD international conference on Management of data*, pp. 505–517, Melbourne, Victoria, Australia, ACM, June 2015.
- [90] G. N. Lance and W. T. Williams, “A general theory of classificatory sorting strategies: Ii. clustering systems,” *The computer journal*, vol. 10, no. 3, pp. 271–277, 1967.
- [91] T. Zhang, R. Ramakrishnan, and M. Livny, “Birch: an efficient data clustering method for very large databases,” in *ACM Sigmod Record*, vol. 25, pp. 103–114, ACM, 1996.

- [92] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," in *ACM SIGMOD Record*, vol. 27, pp. 73–84, ACM, 1998.
- [93] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.
- [94] B. Al-Shboul and S.-H. Myaeng, "Initializing k-means using genetic algorithms," *World Academy of Science, Engineering and Technology*, vol. 54, no. 30, pp. 114–118, 2009.
- [95] C. L. Liu, *Introduction to combinatorial mathematics*, vol. 181. McGraw-Hill New York, 1968.
- [96] M. Henzinger, "Finding near-duplicate web pages: a large-scale evaluation of algorithms," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 284–291, New York, NY, USA, ACM, August 06-11 2006.
- [97] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*, vol. 344. John Wiley & Sons, 2009.
- [98] R. Ng and J. Han, "Efficient and effective clustering method for spatial data mining," in *Proc. 1994 Int. Conf. Very Large Data Bases, Santiago, Chile*, pp. 144–155.
- [99] R. T. Ng and J. Han, "Clarans: A method for clustering objects for spatial data mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 5, pp. 1003–1016, 2002.
- [100] C. C. Aggarwal and S. Y. Philip, "A condensation approach to privacy preserving data mining," in *Advances in Database Technology-EDBT 2004*, pp. 183–199, Crete, Greece, Springer, March 14-18 2004.

- [101] B. Everitt and T. Hothorn, “Cluster analysis,” in *An Introduction to Applied Multivariate Analysis with R*, pp. 163–200, Springer, 2011.
- [102] X. Zhuang, Y. Huang, K. Palaniappan, and Y. Zhao, “Gaussian mixture density modeling, decomposition, and applications,” *IEEE Transactions on Image Processing*, vol. 5, no. 9, pp. 1293–1302, 1996.
- [103] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Knowledge Discovery and Data Mining*, vol. 96, pp. 226–231, 1996.
- [104] A. Hinneburg and D. A. Keim, “An efficient approach to clustering in large multimedia databases with noise,” in *Knowledge Discovery and Data Mining*, vol. 98, pp. 58–65, 1998.
- [105] N. Streekmann, “Graph clustering,” in *Clustering-Based Support for Software Architecture Restructuring*, pp. 45–53, Springer, 2012.
- [106] Y. Zhou, H. Cheng, and J. X. Yu, “Graph clustering based on structural/attribute similarities,” *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 718–729, 2009.
- [107] U. Von Luxburg, “A tutorial on spectral clustering,” *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [108] Y. Sun, J. Han, P. Zhao, Z. Yin, H. Cheng, and T. Wu, “Rankclus: integrating clustering with ranking for heterogeneous information network analysis,” in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pp. 565–576, Saint Petersburg, Russia, ACM, 24–26 March 2009.

- [109] Y. Sun, Y. Yu, and J. Han, “Ranking-based clustering of heterogeneous information networks with star network schema,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 797–806, Paris, France, ACM, June 2009.
- [110] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, “Pathsim: Meta path-based top-k similarity search in heterogeneous information networks,” *37th International Conference on Very Large Data Bases*, 29 August-3 September 2011.
- [111] G. Sheikholeslami, S. Chatterjee, and A. Zhang, “Wavecluster: A multi-resolution clustering approach for very large spatial databases,” in *VLDB*, vol. 98, pp. 428–439, 1998.
- [112] D. Barbará and P. Chen, “Using the fractal dimension to cluster datasets,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 260–264, ACM, 2000.
- [113] W. Wang, J. Yang, R. Muntz, *et al.*, “Sting: A statistical information grid approach to spatial data mining,” in *VLDB*, vol. 97, pp. 186–195, 1997.
- [114] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, *Automatic subspace clustering of high dimensional data for data mining applications*, vol. 27. ACM, 1998.
- [115] R. M. Almuttairi, R. Wankar, A. Negi, and R. R. Chillarige, “Rough set clustering approach to replica selection in data grids (rscdg),” in *10th International Conference on Intelligent Systems Design and Applications (ISDA)*, pp. 1195–1200, Cairo, Egypt, IEEE, November 2010.
- [116] H. Deng and J. Han, “Probabilistic models for clustering,” *Data Clustering: Algorithms and Applications*, p. 61, 2013.

- [117] T. White, *Hadoop: The definitive guide*. ” O’Reilly Media, Inc.”, 2012.
- [118] J. Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.
- [119] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon, “Parallel data processing with mapreduce: a survey,” *AcM SIGMoD Record*, vol. 40, no. 4, pp. 11–20, 2012.
- [120] H.-c. Yang, A. Dasdan, R.-L. Hsiao, and D. S. Parker, “Map-reduce-merge: simplified relational data processing on large clusters,” in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pp. 1029–1040, Beijing, China, ACM, 11-14 June 2007.
- [121] A. Gupta and L. K. Awasthi, “Peer enterprises: A viable alternative to cloud computing?,” in *IEEE International Conference on Internet Multimedia Services Architecture and Applications (IMSAA)*, pp. 1–6, Bangalore, India, IEEE, 9-11 December 2009.
- [122] W. Zhao, H. Ma, and Q. He, “Parallel k-means clustering based on mapreduce,” in *IEEE International Conference on Cloud Computing*, pp. 674–679, LNCS Springer, 2009.
- [123] N. Li, L. Zeng, Q. He, and Z. Shi, “Parallel implementation of apriori algorithm based on mapreduce,” in *13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD)*, pp. 236–241, Kyoto, Japan, IEEE, August 2012.
- [124] Z. Tang, K. Liu, J. Xiao, L. Yang, and Z. Xiao, “A parallel k-means clustering algorithm based on redundance elimination and extreme points optimization employing mapreduce,” *Concurrency and Computation: Practice and Experience*, doi 10.1002/cpe.4109, 2017.

- [125] M. O. Shafiq and E. Torunski, “A parallel k-medoids algorithm for clustering based on mapreduce,” in *15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 502–507, Anaheim, CA, USA, IEEE, 18-20 Dec. 2016.
- [126] Y. He, H. Tan, W. Luo, S. Feng, and J. Fan, “Mr-agrawal1998: a scalable mapreduce-based dbscan algorithm for heavily skewed data.,” *Frontiers of Computer Science*, vol. 8, no. 1, pp. 83–99, 2014.
- [127] D. Pandove and S. Goel, “Prototyping and in-depth analysis of big data benchmarking,” in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, pp. 1222–1229, IEEE, 2015.
- [128] S. Huang, J. Huang, J. Dai, T. Xie, and B. Huang, “The hibenach benchmark suite: Characterization of the mapreduce-based data analysis,” in *IEEE 26th International Conference on Data Engineering Workshops (ICDEW)*, pp. 41–51, Long Beach, CA, USA, IEEE, 1-6 March 2010.
- [129] Z. Ming, C. Luo, W. Gao, R. Han, Q. Yang, L. Wang, and J. Zhan, “Bdgs: A scalable big data generator suite in big data benchmarking,” in *Workshop on Big Data Benchmarks*, pp. 138–154, Springer, 2013.
- [130] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, and M. Stonebraker, “A comparison of approaches to large-scale data analysis,” in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pp. 165–178, Providence, Rhode Island, USA, ACM, 29 June - 2 July 2009.
- [131] L. Wang, J. Zhan, C. Luo, Y. Zhu, Q. Yang, Y. He, W. Gao, Z. Jia, Y. Shi, S. Zhang,

- et al.*, “Bigdatabench: A big data benchmark suite from internet services,” in *IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*, pp. 488–499, Orlando, FL, USA, IEEE, 15-19 Feb. 2014.
- [132] L. R. Medsker, *Hybrid intelligent systems*. Springer Science & Business Media, 2012.
- [133] L. Medsker, “Design and development of hybrid neural network and expert systems,” in *Neural Networks, 1994. IEEE International Conference on World Congress on Computational Intelligence.*, vol. 3, pp. 1470–1474, Orlando, FL, USA, USA, IEEE, 28 june- 2 July 1994.
- [134] P. S. Mohapatra, V. Popova, and L. Xu, “Training auditors in strategic systems auditing using online computerised systems: can prior schema affect auditors’ knowledge acquisition?,” *International Journal of Critical Accounting*, vol. 5, no. 5, pp. 453–468, 2013.
- [135] P. S. Mohapatra, A. Graham, and A. Nandialath, “Did pwc lose reputation post audit failure at satyam computer services? evidence from the indian audit market,” *International Journal of Accounting and Finance*, vol. 5, no. 1, pp. 48–61, 2015.
- [136] J. S. Priya, M. Ramakrishnan, and S. Rajagopalan, “Detection of ddos attacks using ip traceback and network coding technique.,” *Journal of Theoretical & Applied Information Technology*, vol. 62, no. 1, 2014.
- [137] A. Bhagat and R. K. Sahu, “Cloud data security while using third party auditor,” *International Journal of Computer Applications*, vol. 70, no. 16, 2013.
- [138] L. C. Bishnoi, D. Singh, and S. Mishra, “Traffic queues controlling cross-layer mechanism for svc video transmission over ieee 802.11 n wlans,” in *Proceedings of the*

- 2014 International Conference on Information and Communication Technology for Competitive Strategies*, p. 93, Udaipur, Rajasthan, India, ACM, 14-16 Nov. 2014.
- [139] S. P. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [140] D. Li, S. Wang, W. Gan, and D. Li, "Data field for hierarchical clustering," *Developments in Data Extraction, Management, and Analysis*, p. 303, 2012.
- [141] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proceedings of the twenty-first international conference on Machine learning*, pp. 2–9, 4-8 July.
- [142] C. C. Aggarwal and P. S. Yu, "Redefining clustering for high-dimensional applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 2, pp. 210–225, 2002.
- [143] A. Hinneburg and D. A. Keim, "Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering," pp. 506–517, 1999.
- [144] A. Quigley and P. Eades, "Fade: Graph drawing, clustering, and visual abstraction," in *International Symposium on Graph Drawing*, pp. 197–210, Springer, 2000.
- [145] A. Y. Ng, M. I. Jordan, Y. Weiss, *et al.*, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, vol. 14, pp. 849–856, 2001.
- [146] X.-Y. Li and L.-j. Guo, "Constructing affinity matrix in spectral clustering based on neighbor propagation," *Neurocomputing*, vol. 97, pp. 125–130, 2012.

- [147] F. Nie, C. Ding, D. Luo, and H. Huang, “Improved minmax cut graph clustering with nonnegative relaxation,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 451–466, Springer, 2010.
- [148] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, “Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering,” *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1796–1808, 2011.
- [149] F. Nie, X. Wang, M. I. Jordan, and H. Huang, “The constrained laplacian rank algorithm for graph-based clustering,” in *Association for the Advancement of Artificial Intelligence*, pp. 1969–1976, Citeseer, 2016.
- [150] F. Nie, D. Xu, and X. Li, “Initialization independent clustering with actively self-training method,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 1, pp. 17–27, 2012.
- [151] A. Geyer-Schulz and M. Ovelgönne, “The randomized greedy modularity clustering algorithm and the core groups graph clustering scheme,” in *German-Japanese Interchange of Data Analysis Results*, vol. 6322, pp. 17–36, Springer, 2014.
- [152] U. V. Catalyiirek, K. Kaya, J. Langguth, and B. Uçar, “A partitioning-based divisive clustering technique for maximizing the modularity,” *Graph Partitioning and Graph Clustering*, vol. 588, p. 171, 2013.
- [153] H. N. Djidjev and M. Onus, “Scalable and accurate graph clustering and community structure detection,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 1022–1029, 2013.
- [154] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, *Automatic subspace clustering of high dimensional data for data mining applications*, vol. 27. ACM, 1998.

- [155] C.-H. Cheng, A. W. Fu, and Y. Zhang, “Entropy-based subspace clustering for mining numerical data,” in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 84–93, San Diego, California, USA, ACM, 15-18 August 1999.
- [156] T. Murali and S. Kasif, “Extracting conserved gene expression motifs from gene expression data,” in *Pacific symposium on biocomputing*, vol. 8, pp. 77–88, 2003.
- [157] B. Liu, Y. Xia, and P. S. Yu, “Clustering through decision tree construction,” in *Proceedings of the ninth international conference on Information and knowledge management*, pp. 20–29, McLean, Virginia, USA, ACM, 6-11 November 2000.
- [158] K. Kailing, H.-P. Kriegel, and P. Kröger, “Density-connected subspace clustering for high-dimensional data,” in *Proc. SDM*, vol. 4, SIAM, 2004.
- [159] I. Assent, R. Krieger, E. Muller, and T. Seidl, “Dusc: dimensionality unbiased subspace clustering,” in *Seventh IEEE International Conference on Data Mining, ICDM*, pp. 409–414, Omaha, NE, USA, IEEE, 28-31 Oct. 2007.
- [160] P. Baldi and K. Hornik, “Neural networks and principal component analysis: Learning from examples without local minima,” *Neural networks*, vol. 2, no. 1, pp. 53–58, 1989.
- [161] E. Oja, “Principal components, minor components, and linear neural networks,” *Neural Networks*, vol. 5, no. 6, pp. 927–935, 1992.
- [162] E. Achtert, C. Böhm, P. Kroger, and A. Zimek, “Mining hierarchies of correlation clusters,” in *18th International Conference on Scientific and Statistical Database Management, 2006*, pp. 119–128, Vienna, Austria, IEEE, 3-5 July 2006.
- [163] C. C. Aggarwal and P. S. Yu, *Finding generalized projected clusters in high dimensional spaces*, vol. 29. 15-18 May.

- [164] K. Chakrabarti and S. Mehrotra, “Local dimensionality reduction: A new approach to indexing high dimensional spaces.,” in *International conference on very large data bases*, pp. 89–100, 2000.
- [165] J. Li, X. Huang, C. Selke, and J. Yong, “A fast algorithm for finding correlation clusters in noise data,” in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, vol. 4468, pp. 639–647, Springer, 2007.
- [166] C. Böhm, K. Kailing, P. Kröger, and A. Zimek, “Computing clusters of correlation connected objects,” in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pp. 455–466.
- [167] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek, “Robust, complete, and efficient correlation clustering,” in *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 413–418, Minnesota, SIAM, 26-28 April 2007.
- [168] E. Achtert, C. Bohm, H.-P. Kriegel, P. Kroger, and A. Zimek, “On exploring complex relationships of correlation clusters,” in *Scientific and Statistical Database Management, 2007. SSBDM’07. 19th International Conference on*, pp. 7–7, Banff, Alta., Canada, IEEE, 9-11 July 2007.
- [169] A. K. Tung, X. Xu, and B. C. Ooi, “Curler: finding and visualizing nonlinear correlation clusters,” in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 467–478, ACM, 2005.
- [170] P. Mukhopadhyay and B. B. Chaudhuri, “A survey of hough transform,” *Pattern Recognition*, vol. 48, no. 3, pp. 993–1010, 2015.
- [171] A. K. Chattopadhyay, T. Chattyopadhyay, T. De, and S. Mondal, “Independent component analysis for dimension reduction classification: Hough transform and cash algo-

- rithm,” in *Astrostatistical Challenges for the New Astronomy*, pp. 185–202, Springer, 2013.
- [172] P. K. Mishra, K. S. Mishra, and A. Mishra, “A clustering heuristic for multiprocessor environments using computation and communication loads of modules,” *International Journal of Computer Science & Information Technology*, vol. 2, no. 5, pp. 170–182, 2010.
- [173] D. Barbará and P. Chen, “Using the fractal dimension to cluster datasets,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 260–264, Boston, Massachusetts, USA, ACM, 20-23 August 2000.
- [174] A. Gionis, A. Hinneburg, S. Papadimitriou, and P. Tsaparas, “Dimension induced clustering,” in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 51–60, Chicago, Illinois, USA, ACM, 21-24 August 2005.
- [175] R. Haralick and R. Harpaz, “Linear manifold clustering,” in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 132–141, Springer, 2005.
- [176] R. Harpaz and R. Haralick, “Mining subspace correlations,” in *IEEE Symposium on Computational Intelligence and Data Mining*, pp. 335–342, Honolulu, HI, USA, IEEE, 1 March-5 April 2007.
- [177] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park, “Fast algorithms for projected clustering,” in *ACM SIGMoD Record*, vol. 28, pp. 61–72, ACM, 1999.
- [178] C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma, “Subspace clustering of

- high dimensional data.,” in *Proceedings of the 2004 SIAM international conference on data mining*, pp. 517–521, Florida, USA, SIAM, 22-24 April 2004.
- [179] K.-G. Woo, J.-H. Lee, M.-H. Kim, and Y.-J. Lee, “Findit: a fast and intelligent subspace clustering algorithm using dimension voting,” *Information and Software Technology*, vol. 46, no. 4, pp. 255–271, 2004.
- [180] K. Y. Yip, D. W. Cheung, and M. K. Ng, “On discovery of extremely low-dimensional clusters using semi-supervised projected clustering,” in *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pp. 329–340, IEEE, 2005.
- [181] C. Böhm, K. Railing, H.-P. Kriegel, and P. Kröger, “Density connected clustering with local subspace preferences,” in *Fourth IEEE International Conference on Data Mining, 2004. ICDM’04.*, pp. 27–34, Tokyo, Japan, IEEE, 5-8 April 2004.
- [182] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. Murali, “A monte carlo algorithm for fast projective clustering,” in *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pp. 418–427, Madison, Wisconsin, ACM, June 2002.
- [183] M. L. Yiu and N. Mamoulis, “Frequent-pattern based iterative projected clustering,” in *Third IEEE International Conference on Data Mining, ICDM 2003*, pp. 689–692, Melbourne, FL, USA, IEEE, 22-22 Nov. 2003.
- [184] M. L. Yiu and N. Mamoulis, “Iterative projected clustering by subspace mining,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 2, pp. 176–189, 2005.

- [185] J. H. Friedman and J. J. Meulman, “Clustering objects on subsets of attributes (with discussion),” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 66, no. 4, pp. 815–849, 2004.
- [186] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, I. Müller-Gorman, and A. Zimek, “Detection and visualization of subspace cluster hierarchies,” in *Advances in databases: Concepts, systems and applications*, pp. 152–163, Springer, 2007.
- [187] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, “Optics: ordering points to identify the clustering structure,” in *ACM Sigmod Record*, vol. 28, pp. 49–60, ACM, 1999.
- [188] M. Ramakrishnan, G. Rajan, Y. Semenova, and G. Farrell, “Overview of fiber optic sensor technologies for strain/temperature sensing applications in composite materials,” *Sensors*, vol. 16, no. 1, pp. 1–27, 2016.
- [189] K. Y. Yip, D. W. Cheung, and M. K. Ng, “Harp: A practical projected clustering algorithm,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1387–1397, 2004.
- [190] H.-P. Kriegel, P. Kröger, M. Renz, and S. Wurst, “A generic framework for efficient subspace clustering of high-dimensional data,” in *Fifth IEEE International Conference on Data Mining*, pp. 8–13, Houston, TX, USA, IEEE, 27-30 Nov. 2005.
- [191] K. Sequeira and M. Zaki, “Schism: A new approach for interesting subspace mining,” in *Fourth IEEE International Conference on Data Mining, 2004. ICDM’04.*, pp. 186–193, Brighton, UK, IEEE, 1-4 Nov. 2004.
- [192] G. Moise, J. Sander, and M. Ester, “P3c: A robust projected clustering algorithm,” in

- Sixth International Conference on Data Mining, 2006. ICDM'06.*, pp. 414–425, Hong Kong, China, IEEE, 18-22 Dec. 2006.
- [193] G. Moise, J. Sander, and M. Ester, “Robust projected clustering,” *Knowledge and Information Systems*, vol. 14, no. 3, pp. 273–298, 2008.
- [194] P.-N. Tan, M. Steinbach, and V. Kumar, “Data mining cluster analysis: basic concepts and algorithms,” *Introduction to data mining*, 2013.
- [195] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 881–892, 2002.
- [196] T. Zhang, R. Ramakrishnan, and M. Livny, “Birch: an efficient data clustering method for very large databases,” in *ACM Sigmod Record*, vol. 25, pp. 103–114, ACM, 1996.
- [197] S. Guha, R. Rastogi, and K. Shim, “Cure: an efficient clustering algorithm for large databases,” *Information Systems*, vol. 26, no. 1, pp. 35–58, 2001.
- [198] G. Karypis, E.-H. Han, and V. Kumar, “Chameleon: Hierarchical clustering using dynamic modeling,” *Computer*, vol. 32, no. 8, pp. 68–75, 1999.
- [199] M. H. Nadimi and M. Mosakhani, “A more accurate clustering method by using co-author social networks for author name disambiguation,” *Journal of Computing and Security*, vol. 1, no. 4, 2015.
- [200] Y. Liu and Y. Tang, “Network based framework for author name disambiguation applications,” *International Journal of u-and e-Service, Science and Technology*, vol. 8, no. 9, pp. 75–82, 2015.

- [201] C. C. Aggarwal and C. Zhai, *Mining text data*. Springer Science & Business Media, 2012.
- [202] Y. Sun and J. Han, “Mining heterogeneous information networks: a structural analysis approach,” *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 20–28, 2013.
- [203] M. Steyvers and T. Griffiths, “Probabilistic topic models,” *Handbook of latent semantic analysis*, vol. 427, no. 7, pp. 424–440, 2007.
- [204] M. W. Berry and M. Castellanos, “Survey of text mining,” *Computing Reviews*, vol. 45, no. 9, p. 548, 2004.
- [205] M. Naaman, “Social multimedia: highlighting opportunities for search and mining of multimedia data in social media applications,” *Multimedia Tools and Applications*, vol. 56, no. 1, pp. 9–34, 2012.
- [206] H. Jin, J. Chen, H. He, G. J. Williams, C. Kelman, and C. M. O’Keefe, “Mining unexpected temporal associations: applications in detecting adverse drug reactions,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 4, pp. 488–500, 2008.
- [207] J. Scott, *Social network analysis*. Sage, 2012.
- [208] C.-G. Huang, J. Yin, J. Wang, Y.-B. Liu, and J.-H. Wang, “Uncertain neighbors’ collaborative filtering recommendation algorithm,” *Jisuanji Xuebao(Chinese Journal of Computers)*, vol. 33, no. 8, pp. 1369–1377, 2010.
- [209] S. Wei, N. Ye, S. Zhang, X. Huang, and J. Zhu, “Collaborative filtering recommendation algorithm based on item clustering and global similarity,” in *Fifth International Conference on Business Intelligence and Financial Engineering (BIFE)*, pp. 69–72, Lanzhou, China, IEEE, 18-21 Aug. 2012.

- [210] J. Wang, A. P. De Vries, and M. J. Reinders, “Unifying user-based and item-based collaborative filtering approaches by similarity fusion,” in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 501–508, ACM, 2006.
- [211] S. W. Changchien and T.-C. Lu, “Mining association rules procedure to support on-line recommendation by customers and products fragmentation,” *Expert systems with applications*, vol. 20, no. 4, pp. 325–335, 2001.
- [212] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-based systems*, vol. 46, pp. 109–132, 2013.
- [213] S. Guha, N. Mishra, R. Motwani, and L. O’Callaghan, “Clustering data streams,” in *proceedings of 41st annual symposium on Foundations of computer science, 2000*, pp. 359–366, IEEE, 2000.
- [214] L. O’callaghan, A. Meyerson, R. Motwani, N. Mishra, and S. Guha, “Streaming-data algorithms for high-quality clustering,” in *Proceedings of 18th International Conference on Data Engineering*, p. 0685, San Jose, CA, USA, IEEE, 26 Feb.-1 March 2002.
- [215] M. Hahsler and M. Bolaños, “Clustering data streams based on shared density between micro-clusters,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1449–1461, 2016.
- [216] P. Langfelder, B. Zhang, and S. Horvath, “Defining clusters from a hierarchical cluster tree: the dynamic tree cut package for r,” *Bioinformatics*, vol. 24, no. 5, pp. 719–720, 2008.

- [217] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pp. 81–92, VLDB Endowment, 2003.
- [218] Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang, "Multi-dimensional regression analysis of time-series data streams," in *Proceedings of the 28th international conference on Very Large Data Bases*, pp. 323–334, VLDB Endowment, 2002.
- [219] M.-H. Masson and T. Denoeux, "Ensemble clustering in the belief functions framework," *International Journal of Approximate Reasoning*, vol. 52, no. 1, pp. 92–109, 2011.
- [220] L. Kauffman and P. Rousseeuw, "Finding groups in data," *An introduction to cluster analysis*. New York: John Willey & Sons, 1990.
- [221] M. Meilă, "Comparing clusterings?an information based distance," *Journal of multivariate analysis*, vol. 98, no. 5, pp. 873–895, 2007.
- [222] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *Journal of intelligent information systems*, vol. 17, no. 2-3, pp. 107–145, 2001.
- [223] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. 2nd. 2001.
- [224] A. Zimek, "Correlation clustering," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 53–54, 2009.
- [225] R. A. Johnson, D. W. Wichern, *et al.*, *Applied multivariate statistical analysis*, vol. 5. Prentice hall Upper Saddle River, NJ, 2002.
- [226] O. Ledoit and M. Wolf, "Spectrum estimation: A unified framework for covariance

- matrix estimation and pca in large dimensions,” *Journal of Multivariate Analysis*, vol. 139, pp. 360–384, 2015.
- [227] B. S. Duran and P. L. Odell, *Cluster analysis: a survey*, vol. 100. Springer Science & Business Media, 2013.
- [228] K. Sim, V. Gopalkrishnan, A. Zimek, and G. Cong, “A survey on enhanced subspace clustering,” *Data mining and knowledge discovery*, vol. 26, no. 2, pp. 332–397, 2013.
- [229] J. Feng, Z. Lin, H. Xu, and S. Yan, “Robust subspace segmentation with block-diagonal prior,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3818–3825, Columbus, OH, USA, IEEE, 23-28 June 2014.
- [230] Y. Kim and M. Mesbahi, “On maximizing the second smallest eigenvalue of a state-dependent graph laplacian,” *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.
- [231] V. Deufemia, M. Giordano, G. Polese, and G. Tortora, “A visual language-based system for extraction–transformation–loading development,” *Software: Practice and Experience*, vol. 44, no. 12, pp. 1417–1440, 2014.
- [232] V. Granville, *Developing analytic talent: Becoming a data scientist*. John Wiley & Sons, 2014.
- [233] C. Xiao, J. Ye, R. M. Esteves, and C. Rong, “Using spearman’s correlation coefficients for exploratory data analysis on big dataset,” *Concurrency and Computation: Practice and Experience*, vol. 28, no. 14, pp. 3866–3878, 2016.
- [234] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

- [235] G. Gratton and A. Kolotilin, “Euclidean fairness and efficiency,” *Economic Inquiry*, vol. 53, no. 3, pp. 1689–1690, 2015.
- [236] S. Devarajan, “Africa’s statistical tragedy,” *Review of Income and Wealth*, vol. 59, no. S1, pp. S9–S15, 2013.
- [237] W. B. Group, *World Development Indicators 2012*. World Bank Publications, 2012.
- [238] P. Susantitaphong, D. N. Cruz, J. Cerda, M. Abulfaraj, F. Alqahtani, I. Koulouridis, and B. L. Jaber, “World incidence of aki: a meta-analysis,” *Clinical Journal of the American Society of Nephrology*, vol. 8, no. 9, pp. 1482–1493, 2013.
- [239] G. James, D. Witten, T. Hastie, and R. Tibshirani, “Unsupervised learning,” in *An Introduction to Statistical Learning*, pp. 373–418, Springer, 2013.
- [240] D. Sarkar, “Labels and legends,” *Lattice: Multivariate Data Visualization with R*, pp. 151–163, 2008.
- [241] G. Mouret, *No-Mean Clustering Algorithm*. PhD thesis, École Polytechnique de Montréal, 2014.
- [242] C. B. Hurley and R. Oldford, “Pairwise display of high-dimensional information via eulerian tours and hamiltonian decompositions,” *Journal of Computational and Graphical Statistics*, 2012.
- [243] J. Atkins and D. P. Sharma, “Visualization of babble–speech interactions using andrews curves,” *Circuits, Systems, and Signal Processing*, vol. 35, no. 4, pp. 1313–1331, 2016.
- [244] J. V. Kuzhali and S. Vengataasalam, “Hybrid model based feature selection approach

- using kernel pca for large datasets,” *International Review on Computers and Software (IRECOS)*, vol. 8, no. 11, pp. 2586–2592, 2013.
- [245] H. Abdi, “The kendall rank correlation coefficient,” *Encyclopedia of Measurement and Statistics*. Sage, Thousand Oaks, CA, pp. 508–510, 2007.
- [246] N. A. Weiss and C. A. Weiss, *Introductory statistics*. Pearson Education London, 2012.
- [247] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” in *ACM sigmod record*, vol. 29, pp. 1–12, ACM, 2000.
- [248] M. Deshpande and G. Karypis, “Item-based top-n recommendation algorithms,” *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.
- [249] J.-S. Lee, C.-H. Jun, J. Lee, and S. Kim, “Classification-based collaborative filtering using market basket data,” *Expert Systems with Applications*, vol. 29, no. 3, pp. 700–704, 2005.
- [250] A. Demiriz, “Enhancing product recommender systems on sparse binary data,” *Data Mining and Knowledge Discovery*, vol. 9, no. 2, pp. 147–170, 2004.
- [251] A. Gunawardana and G. Shani, “A survey of accuracy evaluation metrics of recommendation tasks,” *Journal of Machine Learning Research*, vol. 10, no. Dec, pp. 2935–2962, 2009.
- [252] M. Hahsler, “recommenderlab: A framework for developing and testing recommendation algorithms,” *Southern Methodist University*, 2011.
- [253] G. G. Chowdhury, *Introduction to modern information retrieval*. Facet publishing, 2010.

- [254] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, p. 19, 2016.
- [255] K. Verbert, H. Drachsler, N. Manouselis, M. Wolpers, R. Vuorikari, and E. Duval, “Dataset-driven research for improving recommender systems for learning,” in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, pp. 44–53, Alberta, Canada, ACM, February 27 - March 01 2011.
- [256] M. Shah, D. Parikh, and B. Deshpande, “Movie recommendation system employing latent graph features in extremely randomized trees,” in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, p. 42, Udaipur, India, ACM, March 04 - 05 2016.
- [257] S. Doods, A. Bellogín, T. D. Pessemier, and L. Martens, “A framework for dataset benchmarking and its application to a new movie rating dataset,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 7, no. 3, p. 41, 2016.

List of Publications

Journals

- 1) Divya Pandove, Shivani Goel and Rinkle Rani, “Correlation Clustering Methodologies and their Fundamental Results”, Expert Systems-SCIE Indexed, Impact factor: 1.18, DOI: 10.1111/exsy.12229
- 2) Divya Pandove, Shivani Goel and Rinkle Rani, “An Intuitive General Rank Based Correlation Coefficient”, Frontiers of Information Technology & Electronic Engineering-SCIE Indexed, DOI of ‘10.1631/FITEE.1601459’, Impact factor: 0.770
- 3) Divya Pandove, Shivani Goel and Rinkle Rani, “General Correlation Coefficient Based Agglomerative Clustering”, Cluster Computing-SCIE Indexed, Impact factor: 2.040 [Minor revision submitted]
- 4) Divya Pandove, Rinkle Rani and Shivani Goel, “Local graph based correlation clustering (LGBACC)”, Knowledge-Based Systems-SCIE Indexed, Impact factor: 4.529, DOI/10.1016/j.knosys.2017.09.034
- 5) Divya Pandove, Shivani Goel and Rinkle Rani, “Systematic Review of Clustering high-dimensional and Large Data Sets”, ACM Transactions on Knowledge Discovery in Data-SCIE Indexed, Impact factor: 1.98 [Accepted]

-
- 6) Divya Pandove, Rinkle Rani and Shivani Goel, “A Novel Correlation based Recommendation System for Big Data using SparkR”, IEEE Transactions on Knowledge and Data Engineering-SCIE Indexed, Impact factor: 4.18 [Under Review]

Conferences

- 1) Divya Pandove, Shivani Goel ,“A Comprehensive Study on Clustering Approaches for Big Data Mining”, International Conference on Electronics & Communication Systems, IEEE, pp. 1333-1338, February 2015
- 2) Divya Pandove, Shivani Goel , “Prototyping and in-Depth Analysis of Big Data Benchmarking”, 14th IEEE International Conference on Ubiquitous Computing and Communications, Liverpool, pp. 1222-1229, October 2015

Posters

- 1) Divya Pandove,“Predictive General Rank Based Correlation Coefficient”, Poster presented at Grace Hopper Conference (Houston- USA), 2016.
- 2) Divya Pandove, “The comparative analysis of various clustering techniques” Poster presented at the SPS 2015: 3rd Security and Privacy Symposium, IIIT Delhi, February 2015.

Appendix A

Methodologies and fundamental results of correlation clustering approaches

Methodology	Proposed By	Technique	Classification	Hardness	Run time	Results
Minimizing disagreements	Bansal, Chawla(2002)	Blum, Constant factor approx.	complete graph	NP complete	$(\frac{1}{\epsilon})k + 1$	Unweighted min. disagreement: $\min_c \sum_{i,j} c_{i,j}(1 - E_{ij}) + \sum_{i,j} (1 - c_{i,j})E_{i,j}$ Weighted min. disagreement: $\min_c \sum_{i,j} w_{ij}c_{ij}(1 - E_{ij}) + \sum_{i,j} w_{ij}(1 - c_{ij})E_{ij}$
Maximizing agreements	Bansal, Chawla(2002)	Blum, PTAS	complete graph	NP Hard	$(\frac{1}{\epsilon})k + 1$, polynomial in n	Unweighted max. disagreement: $\max_c \sum_{i,j} C_{ij}E_{ij} + \sum_{i,j} (1 - c_{ij})(1 - E_{ij})$ Weighted max. disagreement: $\max_c \sum_{i,j} w_{ij}C_{ij}E_{ij} + \sum_{i,j} w_{ij}(1 - C_{ij})(1 - E_{ij})$
Clustering in weighted Graphs	Demaine, Emanuel, Fiat and Nicole(2006)	LP	general graph	APX Hard	$O(\log n)$	LP formulation: $\minimize \sum_{e \in E} -c_e(1 - x_e) + \sum_{e \in E} c_e x_e$ region Growing: $\min((d_{uv} - r)) > 0 : v \notin B(u, r)$
Clustering using qualitative info	Charikar, Venkatesan and Wirth(2003)	LP	General or complete Graph	APX Hard	$O(\log n)OPT_{LP}$	4 factor approx. for minimization in complete graphs. 0.7664 approximation for general graphs. For min disagree: $\minimize \sum_{(ij)} w_{ij}x_{ij} + \sum_{-(ij)} w_{ij}(1 - x_{ij})$ For max agree: $\maximize \sum_{(ij)} w_{ij}(v_i.v_j) + \sum_{-(ij)} w_{ij}(1 - v_i.v_j)$
Clustering with fixed clusters	Goitis and Guruswami (2006)	PTAS	General graph	NP Hard	$O(\sqrt{\log n})$	Develop MINDISAGREE[k] and prove that it is better than MAXAGREE[k]. Provide a PTAS for the $k = 2$ version by using the algorithm MAXAGREE[k]. For MAXAGREE[2] the polynomial time factor is 0.878 and for MAXAGREE[k] the it is 0.7666.]
Clustering with random pivot	Alison, Charikar and Newman(2008)	LP, 3 factor approx.	general graph	—	—	The LP for weighted correlation clustering: $\minimize \sum_{i < j} x_{ij}^+ w_{ji}^- + x_{ji}^- w_{ij}^+ s.t.$ $x_{jk}^- \leq x_{ij}^- + x_{jk}^- \forall i, j, k$ $x_{ij}^+ + x_{ij}^- = 1 \forall i \neq j$ $x_{ij}^-, x_{ij}^+ \geq 0 \forall i \neq j$
Clustering with noisy input	Mathieu and Shudy(2010)	semi definite programming relaxation	general graph	NP Hard	$O(\log n)$	Results for worst case models, semi random models and random models. Semi random model: $E_{alg}[cost(OUT)] \leq OPT + O(n^{-\frac{1}{6}})OPT_{full}$ fully random model: $E_{alg}[Cost(OUT)] \leq L + o(L)$ They give one circumstance in which the there will be an exact reconstruction of the base clustering. They analyse a small noise regime for applications having rare perturbations like mutations.
Clustering with partial information	Demaine and Immorlica(2003)	linear programming	general graph	APX Hard	$O(\log n)$	They give an $O(\log n)$ approximation algorithm(for general graphs) and for graphs excluding the complete bipartite graphs, they deduce an $O(r^3)$ approximation algorithm.

Appendix B

Value of D for a given value of n

n	q	r	d	n	q	r	d	n	q	r	d	n	q	r	d
1	0	1	-1	126	31	2	6015	251	62	3	23810	376	94	0	53016
2	0	2	1	127	31	3	6140	252	63	0	23814	377	94	1	53391
3	0	3	2	128	32	0	6144	253	63	1	24065	378	94	2	53769
4	1	0	6	129	32	1	6271	254	63	2	24319	379	94	3	54146
5	1	1	9	130	32	2	6401	255	63	3	24572	380	95	0	54150
6	1	2	15	131	32	3	6530	256	64	0	24576	381	95	1	54529
7	1	3	20	132	33	0	6534	257	64	1	24831	382	95	2	54911
8	2	0	24	133	33	1	6665	258	64	2	25089	383	95	3	55292
9	2	1	31	134	33	2	6799	259	64	3	25346	384	96	0	55296
10	2	2	41	135	33	3	6932	260	65	0	25350	385	96	1	55679
11	2	3	50	136	34	0	6936	261	65	1	25609	386	96	2	56065
12	3	0	54	137	34	1	7071	262	65	2	25871	387	96	3	56450
13	3	1	65	138	34	2	7209	263	65	3	26132	388	97	0	56454
14	3	2	79	139	34	3	7346	264	66	0	26136	389	97	1	56841
15	3	3	92	140	35	0	7350	265	66	1	26399	390	97	2	57231
16	4	0	96	141	35	1	7489	266	66	2	26665	391	97	3	57620
17	4	1	111	142	35	2	7631	267	66	3	26930	392	98	0	57624
18	4	2	129	143	35	3	7772	268	67	0	26934	393	98	1	58015
19	4	3	146	144	36	0	7776	269	67	1	27201	394	98	2	58409
20	5	0	150	145	36	1	7919	270	67	2	27471	395	98	3	58802
21	5	1	169	146	36	2	8065	271	67	3	27744	396	99	0	58806
22	5	2	191	147	36	3	8210	272	68	0	27744	397	99	1	59201
23	5	3	212	148	37	0	8214	273	68	1	28015	398	99	2	59599
24	6	0	216	149	37	1	8361	274	68	2	28289	399	99	3	59996
25	6	1	239	150	37	2	8511	275	68	3	28562	400	100	0	60000
26	6	2	265	151	37	3	8660	276	69	0	28566	401	100	1	60399
27	6	3	290	152	38	0	8664	277	69	1	28841	402	100	2	60801
28	7	0	294	153	38	1	8815	278	69	2	29119	403	100	3	61202
29	7	1	321	154	38	2	8969	279	69	3	29396	404	101	0	61206
30	7	2	351	155	38	3	9122	280	70	0	29400	405	101	1	61609
31	7	3	380	156	39	0	9126	281	70	1	29679	406	101	2	62015
32	8	0	384	157	39	1	9281	282	70	2	29961	407	101	3	62420
33	8	1	415	158	39	2	9439	283	70	3	30242	408	102	0	62424
34	8	2	449	159	39	3	9596	284	71	0	30246	409	102	1	62831
35	8	3	482	160	40	0	9600	285	71	1	30529	410	102	2	63241
36	9	0	486	161	40	1	9759	286	71	2	30815	411	102	3	63650
37	9	1	521	162	40	2	9921	287	71	3	31100	412	103	0	63654
38	9	2	559	163	40	3	10082	288	72	0	31104	413	103	1	64065
39	9	3	596	164	41	0	10086	289	72	1	31391	414	103	2	64479
40	10	0	600	165	41	1	10249	290	72	2	31681	415	103	3	64892
41	10	1	639	166	41	2	10415	291	72	3	31970	416	104	0	64896
42	10	2	681	167	41	3	10580	292	73	0	31974	417	104	1	65311
43	10	3	722	168	42	0	10584	293	73	1	32265	418	104	2	65729
44	11	0	726	169	42	1	10751	294	73	2	32559	419	104	3	66146
45	11	1	769	170	42	2	10921	295	73	3	32852	420	105	0	66150
46	11	2	815	171	42	3	11090	296	74	0	32856	421	105	1	66569
47	11	3	860	172	43	0	11094	297	74	1	33151	422	105	2	66991
48	12	0	864	173	43	1	11265	298	74	2	33449	423	105	3	67412
49	12	1	911	174	43	2	11439	299	74	3	33746	424	106	0	67416
50	12	2	961	175	43	3	11612	300	75	0	33750	425	106	1	67839
51	12	3	1010	176	44	0	11616	301	75	1	34049	426	106	2	68265
52	13	0	1014	177	44	1	11791	302	75	2	34351	427	106	3	68690
53	13	1	1065	178	44	2	11969	303	75	3	34652	428	107	0	68694
54	13	2	1119	179	44	3	12146	304	76	0	34656	429	107	1	69121
55	13	3	1172	180	45	0	12150	305	76	1	34959	430	107	2	69551
56	14	0	1176	181	45	1	12329	306	76	2	35265	431	107	3	69980
57	14	1	1231	182	45	2	12511	307	76	3	35570	432	108	0	69984
58	14	2	1289	183	45	3	12692	308	77	0	35574	433	108	1	70415
59	14	3	1346	184	46	0	12696	309	77	1	35881	434	108	2	70849
60	15	0	1350	185	46	1	12879	310	77	2	36191	435	108	3	71282
61	15	1	1409	186	46	2	13065	311	77	3	36500	436	109	0	71286
62	15	2	1471	187	46	3	13250	312	78	0	36504	437	109	1	71721
63	15	3	1532	188	47	0	13254	313	78	1	36815	438	109	2	72159
64	16	0	1536	189	47	1	13441	314	78	2	37129	439	109	3	72596
65	16	1	1599	190	47	2	13631	315	78	3	37442	440	110	0	72600
66	16	2	1665	191	47	3	13820	316	79	0	37446	441	110	1	73039
67	16	3	1730	192	48	0	13824	317	79	1	37761	442	110	2	73481
68	17	0	1734	193	48	1	14015	318	79	2	38079	443	110	3	73922
69	17	1	1801	194	48	2	14209	319	79	3	38396	444	111	0	73926
70	17	2	1871	195	48	3	14402	320	80	0	38400	445	111	1	74369
71	17	3	1940	196	49	0	14406	321	80	1	38719	446	111	2	74815
72	18	0	1944	197	49	1	14601	322	80	2	39041	447	111	3	75260
73	18	1	2015	198	49	2	14799	323	80	3	39362	448	112	0	75264
74	18	2	2089	199	49	3	14996	324	81	0	39366	449	112	1	75711
75	18	3	2162	200	50	0	15000	325	81	1	39689	450	112	2	76161
76	19	0	2166	201	50	1	15199	326	81	2	40015	451	112	3	76610
77	19	1	2241	202	50	2	15401	327	81	3	40340	452	113	0	76614
78	19	2	2319	203	50	3	15602	328	82	0	40344	453	113	1	77045
79	19	3	2396	204	51	0	15606	329	82	1	40671	454	113	2	77519
80	20	0	2400	205	51	1	15809	330	82	2	41001	455	113	3	77972
81	20	1	2479	206	51	2	16015	331	82	3	41330	456	114	0	77976
82	20	2	2561	207	51	3	16220	332	83	0	41334	457	114	1	78431
83	20	3	2642	208	52	0	16224	333	83	1	41665	458	114	2	78889
84	21	0	2646	209	52	1	16431	334	83	2	41999	459	114	3	79346
85	21	1	2729	210	52	2	16641	335	83	3	42332	460	115	0	79350
86	21	2	2815	211	52	3	16850	336	84	0	42336	461	115	1	79809
87	21	3	2900	212	53	0	16854	337	84	1	42671	462	115	2	80271
88	22	0	2904	213	53	1	17065	338	84	2	43009	463	115	3	80732
89	22	1	2991	214	53	2	17279	339	84	3	43346	464	116	0	80736
90	22	2	3081	215	53	3	17492	340	85	0	43350	465	116	1	81199
91	22	3	3170	216	54	0	17496	341	85	1	43689	466	116	2	81665
92	23	0	3174	217	54	1	17711	342	85	2	44031	467	116	3	82130
93	23	1	3265	218	54	2	17929	343	85	3	44372	468	117	0	82134
94	23	2	3359	219	54	3	18146	344	86	0	44376	469	117	1	82601
95	23	3	3452	220	55	0	18150	345	86	1	44719	470	117	2	83071
96	24	0	3456	221	55	1	18369	346	86	2	45065	471	117	3	83540
97	24	1	3551	222	55	2	18591	347							

n	q	r	d	n	q	r	d	n	q	r	d	n	q	r	d
501	125	1	94249	626	156	2	147265	751	187	3	212060	876	219	0	287766
502	125	2	94751	627	156	3	147890	752	188	0	212064	877	219	1	288641
503	125	3	95252	628	157	0	147894	753	188	1	212815	878	219	2	289519
504	126	0	95256	629	157	1	148521	754	188	2	213569	879	219	3	290396
505	126	1	95759	630	157	2	149151	755	188	3	214322	880	220	0	290400
506	126	2	96265	631	157	3	149780	756	189	0	214326	881	220	1	291279
507	126	3	96770	632	158	0	149784	757	189	1	215081	882	220	2	292161
508	127	0	96774	633	158	1	150415	758	189	2	215839	883	220	3	293042
509	127	1	97281	634	158	2	151049	759	189	3	216596	884	221	0	293046
510	127	2	97791	635	158	3	151682	760	190	0	216600	885	221	1	293929
511	127	3	98300	636	159	0	151686	761	190	1	217359	886	221	2	294815
512	128	0	98304	637	159	1	152321	762	190	2	218121	887	221	3	295700
513	128	1	98815	638	159	2	152959	763	190	3	218882	888	222	0	295704
514	128	2	99329	639	159	3	153596	764	191	0	218886	889	222	1	296591
515	128	3	99842	640	160	0	153600	765	191	1	219649	890	222	2	297481
516	129	0	99846	641	160	1	154239	766	191	2	220415	891	222	3	298370
517	129	1	100361	642	160	2	154881	767	191	3	221180	892	223	0	298374
518	129	2	100879	643	160	3	155522	768	192	0	221184	893	223	1	299265
519	129	3	101396	644	161	0	155526	769	192	1	221951	894	223	2	300159
520	130	0	101400	645	161	1	156169	770	192	2	222721	895	223	3	301052
521	130	1	101919	646	161	2	156815	771	192	3	223490	896	224	0	301056
522	130	2	102441	647	161	3	157460	772	193	0	223494	897	224	1	301951
523	130	3	102962	648	162	0	157464	773	193	1	224265	898	224	2	302849
524	131	0	102966	649	162	1	158111	774	193	2	225039	899	224	3	303746
525	131	1	103489	650	162	2	158761	775	193	3	225812	900	225	0	303750
526	131	2	104015	651	162	3	159410	776	194	0	225816	901	225	1	304649
527	131	3	104540	652	163	0	159414	777	194	1	226591	902	225	2	305551
528	132	0	104544	653	163	1	160065	778	194	2	227369	903	225	3	306452
529	132	1	105071	654	163	2	160719	779	194	3	228146	904	226	0	306456
530	132	2	105601	655	163	3	161372	780	195	0	228150	905	226	1	307359
531	132	3	106130	656	164	0	161376	781	195	1	228929	906	226	2	308265
532	133	0	106134	657	164	1	162031	782	195	2	229711	907	226	3	309170
533	133	1	106665	658	164	2	162689	783	195	3	230492	908	227	0	309174
534	133	2	107199	659	164	3	163346	784	196	0	230496	909	227	1	310081
535	133	3	107732	660	165	0	163350	785	196	1	231279	910	227	2	310991
536	134	0	107736	661	165	1	164009	786	196	2	232065	911	227	3	311900
537	134	1	108271	662	165	2	164671	787	196	3	232850	912	228	0	311904
538	134	2	108809	663	165	3	165332	788	197	0	232854	913	228	1	312815
539	134	3	109346	664	166	0	165336	789	197	1	233641	914	228	2	313729
540	135	0	109350	665	166	1	165999	790	197	2	234431	915	228	3	314642
541	135	1	109889	666	166	2	166665	791	197	3	235220	916	229	0	314646
542	135	2	110431	667	166	3	167330	792	198	0	235224	917	229	1	315561
543	135	3	110972	668	167	0	167334	793	198	1	236015	918	229	2	316479
544	136	0	110976	669	167	1	168001	794	198	2	236809	919	229	3	317396
545	136	1	111519	670	167	2	168671	795	198	3	237602	920	230	0	317400
546	136	2	112065	671	167	3	169344	796	199	0	237606	921	230	1	318319
547	136	3	112610	672	168	0	169344	797	199	1	238401	922	230	2	319241
548	137	0	112614	673	168	1	170015	798	199	2	239199	923	230	3	320162
549	137	1	113161	674	168	2	170689	799	199	3	239996	924	231	0	320166
550	137	2	113711	675	168	3	171362	800	200	0	240000	925	231	1	321089
551	137	3	114260	676	169	0	171366	801	200	1	240799	926	231	2	322015
552	138	0	114264	677	169	1	172041	802	200	2	241601	927	231	3	322940
553	138	1	114815	678	169	2	172719	803	200	3	242402	928	232	0	322944
554	138	2	115369	679	169	3	173396	804	201	0	242406	929	232	1	323871
555	138	3	115922	680	170	0	173400	805	201	1	243209	930	232	2	324801
556	139	0	115926	681	170	1	174079	806	201	2	244015	931	232	3	325730
557	139	1	116481	682	170	2	174761	807	201	3	244820	932	233	0	325734
558	139	2	117039	683	170	3	175442	808	202	0	244824	933	233	1	326665
559	139	3	117596	684	171	0	175446	809	202	1	245631	934	233	2	327599
560	140	0	117600	685	171	1	176129	810	202	2	246441	935	233	3	328532
561	140	1	118159	686	171	2	176815	811	202	3	247250	936	234	0	328536
562	140	2	118721	687	171	3	177500	812	203	0	247254	937	234	1	329471
563	140	3	119282	688	172	0	177504	813	203	1	248065	938	234	2	330409
564	141	0	119286	689	172	1	178191	814	203	2	248879	939	234	3	331346
565	141	1	119849	690	172	2	178881	815	203	3	249692	940	235	0	331350
566	141	2	120415	691	172	3	179570	816	204	0	249696	941	235	1	332289
567	141	3	120980	692	173	0	179574	817	204	1	250511	942	235	2	333231
568	142	0	120984	693	173	1	180265	818	204	2	251329	943	235	3	334172
569	142	1	121551	694	173	2	180959	819	204	3	252146	944	236	0	334176
570	142	2	122121	695	173	3	181652	820	205	0	252150	945	236	1	335119
571	142	3	122690	696	174	0	181656	821	205	1	252969	946	236	2	336065
572	143	0	122694	697	174	1	182351	822	205	2	253791	947	236	3	337010
573	143	1	123265	698	174	2	183049	823	205	3	254612	948	237	0	337014
574	143	2	123839	699	174	3	183746	824	206	0	254616	949	237	1	337961
575	143	3	124412	700	175	0	183750	825	206	1	255439	950	237	2	338911
576	144	0	124416	701	175	1	184449	826	206	2	256265	951	237	3	339860
577	144	1	124991	702	175	2	185151	827	206	3	257090	952	238	0	339864
578	144	2	125569	703	175	3	185852	828	207	0	257094	953	238	1	340815
579	144	3	126146	704	176	0	185856	829	207	1	257921	954	238	2	341769
580	145	0	126150	705	176	1	186559	830	207	2	258751	955	238	3	342722
581	145	1	126729	706	176	2	187265	831	207	3	259580	956	239	0	342726
582	145	2	127311	707	176	3	187970	832	208	0	259584	957	239	1	343681
583	145	3	127892	708	177	0	187974	833	208	1	260415	958	239	2	344639
584	146	0	127896	709	177	1	188681	834	208	2	261249	959	239	3	345596
585	146	1	128479	710	177	2	189391	835	208	3	262082	960	240	0	345600
586	146	2	129065	711	177	3	190100	836	209	0	262086	961	240	1	346559
587	146	3	129650	712	178	0	190104	837	209	1	262921	962	240	2	347521
588	147	0	129654	713	178	1	190815	838	209	2	263759	963	240	3	348482
589	147	1	130241	714	178	2	191529	839	209	3	264596	964	241	0	348486
590	147	2	130831	715	178	3	192242	840	210	0</					