Feature Subset Selection Based on Consensus Clustering

by D. Sandhya Rani

Submission date: 29-Dec-2021 10:34AM (UTC+0530)
Submission ID: 1736208168
File name: 10MCPC15.pdf (756.77K)
Word count: 22736
Character count: 114318

Feature Subset Selection based on Consensus Clustering

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Award of Degree of

Doctor of Philosophy

in

Computer Science

by

D. Sandhya Rani

Reg. No. 10MCPC15

School of Computer and Information Sciences

University of Hyderabad,

(P. O.) Central University, **GachiBowli, Hyderabad – 500 046, India.**

December 29, 2021

Dedicated to Almighty & my beloved husband and daughter

CERTIFICATE

This is to certify that the thesis entitled **Feature Subset Selection based on Consensus Clustering** submitted by **D. Sandhya Rani** Reg. No. **10MCPC15**, in partial fulfillment of the requirements for award of **Doctor of Philosophy** in **Computer Science** is a bonafide work carried out by her under our supervision.

This thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other University or Institution for award of any degree or diploma. Parts of this thesis have been presented in the following conferences:

1. D. SANDHYA RANI, T. SOBHA RANI, S. DURGA BHAVANI. **Consensus Clustering for Dimensionality Reduction**. *Seventh International Conference on Contemporary Computing (IC3), 2014*, pages 148-153, IEEE 2014.
   doi: 10.1109/IC3.2014.6897164.

2. D. SANDHYA RANI, T. SOBHA RANI, S. DURGA BHAVANI. **Feature subset selection using consensus clustering**. *Eighth International Conference on Advances in Pattern Recognition (ICAPR), 2015*, pages 1-6, IEEE 2015.
   doi: 10.1109/ICAPR.2015.7050659.

Subjects passed for fulfillment of the course work

Sl.No	Code	Name of the Subject	Result
1	CS801	Data structures & Algorithms	Pass
2	CS802	Operating System and Programming	Pass
3	AI875	Trends in Soft Computing	Pass
4	AI879	Data Mining	Pass

Dr. T. Sobha Rani Dr. S. Durga Bhavani
Supervisors
School of CIS,
University of Hyderabad.

Prof. Chakravarthy Bhagvati
Dean
School of CIS,
University of Hyderabad.

DECLARATION

I, D. Sandhya Rani, Reg. No. 10MCPC15, hereby declare that this thesis entitled **Feature Subset Selection based on Consensus Clustering** submitted by me under the supervision of Dr. T. Sobha Rani and Dr. S. Durga Bhavani, School of Computer and Information Sciences, University of Hyderabad is a bonafide research work. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma.

Date: **D. Sandhya Rani**

10MCPC15

Signature of the Student

// Countersigned //

Signature of the Supervisors

Dr. T. Sobha Rani Dr. S. Durga Bhavani

Acknowledgements

I am indebted to my Research Supervisor Dr.T. Sobha Rani, Associate Professor, University of Hyderabad for her vital encouragement, support, and freedom provided in my research. I have no words to mention her consistent interest that made this research to take this form apart from the fruitful knowledge gained.

It gives me great pleasure and pride to express my deep sense of gratitude for my Research Co-Supervisor and Dr. S. Durga Bhavani, Professor, University of Hyderabad for her expertise and inspiring guidance. I proclaim my indebtedness to her for constant encouragement with useful suggestions during the entire tenure of this work.

It's my privilage and pride to express my gratitude to my DRC members Dr. Atul Negi, Professor, and Dr. Siba K. Udgata, Professor, UoH. I am very much thankful to them for their valuable suggestions during DRC meetings. I consider my self-fortunate in that it would have been impossible to achieve this goal without their support and care.

Furthermore, my heartfelt gratitude and sincere thanks to Advisor Dr. C. Madhusudhana Reddy and Chairman Dr. C.V. Raghava, CVR College of Engineering for the constant encouragement and support they have extended in pursuing my research work. I thank my colleague Dr. M. Raghava for his insightful comments and suggestions. I would like to offer my special thanks to Mr. Ramesh and Mr. K. Monachary for helping me in writing code. I thank my friend Mrs. C. V. Krishnaveni for her encouragement and support all through my research.

Words fail to express my appreciation to my husband Dr. G. BalaKrishna for his understanding, patience and helping me constantly throughout my research work. I also thank little princess of my life my daughter Himani for her well understanding and to not disturb me during entire tenure of research. I could not have completed my research without the support of all these wonderful people! ...

Publications

1. D. SANDHYA RANI, T. SOBHA RANI, S. DURGA BHAVANI. **Consensus Clustering for Dimensionality Reduction**. *Seventh International Conference on Contemporary Computing (IC3), 2014*, pages 148-153, IEEE 2014.
   doi: 10.1109/IC3.2014.6897164.

2. D. SANDHYA RANI, T. SOBHA RANI, S. DURGA BHAVANI. **Feature subset selection using consensus clustering**. *Eighth International Conference on Advances in Pattern Recognition (ICAPR), 2015*, pages 1-6, IEEE 2015.
   doi: 10.1109/ICAPR.2015.7050659.

Abstract

The main goal of the feature selection algorithms is to select minimal number of features, while retaining good classification accuracy. The feature subset selection problem is an NP hard problem since the best feature subset needs to be selected from the original set. There is a need for computationally efficient algorithms that find near optimal feature subsets. Scalability and reduction in the number of features are some of the major concerns of the feature selection algorithms in the literature. Specially when dealing with big data that contains huge number of redundant and irrelevant features, it becomes a great challenge to obtain the optimal feature subset while retaining good classifier accuracy.

Different algorithms may give different feature subsets for a dataset which 'cluster' or 'classify' the data well. In this situation, can a consensus among the different subsets of features describe the data better? This motivates us to use the idea of consensus clustering for feature subset selection. The goal of this work is to propose efficient algorithms that work on small as well as large datasets.

We propose three new approaches based on genetic algorithms (GACC), community discovery(CDCC) algorithms and feature ranking (FRCC) algorithms that generate feature subsets. In each of these approaches, Best-of-k(BoK) consensus clustering algorithm is used to arrive at the final feature subset. To the best of our knowledge, consensus clustering has not been used for feature subset selection in the literature.

In Genetic algorithm based consensus clustering(GACC), feature subsets are represented as chromosomes. Consensus clustering is used to identify the best feature subset. The results obtained on benchmark datasets are on par with the results available in the literature.

The CDCC approach works on the feature space rather than the original data space. The feature space is represented as a graph which is partitioned using the community

discovery algorithms available in the social networks literature. Consensus clustering algorithm is applied on the communities detected by the different community discovery algorithms to obtain the final feature subset. This method is implemented on several benchmark datasets. We obtain 50% reduction in the number of features selected and classifier accuracy is on par when compared to the latest literature.

A fast and scalable approach for feature selection FRCC has been designed using the available feature ranking algorithms from the literature. The feature weights of each ranking algorithm is treated as a one-dimensional space which is partitioned using K-means clustering algorithm. The consensus clustering algorithm obtains the best partitioning in which the top-weighted cluster is taken as the best feature subset which is further pruned to obtain the optimal feature subset by removing the redundant and irrelevant features. In addition to the small datasets, FRCC is tested on high-dimensional microarray data sets and it clearly outperforms many recent algorithms in the literature on small, big and large dimensional data sets.

FRCC is further extended to design a parallelizable algorithm to address feature reduction in big data. The algorithm Hybrid feature selection(HFS) has been tested on datasets with lakhs of instances and dimensionality in hundreds. HFS proves to be very effective in terms of feature reduction and accuracy in comparison to the results obtained by recent algorithms in the literature.

Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

PCA	**P**rinciple **C**omponent **A**nalysis
SBC	**S**elective **B**ayesian **C**lassifi
FRNN	**F**uzzy **R**ough **N**eural **N**etworks
BoK	**B**est (of) **K**
BOEM	**B**est **O**ne **E**lement **M**ove
NMI	**N**ormalized **M**utual **I**nformation
FCBF	**F**ast **C**orrelation **B**ased **F**ilter
SU	**S**ymmetric **U**ncertainty
mRMR	minimal **R**Redundancy **M**aximum **R**elavance
VI	**V**ariation (of) **I**nformation
SDD	**S**ymmetric **D**ifference **D**istance
ARI	**A**djusted **R**and **I**ndex
QPI	**Q**uality **P**artition **I**ndex

Chapter 1

Introduction

Application domains such as networks, e-commerce and bio-informatics are adding more and more data to the databases rapidly. The explosion of data is not only in terms of instances but also in the number of dimensions. In general, datasets with large number of dimensions with a fixed number of data points become increasingly "sparse" as the dimensionality increases [106]. The process of reducing the number of dimensions(or features) under consideration is called Feature reduction, also called as feature subset selection. This deals with the extraction of the relevant features from the given data and these approaches try to find a subset of the features which describes the data effectively [47].

In pattern recognition, it is helpful to choose a feature subset that best discriminates patterns belonging to different classes. The main goal of feature selection algorithms is to select minimal number of features, while retaining good classification accuracy. Any technique for high-dimensional data must deal with the "Curse of dimensionality", which is, as the number of dimensions increases, the performance of data analysis techniques will degrades. Many of the features may be irrelevant and redundant and they greatly affect the classifier accuracy [10].

Irrelevant features do not contribute to prediction and, may confuse the algorithm during classification. And redundant features do not add to the information already available in other features, thus they do not improve classifier accuracy. So, it is essential to eliminate irrelevant and redundant features. Thus, feature selection is a pre-processing technique that attempts to identify a subset of features.

The feature subset selection problem is an NP hard problem since the best feature

subset needs to be selected from the original set. In general, two basic dimensionality reduction techniques called feature selection and feature extraction are in use. Feature selection is the process of selecting best feature subset from the original set. The general approaches like wrapper and filter methods are used for feature subset selection that attempt to improve the classifier accuracy and predictor performance and also helps in better understanding of data [47].

Filter approach directly operates on the dataset and gives subset of features or ranking of features as output, where as wrapper approach uses learning algorithm to evaluate the performance of feature subset. Different learning criteria like classifier accuracy, distance measures have been used in the literature, to evaluate the feature goodness.

1.0.1 Gaps in the current literature for feature selection problem

Most of the traditional feature selection algorithms are taking significant learning time to find the best feature subset. Scalability and reduction in number of features are the major concerns in recent genetic algorithm based feature selection algorithm, even though classifier accuracy is high. Parallelizable algorithms like fuzzy rough set approaches are proposed in the recent literature with the aim of optimal usage of memory while reducing the run time. Many of these methods do not show much reduction in the number of features. Specially when dealing with big data that contains huge number of redundant and irrelevant features, it becomes a great challenge to obtain the optimal feature subset while retaining good classifier accuracy.

1.1 Motivation

As feature subset selection is an NP-hard problem, there is a need for an efficient approach for feature selection. Further, different algorithms may give different feature subsets for a dataset which 'cluster/classify' the data well. In this situation, can a consensus among the different subsets of features describe the data better? This motivates us to use the idea of consensus clustering for feature subset selection.

In addition, many features may be irrelevant and redundant. Such features can also affect the classifier accuracy. Hence, the final feature subset obtained using consensus clustering may have to be pruned. High dimensional data as well as large data are addi-

tional challenges for feature subset selection problem. Does working on the feature space rather than the instance space help in addressing the big data challenge? In this work, consensus clustering is used in each of the proposed algorithms to arrive at a nearly optimal feature subset.

1.1.1 Consensus clustering

In general, consensus clustering has been used in the literature to find the best clustering among various input clusterings (generated by different clustering algorithms). Here, we use this approach to find the best feature subset. There are many approximation algorithms that have been proposed for consensus clustering in the literature. One of the popular methods is BoK (Best of k) which chooses the best of the clusterings based on dissimilarity measures like Symmetric distance difference(SDD), adjusted rank index (ARI) etc. [6].

1.2 Problem statement

The problem is to select core set of features from the original feature set that describes the dataset well. This statement holds not only for big datasets, but also for small datasets.

1.2.1 Objectives

- To propose an efficient algorithm that works on small and large datasets.
- To propose a scalable approach to feature selection.
- To work on the feature space rather than data space.
- To apply consensus clustering algorithm for feature selection problem.
- To achieve a near optimal solution for the NP-hard problem.
- To achieve reduction in the feature subset.

1.3 Proposed algorithms

We propose three new approaches based on genetic algorithms (GA), community discovery algorithms from the area of social networks and feature ranking algorithms with K-

means clustering to find the feature subsets. For aggregation, we use a novel approach called consensus clustering [37, 30, 76, 112] to obtain the final feature subset. To the best of our knowledge, consensus clustering has not been used for feature subset selection.

1.3.1 Genetic algorithm based feature selection using consensus clustering (GACC)

A novel genetic algorithm based feature selection that uses consensus clustering is proposed. In this method, chromosomes (feature subsets) are produced at random in the initial population. The dataset is projected along each of the feature subsets specified by the chromosome, to which K-means algorithm is applied. The technique of consensus clustering is applied to the different clusterings of the dataset, in order to obtain the best partitioning of the dataset. The top two chromosomes are retained from each population based on the best-of-k consensus clustering technique. In the next population, the two best chromosomes are retained and crossover and mutation operations are applied to the remaining chromosomes. The method is continued until the same top chromosome as in the preceding population is picked. Time complexity for this algorithm is O(Npm), where 'N' instances, 'm' is the number of generations/populations, 'p' is number of chromosomes in each population.

1.3.2 Community discovery based feature selection using consensus clustering (CDCC)

Time can be saved by working on feature space rather than original data space, when the number of features is very small in comparison to the number of data instances. Feature subset selection methods try to select the most "representative" features that are highly correlated to the target class. On the basis of this concept, Song et al. developed an algorithm known as FAST[98]. Instead of using the data space, they used the feature space to construct a graph. With the inspiration from the FAST algorithm, we developed our second novel algorithm "Community discovery using consensus clustering" (CDCC).

As there are many graph partitioning algorithms available in the literature, final feature subset is not robust. To get a stable feature subset, we developed CDCC method which begins with the construction of a complete graph with features $\{f_i\}$ serving as vertices.

Pearson's correlation coefficient(PCC) r between features (f_i, f_j) is used to calculate the edge weight. The graph is partitioned using a few popular community discovery algorithms [31, 18]. Now, a consensus clustering algorithm BoK(Best-of-K) is used to select the best partitioning for the graph. Optimal feature subset is formed with the representative features from each cluster of the best partitioning.

1.3.3 Feature ranking based feature selection using consensus clustering (FRCC)

A fast and scalable approach for feature selection can be designed using the available feature ranking algorithms in the literature. The features are ranked in the order of importance, with the most significant feature being at the top of the list. Then, with the use of a threshold value, all the features whose values above a threshold will be treated as relevant and form the feature subset. But, as there are many feature ranking algorithms available in the literature and there may be differences in the rankings provided by the algorithms, making the final result unstable and also threshold selection has an impact on the output. We propose an approach in which the aggregation is carried out by applying consensus clustering.

The scalability and efficiency of this approach motivated us to apply this feature ranking method FRCC to feature selection in big data.

1.3.4 Hybrid feature selection (HFS)

Due to the scalability of FRCC algorithm, we apply this consensus method to the large scale domains like big data. Dataset is divided into samples and on each sample FRCC is applied to obtain feature subset from each sample. Final feature subset is obtained by performing union and intersection operations on feature subsets obtained from each sample.

1.4 Thesis Contributions

- Contribution 1: A genetic algorithm based feature selection using consensus clustering (GACC) is proposed. Genetic algorithm is a randomized approach which

searches the solution space to obtain a near optimal solution. It can be viewed as a proof of concept for consensus based feature selection.

- Contribution 2: Community discovery based feature selection using consensus clustering (CDCC) is proposed that eliminates irrelevant and redundant features successfully. This works for both numerical and categorical features. This algorithm works in the feature space rather than data space and hence is more scalable.

- Contribution 3: A scalable feature ranking based feature selection using consensus clustering (FRCC) is proposed. Due to its scalability, it can be implemented on high-dimensional datasets.

- Contribution 4: Hybrid feature selection based on consensus clustering technique is proposed. This is a parallelizable approach and can be successfully implemented on big-data applications.

1.5 Chapter Organization

Chapter 1 of the thesis is the introduction to feature subset selection problem and its importance in data mining applications. As, we are using consensus clustering for feature subset selection, fundamentals of consensus clustering are discussed here. Chapter organization and contributions of the thesis are presented.

Chapter 2 of the thesis presents the related literature of standard feature subset selection methods like Genetic algorithms, subset search methods, Ant-colony optimization methods etc., as well as the approximation algorithms for consensus clustering. Basic definitions and heuristics used in the thesis are described here. Time complexity of each consensus method is presented and justification for choosing BoK(Best-of-K) consensus method is also explained. A few algorithms for consensus clustering that have been proposed recently are also explained.

Chapter 3 contains our first contribution to feature subset selection. Here, we verified whether consensus clustering can be applied for feature selection problem by using one synthetic dataset. As we obtained good results, we developed a genetic algorithm based feature selection using consensus clustering algorithm(GACC) for feature selection. In

this chapter we present, a novel GACC algorithm, its time complexity and experiments conducted on it and finally the results.

Chapter 4 contains our second contribution. A novel community discovery based feature selection algorithm using consensus clustering is presented here. CDCC algorithm, its time complexity, and the experimentation set up along with the details of the results are described in this chapter.

Chapter 5 contains a new scalable feature subset selection algorithm with consensus clustering based on feature ranking methods(FRCC) is described here. For implementation, we consider a few challenging high-dimensional micro-array datasets from the literature. Detailed algorithm, time complexity and results are presented here. Further, a hybrid feature selection algorithm is proposed which can be applied to big data. Experimentation is carried out on large scale datasets and presented here.

Chapter 2

Related Literature

Feature selection has gained much importance in the field of data mining which selects a subset of relevant features for use in predictive model construction. The contributions of this thesis are all feature selection approaches, so here we mentioned some of the latest and classical methods available for feature selection in the literature. We compare our methods with genetic algorithm based methods, graph based methods, feature ranking and ensemble feature selection methods. Chapter organization is as follows: section 2.1 describes the basics of dimensionality reduction methods. Methods related to feature selection approach are explained in section 2.2. section 2.3 describes consensus clustering method and its related algorithms.

2.1 Dimensionality reduction

Dimensionality reduction techniques are usually divided into two groups. One is feature extraction and the other is feature selection.

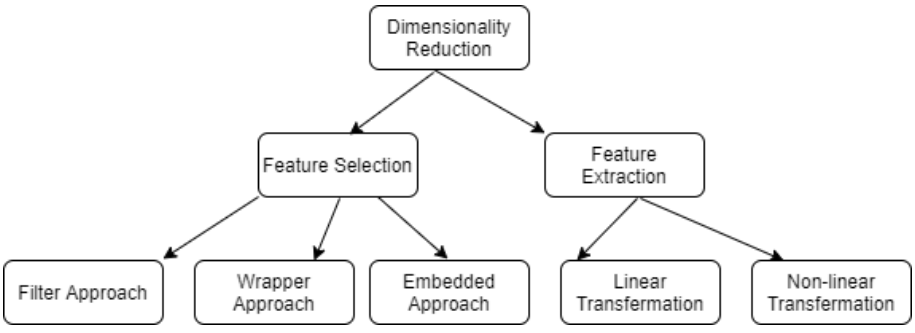

Figure 2.1: Dimensionality reduction methods.

2.1.1 Feature extraction

Feature extraction combines the original features to get reduced new set of features. One of the most popular methods is Principle Component Analysis(PCA). Principal Component Analysis (PCA), first introduced by Pearson [78], is one of the most common linear method for dimensionality reduction. PCA is a linear projection that minimises the average projection cost. The goal of this method is to find an orthogonal projection of the data in a low-dimensional linear subspace with the greatest variance. While PCA may be effective at reducing dimensionality, it has significant drawbacks.

- It is difficult to interpret new features.
- Data from different clusters may have varying feature correlations, and it may not always be possible to reduce too many dimensions without losing important information.
- Computing eigen vectors for very high-dimensional data is impossible.

2.1.2 Feature selection

Feature selection is a process of selecting important features while discarding the irrelevant and redundant features. Feature selection can be further classified into three approaches listed below:

1. **Filter approach:** Filters use the general features of training data to perform feature selection as a pre-processing step that is independent of the learning algorithm. This model is advantageous because of its minimal computational cost and ability to generalise.
2. **Wrapper approach:** Wrappers, which use a learning algorithm and evaluate the usefulness of subsets of features based on its prediction performance. In other words, the feature selection algorithm calls the learning algorithm as a subroutine to evaluate each subset of features, while reducing the computational cost on the learning process. The involvement of the classifier tends to produce higher performance results than filters.

3. **Embedded approach:** They do feature selection during the training process and are usually specific to a particular learning machine [33, 65]. As a result, the search for the best subset of features is embedded into the classifier's design and can be viewed as a search in the combined space of feature subsets and hypotheses. This method captures dependencies at a lower cost of computation than wrappers.

2.2 Feature selection methods

2.2.1 Genetic algorithm based feature selection methods

Since the subset selection problem is exponential in nature, getting the best solution is indeed very time consuming process. Metaheuristic is a high level framework that can be used to search for a near optimal solution in case of optimization problems involving subset selection. There are many metaheuristics available in literarure. Simulated annealing, Gentic algorithm, Ant colony optimization, particle swarm optimization and Tabu search are some of the popular metaheuristics.

Genetic algorithm(GA) is one of the popular feature selection methods that have been implemented successfully[61, 34, 16, 17, 91, 15]. Feature subsets can be formed efficiently using GAs, with the help of crossover and mutation operations and also it avoids exhaustive search for solution. One of the recent GA based algorithm is TCbGA[61] proposed by Benteng Ma and Yong Xia. It is a heuristic-guided stochastic and parallel approach. This method finds a global optimal feature subset by searching through a large number of feature combinations. Every chromosome in the population encodes a feature selection using binary encoding, which is then transmitted to the next generation. Fitness is determined by the accuracy of an SVM classifier obtained through the use of selected features. Ahn et al.,[34] used a genetic algorithm in the bankruptcy problem for case-based reasoning that simultaneously optimizes instance selection and feature weighing. A hybrid genetic algorithm is proposed by Kabir et al.[15] that uses specific local search method to find feature subset. Tsai et al.[16] has provided an extensive study of feature subset selection methods using genetic algorithms. Yang et al.[109] proposed GARIPPER algorithm, where fitness function is generated using classifier accuracy of RIPPER algorithm. In order to obtain the best feature subset the algorithm is repeated for a specific number of generations until the

expected classifier accuracy is achieved.

2.2.2 Graph based feature selection methods

Representation of feature space using graph and finding the feature groups is one of the best way to find optimal feature subset as this framework provides the underlying relationships between features or feature vectors. Laplacian score[38] and Fisher score[32] are some popular graph based feature selection methods in the literature. Graph clustering with ant-colony optimization(GCACO) [71] can be used to deal with features that are redundant or irrelevant. There are three steps in this process: i) The construction of a graph in feature space ii) Organizing the features into groups iii) find the optimal feature subset using ant-colony optimization. The time complexity increases in direct proportion to the number of features. Graph clustering with node centrality for feature election(GCNC) [70] is similar to GCACO, which is comprised of three steps. However, instead of using the ACO method, it makes use of term variance and node centrality to identify the most representative characteristics. Song et al.,[98] proposed FAST algorithm that contains three steps. In the first step, it removes irrelevant features using Symmetric Uncertainty measure. Then in the second step it constructs a minimum spanning tree(MST) from feature graph. Finally the minimum spanning tree is partitioned for selecting the representative features. The time complexity of the algorithm is O(Mlog^2M) where M denotes the count of features. Unsupervised feature selection method based on ant colony optimization(UFSACO) [92] is the Ant Colony Optimization(ACO) method used in this algorithm to determine the best feature subset. This algorithm falls under the "filter-based multivariate method". In order to reduce the redundancy among the selected features, UFSACO does not take into consideration any learning model.

Hypergraph based information theoretic approach for feature selection is proposed by Zhang et al[113]. In this method, a hypergraph is constructed on feature space and multi-dimensional interaction between features is used as edge weight. To find the feature subset hypergraph clustering algorithm is used. A non-redundant feature subset selection based on graph-theoretic approach is proposed by Mandal et al.[68]. In this method, feature subset selection is done by finding the in densest subgraph from weighted graph. Corresponding nodes(features of the densest subgraph will form final feature subset with non-redundant features. But relevancy of the features was not considered. So, they may not

represent best features. Bandopadhyay et al.[8] proposed an unsupervised feature subset selection method to overcome this problem by combining densest subgraph with feature clustering.

2.2.3 Ensemble feature ranking algorithms

There are some popular methods available in the literature based on ensemble approach. They are EnsRank [80], FRMV[40], EFR[45], FRSD [43] and so on. Ensemble ranking approach(EnsRank) assembles groups of rankers into a single entity. A single ensemble list is then constructed from the output of rankers by employing an aggregation function that assigns a "overall score" to each of the features in the ensemble. The final feature subset will be selected from the ordered ensemble feature list by applying a predefined threshold.

The FRMV [40] approach gathers various feature rankings from different views of the same data set, then combines all of the feature rankings into a single consensus one. FRMV [40] is typically able to determine a better feature ranking when compared to other feature ranking algorithms. Jong et al. [45] have proposed EFR method for feature ranking. In this paper, they use ensemble method for feature ranking, by combining feature rankings obtained by independent runs of the evolutionary algorithm ROC-based genetic learner.

The random subspace method is combined with the silhouette decomposition scheme in the FRSD algorithm [43]. The random subspace approach requires randomly sampling features and then building separate models in each subspace to create a large number of subspaces. Since the individual models are generated with a minimal number of features, the random subspace technique performs better when applied to high-dimensional data sets. The FRSD [43] builds cluster structures in each random subspace and estimates their average silhouette widths. These average widths are broken down into components that indicate how each attribute contributes to cluster formation.

Slavkov et al. [97] proposed an approach for analysing feature ranks. The method combines the concept of prediction model error to the "correctness" of feature ranking. In addition, the method is used to compare various ranking methodologies as well as various aggregation approaches like mean, median, min and max for merging feature ranks.

2.2.4 Feature ranking based approaches

The INTERACT [115] method uses SU measure as FCBF,but it additionally takes into account the consistency contribution, which is a measure of consistency after removing a feature. The algorithm is divided into two sections. First, based on their SU values features are arranged in descending order. Then, starting at the end of the feature rank list, features are examined one by one. If a feature's consistency contribution is less than a predetermined threshold, it is deleted; otherwise, it is selected. According to the authors, this technique can handle feature interaction and picks relevant features efficiently.

Lei et al. [82], have used symmetric uncertainty(SU) as a metric for finding feature correlation. The algorithm first calculates SU value of each feature and then using predefined threshold selects relevant features by ordering in decreasing order of SU values. Then, the ordered list is further processed for filtering redundant features. If M is the feature count and N is the count of instances then O(NMlogM) represents the complete time complexity of the algorithm. Another algorithm for finding feature correlations is FAST and it is also based on symmetric uncertainty measure(SU).

CFS (Correlation-based Feature Selection) [35] is a simple multivariate filter technique that ranks feature subsets using a heuristic evaluation function based on correlation. The evaluation function is biased in favour of subsets with attributes that are substantially correlated with the class but uncorrelated with one another. Irrelevant features will be discarded because their correlation with the class will be low. Further based on the correlation between the features, redundant features will be eliminated.

2.2.5 Fuzzy roughset neural network based feature selection

Fuzzy rough set approach is one of the populer technique used for feature selection [19, 20, 104]. It is a combination of fuzzy set and rough threory. A cloud computing technique DFRS [49] is a distributed fuzzy rough set (DFRS) based feature selection strategy that distributes computing jobs to different nodes. First, each node's capacity is determined by solving an optimized problem based on its processing and memory resources. The samples are then sent with the necessary interconnections using a lightweight data decomposition algorithm. Instead of doing individual computations, the dispersed nodes pool their resources to integrate global data and produce correct features. The main perspective of this

is work is to apply distributed technique on fuzzy rough set model.

Fuzzy decision tree(FDT) [96] algorithm starts by arranging the continuous values of features in the desired order to produce the “cut-point”. In the second step, the cut-point is ”fuzzified”, which is accomplished through the use of the entropy evaluation function. This step is performed on all attributes recursively in order to determine the best “cut-point ”. Then, in order to generate additional branches and nodes, the attribute with the lowest value is chosen. Once the stopping criterion is met, the process stops.

Zhao et al. [114] developed a feature selection approach using Fuzzy Rough Neural Networks (FRNN). A heuristic backward search approach is used to apply FRNN to feature selection. It employs both neural networks and feature selection. This algorithm has a time complexity of O($NlogN$).

2.2.6 Methods that deal with irrelevant and redundant features

Traditionally, the focus of feature subset selection method is on search for the relevant features. A well-known example is Relief [46]. This is a feature weighting algorithm proposed by Kira and Rendell. It is fast, easy to implement and accurate. But, it deals only with two-class data. This algorithm can remove only irrelevant features but does not identify redundant features. This cannot handle noisy data. ReliefF [64] is an extension of Relief, and can handle multi-class data and also noisy data. But this method cannot identify redundant features.

However, along with irrelevant features, redundant features also affect speed and accuracy of the algorithm. Methods like RRFS [24], FCBF [110], FAST [98], mRMR [79] and NMIFS [23] can eliminate irrelevant and also redundant features. Relevance-redundancy feature selection(RRFS) is based on the selection of relevant features and the elimination of redundant features. It selects a feature subset based on a specific threshold using mutual information based criteria.The time complexity of the RRFS method is log-linear in proportion to the number of features in the dataset.FCBF algorithm is based on correlation between features [110]. To find the correlation between features Lei et al. [] have used an *entropy* based measure *symmetric uncertainty(SU)*. First, it calculates SU value for each feature, selects relevant features based on predefined threshold and orders them in descending order according to their SU values. Then, it further processes ordered list to remove redundant features. If the number of features is M and number of instances is N

then, the overall time complexity of this algorithm is *O(NMlogM)*.

FAST algorithm also uses symmetric uncertainty measure(SU) to find the correlations. It consists of three steps [98]. First, it removes irrelevant features using SU measure as in FCBF. Then, constructs a minimum spanning tree(MST). Finally, partitions MST and selects representative features. The time complexity of this algorithm is O(Mlog^2M) with M number of features.

mRMR and NMIFS are incremental search algorithms which selects one feature at a time. mRMR and NMIFS, both use mutual information(MI) measure to find the optimal feature subset. Given a feature subset F with M features, the goal is to find a subset S with k features that maximizes MI(C; S) where k $<$ M. Both select first feature f_i that has a maximum MI(C; f_i), and selection criteria for remaining features is different.

2.2.7 Other methods

There are many other feature selection algorithms proposed in recent literature [92, 27, 67, 58, 66]. The following are some of the state-of-the-art methods for feature subset selection.

2.2.7.1 Bayesian Networks

Petri et al. [51] offer a data reduction technique for visualisation of high-dimensional data. They transform high-dimensional data vectors to low-dimensional data vectors using multidimensional scaling. They define unsupervised Bayesian distance measure, which is an extension of supervised Bayesian distance measure, to detect object similarity. The class-color clarity test and the Naive Bayesian classifier (NBC) are used to validate data presentation in 2D. However, for data sets where the Naive Bayesian classifier performs weakly, this test failed.

2.2.7.2 Selective Bayesian Network(SBC)

Ratanamahatana et al. [14] employed this classifier to increase the performance of a Naive Bayesian classifier when features are irrelevant or redundant. They used C4.5 to create the decision tree and chose the features that are the most significant in the first three layers of the decision tree. They did it five times and combined all of the attributes selected in each

iteration. To check the accuracy, a naive Bayesian classifier is used.

2.2.7.3 NBTree

This is a combination of the decision tree induction and naive Bayesian classifier, which is used to overcome the limitations of each separately. To improve the accuracy of a naive Bayesian classifier on high-dimensional datasets, Kohavi [48] proposed the NBTree algorithm. The results showed that the NBTree algorithm outperformed the Naive Bayesian (NB) and C4.5 decision tree induction algorithms.

2.2.7.4 SBPCA

In [93], Acharya formulated the Supervised Bergman PCA(SBPCA) dimensionality reduction technique and shown how it directly maximises the goal of prediction with reduced dimensions. It is also demonstrated that SBPCA outperforms PCA when classes are linearly separable.

2.3 Consensus clustering

Clustering is mainly used to group similar elements together. There are many clustering algorithms available in the literature [37], and different clustering algorithms may cluster the data differently. Then it is very difficult to judge which way of clustering the data is the best one.

Consensus clustering, also called aggregation of clustering or cluster ensembling [30], refers to obtain a single better clustering(consensus) from different clusterings for the same data set [76, 112]. Consensus clustering is thus the problem of reconciling clustering information about the same data set coming from different sources or from different runs of the same algorithm [99, 5], and also is known as median partition [25].

Basically clustering algorithms are sensitive to initial clustering settings, similarity measures used etc,[69]. To address these issues idea of consensus clustering has been proposed. Consensus clustering takes as input various clusterings generated by running the various clustering algorithms [30] or running same algorithm many times by changing initial input parameters to generate median partition. This method also can be used to represent the consensus over multiple runs of clustering algorithm with random restart

points, so as to deal with the problem of sensitivity to initial parameter settings. Since there can be exponentially many number of input clusterings, finding consensus among them is an NP-hard problem. Hence, many heuristics are proposed in the literature to address this problem[6].

2.3.0.1 Approximation algorithms for consensus clustering

Several algorithms are proposed to arrive at a consensus of the clusterings. There are basically two ways in which this is carried out. First option is to choose one of the clusterings as the consensus based on some dissimilarity measures. Second is to reorganize clusterings to arrive at a consensus. In literature, we could find many approximation algorithms like best-of-k (BoK), majority rule(MR), best one element move(BOEM), CC-Pivot, CCLP-Pivot, Average Link (AL), Furthest, simulated annealing etc.[6]. Algorithms based on graph partitioning HGPA, CSPA, MCLA [99] consensus clustering algorithms are based on graph partitioning [100].

All approximation algorithms in the literature use dissimilarity matrix to find the consensus. But, the time complexities of these approximation algorithms other than BoK are atleast O(N^2). And BoK has linear time complexity of O(k^2N), where 'k' is number of input clusterings and N is number of instances. It is a 2-approximation algorithm.

CC-Pivot [72] is designed on tournament graphs to solve the feedback arc set problem. It can also be used for consensus clustering and rank aggregation problems. This recursive method chooses a random pivot item P repeatedly and separates the objects based on their relationship with the pivot, similar to Quick sort. CCLP-Pivot is an extension to CC-Pivot that works for LP-problems. But the complexity of this problem is very high as O(N^8) for 'N' instances, due to triangle inequality constraints.

A standard agglomerative algorithm proposed by Gionis et al [3] is Average Linkage algorithm. Every object is placed in its own (singleton) cluster at the start. The two clusters with the shortest average distance between objects in one cluster to other are then repeatedly merged. This method is repeated until the average distance between each pair of clusters is at least 1/2. The time complexity of this algorithm is O($N^2(logN+M)$) with N veritices and M input clusterings. CSPA is a Cluster-based similarity partitioning algorithm, that uses pairwise similarity to recluster the objects using dissimilarity measure mentioned in subsubsection 2.3.0.2. HGPA is used to partition the hypergraphs. The ob-

jective of this method is to attain maximum mutual information. Clusters are represented with hyperedges. MCLA works by aggregating and collapsing related hyperedges.

2.3.0.2 Dissimilarity Measures

Various measures like dissimilarity measure [6][11], Quality Partition Index(QPI) [102] and Normalized Mutual Information(NMI) [99] are used in finding consensus clustering algorithms. Most of the works have used dissimilarity measure.

2.3.0.3 Symmetric distance difference (SDD)

Distance between two clusterings C_1 and C_2 can be calculated using:

$$d(C_1,C_2) = (b+c)\ or\ \binom{n}{2} - (a+d) \tag{2.3.1}$$

where

a = number of pairs of objects clustered in C_1 and C_2

b = number of pairs of objects clustered inC_1 but not clustered in C_2

c = number of pairs of objects clustered in C_2 but not clustered in C_1

d = number of pairs of objects not clustered in C_1 and C_2

With given input clusterings (C_1, C_2, C_3,.....C_k) we need to find a partitioning C^* such that

$$C^* = argmin_C \sum_{i=1}^{k} d(C_i,C) \tag{2.3.2}$$

The best clustering is the one which has minimum dissimilarity from all remaining clusterings. We use best-of-k(BoK) approximation algorithm because of its linear time complexity of O(k^2N)[6], where 'k' is number of input clusterings and 'n' is number of instances. It picks best clustering from k- input clusterings. This is a 2-approximation algorithm [26].

2.3.0.4 Adjusted Rand Index (API)

This is derived form original rand index [86] measure. If X and Y are the two partitioning or clusters, ARI can be defined using contingency table of size nXn with each entry $[n_{ij}]$ represents number of common objects in X_i and Y_j.

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{2.3.3}$$

where a_i is the sum of $[n_{ij}]$ row wise, and b_j is the sum of $[n_{ij}]$ column wise in contingency table.

2.3.0.5 Normalized Mutual Information (NMI)

It is an entropy based measure [95] used to evaluate the quality of the clustering. From the given set of input clusterings, the one which has highest NMI score will be selected as best clustering. If P, C denote labels of class and clusters respectively then NMI can be defined as follows:

$$NMI(P,C) = \frac{2 \times I(P;C)}{H(P) + H(C)} \tag{2.3.4}$$

Where P, C represents class and cluster labels respectively. $H(.)$ denotes entropy. $I(P;C)$ denotes mutual information between P and C.

2.3.1 Recent consensus clustering algorithms

Recently, Huang el al. [41] proposed weighted consensus clustering algorithms namely LWEA and LWGP. Banerjee et al. [9] have developed WHAC which is an improved version of LWEA and further a new coupled ensemble selection method (CES) is proposed. Basically, the performance of consensus clustering algorithms depends on reliability of input clusterings in the ensemble. LWEA and LWGP both deal with this problem using cluster uncertainty estimation (ECI) and local weighting strategy. LWGP is based on bipartite graph formulation and partitioning. Tcut [111] is used to partition the graph into disjoint sets called clusters. WHAC is a hierarchical consensus clustering algorithm used to learn the cluster ensemble. To evaluate the merit of the clustering, a cluster-level sur-

prisal measure is defined. Results shows that WHAC is performing better compared to LWEA. The time complexity of WHAC is $\Theta(N^2Pk^2 + N^2logN)$, where N, P, k are number of instances, true number of clusters and size of ensemble. Further, CES gives a new direction for theoretical research in quality ensemble selection.

In this chapter we gave an overview of dimensionality reduction problem and importance of feature selection. Some of the recent methods for feature selection problem are mentioned here along with their advantages and disadvantages. As we are addressing feature subset selection problem using consensus clustering method, it is described in detail in section 2.3 along with approximation algorithms used for consensus clustering and measures used to find the quality of clustering.

Chapter 3

Genetic algorithm based feature subset selection

3.1 Introduction

Genetic Algorithm (GA) is a search-based optimization technique generally used to find optimal or near-optimal solutions to hard problems that may take exponential time. It is very much used in machine learning to handle optimization problems. Although genetic algorithms are adequately randomised by nature, they outperform random local search. They offer excellent parallel capabilities and are particularly beneficial when the search space is wide and there are many parameters to consider [39]. With this GA concept, a novel genetic algorithm based feature selection using consensus clustering is proposed (GACC). A dataset with some random features is represented as a chromosome. Dissimilarity measure is used as ”fitness function” and ”selection” is chosen to retain the best chromosomes in next population.

3.1.1 Motivation

The cardinality of the ”best” subset of features that ”describes” the data well is referred to as intrinsic dimensionality of a data collection. By looking at the best features that ‘clusters’ the data well, a subset of features that best characterises the data can be found. The data is then clustered differently by different clustering techniques. As a result, if

we achieve consensus across these several clusterings, the feature subset that discovers the consensus is referred to as a 'optimal subset' of features. With this motivation, a randomized approach is proposed to verify the feasibility of applying consensus clustering to feature selection problem. Further, it motivated us to develop a genetic algorithm based on consensus clustering algorithm for feature selection (GACC), as GA is proved to be one of the best search-based optimization technique.

3.2 Related work

Feature subset selection is quite an old problem and method that deals with feature space is a subset-search method which is more popular. GARIPPER, WBFS are some of the GA based methods used in literature. GARIPPER generates strings corresponding to feature subset and uses classifier accuracy of RIPPER algorithm as fitness function [109]. Crossover and mutation operators are used to generate chromosomes in new population. The algorithm stops after a fixed number of generations or until the desired classifier accuracy is met and gets the best set of features. In paper [56] a wrapper-based feature selection(WBFS) method is proposed to select the feature subset. This uses genetic algorithm(GA) and K-nearest neighbor(KNN) to rank the importance of features. Other classical methods for feature selection problem used here for comparison are SBC, FRNN-FS and SBPCA.

SBC classifier is used [14] to improve the performance of naive bayesian classifier when features are irrelevant or redundant. SBC uses C4.5 first to generate decision tree and selects attributes which are at first three levels of decision tree as the most important features. Experiment is repeated it for 5 times and the union is performed on all attributes selected at every iteration. Naive bayesian classifier is used to test the accuracy. Zhao et al. [114] developed feature selection algorithm based on Fuzzy Rough Neural Networks(FRNN). FRNN is applied to feature selection by heuristic backward search strategy. It is a combination of neural networks and feature selection. The time complexity of this algorithm is O(NlogN) where 'N' is number of features. Acharya in [93] formulated a dimensionality reduction technique called Supervised Bergman PCA(SBPCA), and have shown that how this will directly optimizes the goal of prediction using reduced dimensions. It is also shown SBPCA outperforms PCA, provided classes are linearly separable.

3.2.1 Background

There are many optimization problems in computer science, which are NP-Hard. In this scenario, genetic algorithms have proved to be efficient in providing near-optimal solutions in reasonable time. So, GA is chosen here to find solution for feature selection problem. The basic terminology used in GA is as follows:

- **Population:** Subset of possible solutions to given problem.
- **Chromosome:** It is the representation of one solution for the given problem.
- **Genetic operators:** They are used to generate other set of chromosomes (offspring) from the existing chromosomes. They include crossover, mutation, selection, operators etc.
- **Fitness function:** Used to test the quality of the chromosome while selecting the best chromosome from the population.

Figure 3.1 shows the flow of steps in general genetic algorithm.

The proposed GACC algorithm uses the following terminology:

- **Chromosome creation**: It is a random feature subset('On' positions of the chromosome represents features used)
- **Fitness function:** As every chromosome is clustered using K-means in GACC, to select the best chromosome from the population, GACC uses dissimilarity measure as fitness function defined as follows:

$$d(C_1, C_2) = (b + c) \; or \; \binom{n}{2} - (a + d) \tag{3.2.1}$$

where a = number of pairs of objects clustered in C_1 and C_2

b = number of pairs of objects clustered inC_1 but not clustered in C_2

c = number of pairs of objects clustered in C_2 but not clustered in C_1

d = number of pairs of objects not clustered in C_1 and C_2

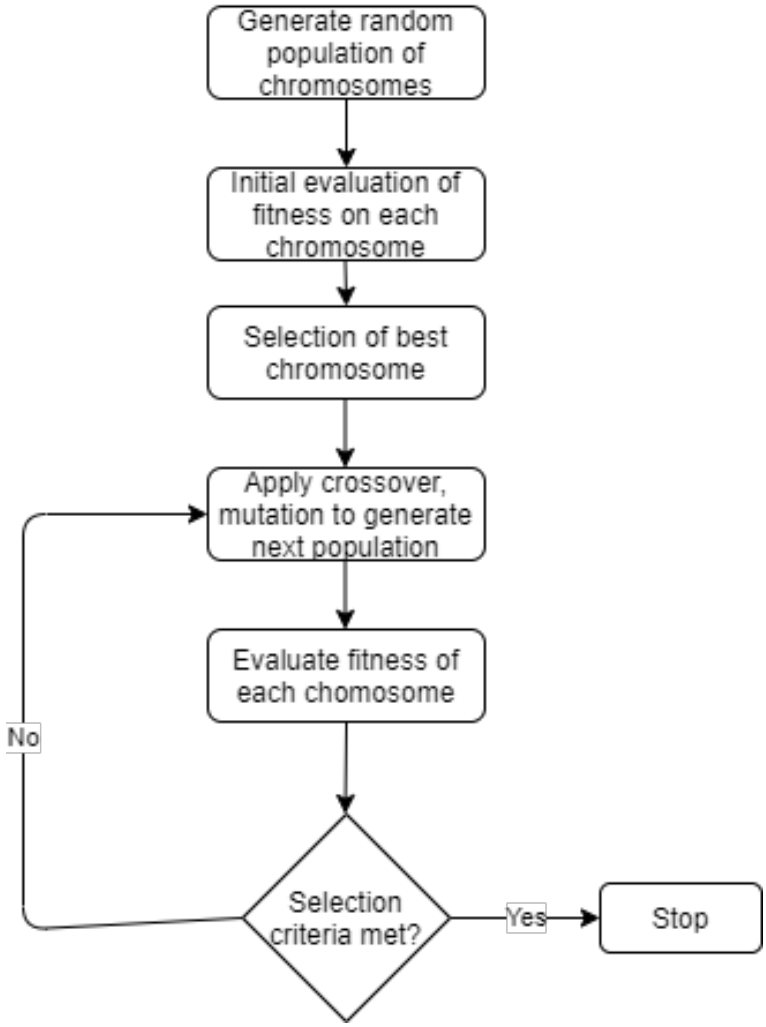

Figure 3.1: Flow of steps performed in genetic algorithm.

		F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	Feature Subset
Population	Chromosome 1	1	0	0	1	1	0	0	0	$\{F_1, F_4, F_5\}$
	Chromosome 2	0	0	1	0	1	0	1	1	$\{F_3, F_5, F_7, F_8\}$
	Chromosome 3	0	1	1	0	0	0	1	0	$\{F_2, F_3, F_7\}$
	Chromosome 4	0	0	1	0	0	1	0	0	$\{F_3, F_6\}$

Figure 3.2: Chromosome representation in GACC.

The total dissimilarity of a clustering C_i from all the other 'k-1' clusterings is calculated as:

$$\delta(C_i) = \sum_{j \neq i} dC_i, C_j$$

The best clustering is the one which has minimum δ value.

- **Genetic operators:** Crossover and mutation are used to generate the new offspring from existing chromosomes of previous population with 60% and 5% crossover and mutation probabilities.

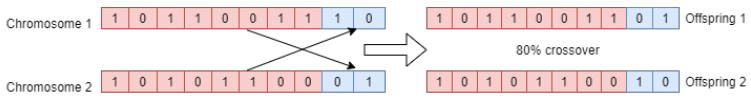

Figure 3.3: Crossover operation with 80% probability

- **Selection criteria:** GACC retains the top two best chromosomes in the next population.

3.2.1.1 Cluster validity indices

Cluster validity indices are used to verify the compactness of the clusters generated by the clustering algorithm. There are many such validity indices available in literature like Dunn Index, Davies-Bouldin(DB), Root-mean-square standard deviation(RMSSDT), Silhouatte etc[52]. One of the most cited indices is the Dunn index and it identifies clusters which are well seperated and compact[94]. Now, we validate the results with cluster validity index Dunn.

Cluster validity verification using Dunn Index (DI)

$$Dunn = \frac{d_{min}}{d_{max}} \tag{3.2.2}$$

Where d_{min} is the smallest distance between two objects from different clusters and d_{max} is the largest distance from same cluster.
Highest Dunn index indicates the better clustering quality.

3.3 Randomized approach

With the motivation of finding "best" feature subset that clusters the data in a well-separated manner, we tried to partition the dataset by retaining different combinations of feature subsets. To select best feature subset that clusters the data well, consensus clustering algorithm is used. To verify the application of consensus clustering method for feature subset selection problem, a feasibility study has been done using a randomized approach. To test this method, a synthetic dataset is constructed with 200 instances, 6 features and all points are well separated into four clusters. The six features are namely x, y, z, x^2, xy and z^2 where x and y are chosen in such a way that they are independent. Then, a correlation measure is used to find the correlation between each pair of features. Table 3.1 illustrates the cor-

relations between features and it can be seen that {x, y} are less correlated as they are independent and {x,x^2} are having high correlation, which means they are redundant.

Table 3.1: Correlation matrix of the synthetic data set

	x	y	z	x^2	xy	z^2
x	1.000	0.431	0 .736	**0.970**	**0.873**	0.713
y	0.431	1.000	0.559	0.512	0.713	0.556
z	0.736	0.559	1.000	0.768	**0.810**	**0.973**
x^2	**0.970**	0.512	0.768	1.000	**0 .934**	0.758
xy	0.873	0.713	**0.81**	**0.934**	1.000	0 .808
z^2	0.713	0.556	**0.973**	0.758	**0.808**	1.000

3.3.1 Algorithm

The dataset is generated using 6 dimensions (x, y, z, x^2, xy and z^2) The following steps have been followed in this approach.

- **Step 1:** Select a feature subset for size S = 2,3,,5 randomly and project the dataset along these feature dimensions.
- **Step 2:** K-means clustering is applied on each sample by varying 'K' value.
- **Step 3:** To find best 'K' value, best-of-k consensus clustering is applied.
- **Step 4:** Then, by considering all partitionings obtained by samples having two features (S = 2), best-of-k(Bok) is applied to choose one best partitioning or clustering.
- **Step 5:** Step 4 is repeated for feature subset of size 3,...M-1. This will generate one best feature subset per each size S = 2,3,...M-1.
- **Step 6:** Finally, Bok consensus clustering is applied on all partitionings generated after Step 4 to find the final best partitioning.
- **Step 7:** Final feature subset is formed with the features present in the best partitioning.

In the Table 3.2 'M' represents size of feature subset and 'K' represents number of clusters in K-means clustering algorithm. F(i, p) is a feature subset of size 'i', obtained using K-means with K = p.

Table 3.2: Description of randomized method.

K → M ↓	**2**	**3**	**4**	**.**	**p**	**.**
2						consensus row-wise with fixed 'M'
3						.
.						.
i					F(i,p)	.
.						.
.						.
.						Overall consensus(M,K)

3.3.2 Experiments and results

We have implemented randomized approach on a synthetic dataset having 200 points and 6 dimensions and on four benchmark datasets. The four datasets chosen are Wine that has less features and less instances; Pima, Breast cancer and WQW having more instances.

3.3.2.1 Synthetic dataset

This data set has 200 points and 6 dimensions. The input points are considered in such a way that they all are well separated into 4 groups. And dimensions are x, y, z, x^2, xy, z^2 where z =f(x). Table 3.1 illustrates the correlations between the dimensions. It can be seen that 'x' is correlated with 'x^2', it means they are redundant, where as 'x' and 'y' are less correlated, so they are independent.

The dataset is clustered using K-means algorithm for K = 2, 3, 4, 5, 6 and consensus among the clusterings is found using best-of-k algorithm (best clustering is the one which has least dissimilarity value). The results of the dissimilarity values obtained for each of the clusterings are given in Table 3.3. Each value in the columns of 2-6 of Table 3.3 is $\sum_{j=2}^{6} d(C_K, C_j)$. Last column represents best clustering with minimum dissimilarity value. After finding the best 'K' value for each feature subset, we can find consensus among 2-feature subsets, to find the best one.

In a similar fashion the experiment is carried out to find consensus among 3-feature, 4-feature and 5-feature subsets. In order to find the best feature subset that describes the original data, a consensus is used among the consensus of the 2, 3, 4 and 5 feature subsets. This is shown in Table 3.4, and it shows the best feature subset in a highlighted manner.

For subsets of size 4, {x,y,z, xy} as well as {x,y,xy,z^2} are obtained as the optimal feature subsets by consensus.

Table 3.3: 'K'= 4 is obtained as the best 'K' which gives least dissimilarity value by the consensus for all feature subsets of sizes 2, 3, 4 and 5.

Feature Subset	Dissimilarity values for K clusters					
	K=2	K=3	K=4	K=5	K=6	Consensus
x,y	30417	15417	**12770**	13366	15838	(K=4)
x,z	25138	14080	**13268**	15199	18347	(K=4)
y,z	23507	15284	**13153**	13432	15184	(K=4)
xy,z^2	22512	20859	**17925**	20558	22274	(K=4)
x,y,z	9133	9011	**6487**	6977	8688	(K=4)
x, x^2, xy	5640	**4121**	**4121**	5087	5626	(K=3) (K=4)
x, y, xy	9626	9156	**6774**	7349	8867	(K=4)
z, xy, z^2	9681	9999	**8859**	9075	10918	(K=4)
x,y,z,xy	8756	7598	**7552**	10257	9587	(K=4)
y,z,xy,z^2	10214	9875	**6788**	8977	10586	(K=4)
x,x^2,xy,z^2	9876	8674	**7684**	9876	8821	(K=4)
x,y,xy,x^2,z^2	5575	**5025**	**5025**	6547	6678	(K=3) (K=4)

Table 3.4: Overall consensus among feature subsets of sizes 2,3,4 and 5 with K = 4 high lighted in Table 3.3

x,y (K=4)	x,z (K=4)	y,z (K=4)	xy,z^2 (K=4)	FeatureSubset (Consensus)
12040	13700	15266	16126	**x,y**
x,y,z (K=4)	x,x^2,xy (K=4)	x,y,xy (K=4)	z,xy,z^2 (K=4)	
8319	12439	8361	15527	**x,y,z**
x,y,z,xy (K=4)	y,z,xy,z^2 (K=4)	x,x^2,xy,z^2 (K=4)	x,y,xy,z^2 (K=4)	
5554	10258	11678	**5554**	**x,y,z,xy**
x,y,z,x^2,xy (K=4)	x,z,x^2,xy,z^2 (K=4)	y,z,x^2,xy,z^2 (K=4)	x,y,xy,x^2,z^2 (K=4)	
4109	10321	5803	**4109**	**x,y,xy,x^2,z^2**

From Table 3.5, though 3 feature subsets show minimum dissimilarity value of 294, dunn index distinguishes {x,y} as the best feature subset with highest value of 0.0769. Hence, it is clear that x and y can be chosen as features to describe the data set adequately which confirms that independent features can be identified from the consensus clustering

method. Based on the solution obtained by the randomized approach, it is proved that consensus clustering approach can select an optimal feature subset.

Table 3.5: Consensus of the best 2,3,4 and 5-feature subsets.

Feature subset	x,y	x,y,z	x,y,z,xy	x,y,z,x^2,xy	Consensus
Dissimilarity value	**294**	294	294	494	x,y with K=4
Dunn Index	**0.0769**	0.0625	0.0085	0.0048	x,y with K=4

3.3.2.2 Benchmark datasets

Randomized method is also implemented on benchmark datasets namely Pima, Breast-cancer (BC), Wine quality white(WQW) and Wine given in Table 3.11. Features selected using random method based on consensus clustering are presented in Table 3.6, Table 3.7, Table 3.8 and Table 3.9. The overall results for Pima, BC, WQW and Wine compared to the literature are presented in Table 3.10. The results obtained are comparable with other methods in the literature. From these results, it is clear that the reduction in the number of features is almost 50%.

Table 3.6: Features obtained by random method for Pima Dataset.

Original features: 0-Number of times pregnant, 1-Plasma glucose concentration, 2-BP, 3-triceps skin fold thickness, 4-Serum insulin, 5- BMI, 6-Diabetes pedigree function, 7-Age.

S.No	Feature subset	Sum of Dissimilarities
1	{3,7}(K=3)	317036
2	{5,6,7}(K=3)	273894
3	{0,1,4,7}(K=3)	220568
4	**{0,1,4, 6,7}(K=3)**	**214378**
	Consensus	**{ 0,1,4,6,7} with (K=3)**

This method suffers from two problems:

1. It requires an exponential time complexity. As the number of dimensions increases, so the complexity increases.

2. It is inadequate to handle huge dimensions.

The randomized approach is treated as a proof of concept. Therefore, a search-based optimization technique using genetic algorithm with consensus clustering is proposed for

Table 3.7: Features selected by random method for Breast cancer dataset.

Original features: 0-Lump thickness, 1-Uniformity of cell size, 2-Marginal Adhesion, 3-Single epithelial cell size, 4-Bare Nuclei, 5- Bland Chromatin, 6-Normal Nuclei, 7-Mitoses, 8-Uniformity of cell shape, 9- Sample coder.

S.No	Feature subset	Sum of Dissimilarities
1	{1,3,4,6}(K=4)	16473
2	**{0,3,4,5,6}(K=4)**	**7945**
3	{0,2,4,5,6,7}(K=4)	8143
4	{0,1,2,3,4,5,6}(K=4)	9037
	Consensus	**{0,3,4,5,6} with (K=4)**

Table 3.8: Features selected by random method for Wine quality white dataset

Original features: 0-Fixed acidity, 1- volatilearidity, 2-citric acid, 3-Recidual sugar, 4-Chlorides, 5- Free sulfurdioxide, 6-Total sulfurdioxide, 7-Density, 8-PH, 9- Sulphates, 10-Alcohol.

S.No	Feature subset	Sum of Dissimilarities
1	{2,3,8,9}(K=6)	13633486
2	{0,4,5,7,9}(K=6)	14020282
3	{1,3,6,7,8,9}(K=7)	12284474
4	{0,2,3,6,7,9,10}(K=7)	12557888
5	{0,1,2,4,5,6,8,10}(K=5)	12912590
6	**{0,1,2,3,6,7,8,9,10}(K=6)**	**12153452**
	Consensus	**{0,1,2,3,6,7,8,9,10} with (K=6)**

feature selection.

3.4 GA based feature selection using consensus clustering (GACC)

To perform much better random local search, a novel genetic algorithm based feature selection using consensus clustering is proposed (GACC).

GACC method is described as follows:

- In the initial population, Chromosomes (feature subsets) are produced at random.
- The K-means algorithm is used to cluster each data set projected along the feature dimensions ('on' positions of chromosome) specified by the feature subset.
- Then, to find consensus across all feature subsets, dissimilarity value is used.

Table 3.9: Features obtained by random method for Wine dataset

Original features: 0-Alcohol,1- MalicAcid,2- Ash, 3-Alcanity of Ash, 4-Magnisium, 5-total Phenols, 6-flavonoids, 7-Non-Flavonoid phenols, 8-Proanthocyanins, 9-Color intensity, 10-Hue, 11-Diluted of wines, 12-proline.

S.No	Feature subset	Sum of Dissimilarities
1	{5,7,8,10,11}(K=3)	3676
2	{0,1,5,6,7,9}(K=3)	5974
3	**{0,4,5,8,9,10,11}K=3)**	**3298**
4	1,2,4,5,6,9,10,11(K=3)	4108
	consensus	**{0,4,5,8,9,10,11} with (K=3)**

Table 3.10: Comparison of randomized approach with methods in the literature.

Dataset	#original features	#reduced features	best 'K' value	Accuracy% (J48)	Literature Accuracy% (J48)
Pima	8	4	3	72.1	79[14]
BC	10	5	4	94.7	97[14]
WQW	11	9	6	59.62	58
Wine	13	7	3	92.1	94.7[114]

- In the next generation, by using 'elite' selection method, the two best chromosomes are chosen based on dissimilarity values, and are retained in the population. The remaining chromosomes are added to the population by generating them using crossover and mutation operations.

- The algorithm is terminated once the same best chromosome is obtained in two consecutive generations.

This is described clearly in Algorithm 3.1.

3.4.1 Time complexity

Initially, to perform K-means clustering for each chromosome with 'K' number of clusters with 'N' instances it takes O(KNI) time, where 'I' is the number of iterations. To find the best chromosome, best-of-k(BoK) algorithm takes O(k^2N) time. Therefore, overall time complexity of GACC algorithm is O($(k^2+KI)N$) where 'k' is the number of input clusterings given to besk-of-k algorithm and 'K' is the number of clusters used for K-means algorithm.

Algorithm 3.1 Genetic algorithm based feature selection using consensus clustering (GACC)

Input: Dataset of size N and F: Set of all features where $|F|$= D.
Output: Feature subset selected.

1: chromosome := (bit vector, K), where bit vector is of size D and represents a feature subset of F, where each bit is randomly chosen to be 0 or 1. 'K' represents number of clusters in K-means and chosen randomly between 2 to n.
2: Generate 'P' a set containing chromosomes in first generation
3: **repeat**
4: **for** Each chromosome in P **do**
5: Apply K-Means clustering algorithm, with 'K' chosen in the chromosome.
6: **end for**
7: Compute dissimilarity value for each clustering generated.
8: Pass the clusterings generated by each chromosome in P as input to best-of-k(BOK) and order them in increasing order of their fitness values. /*Dissimilarity values are used as fitness values*/
9: Select top two chromosomes C_1, C_2 from this generation and use them in the next generation. /*elite selection*/
10: To generate remaining (P-2) chromosomes in the next generation, apply crossover and mutation operations on the chromosomes to obtain P_1.
11: P $\leftarrow$ P_1 U $\{C_1, C_2\}$
12: **until** no change in the best chromosome
13: The best feature subset is derived from the 'on' positions of the best chromosome.

3.4.2 Experiments and results

3.4.2.1 Datasets used for GACC

Experiments have been conducted on UCI machine learning data sets such as Wine, Wine quality-white(WQW), breast cancer(BC) and Pima. Wine, Wine quality-white, Lung-cancer and breast cancer data sets are challenging datasets for classification. In order to test GACC algorithm, we selected such data sets. Table 3.11 shows the number of features along with other relevant details of these data sets.

We re-organized the datasets into four groups (i) less instances with less number of features, (ii) more instances with less number of features, (iii) less instances with more number of features and (iv) more instances with more number of features which are given in Table 3.12.

Table 3.11: Description of datasets

Dataset	# Instances	# Dimensions	#Classes
Seed	210	7	3
Pima	768	8	2
Breast Cancer	699	10	2
WQW	4878	11	11
Wine	178	13	3
Zoo	101	16	7
Ionospere	351	34	2
Lungcancer	32	56	2
Isolet	7797	617	26

Table 3.12: Benchmark data sets chosen from UCI with variation in number of features and number of instances

	Less #Instances	More #Instances
Less #features	Seed(7, 210) Wine(8, 178) Zoo(16, 101)	Pima (8, 768) Breastcancer(10, 699) WQW(11, 4878)
More # features	Ionosphere(34, 351) Lungcancer(56, 32)	Isolet(617,7797)

Table 3.13: Comparison of results obtained using GACC with FRNN-FS[114], GAPIPPER[109], SBC[14] and SBPCA[93](# original features mentioned with the dataset in the first column of the Table).

Dataset	GACC		Others		
	Reduced Features	Accuracy %(J48)	Other Methods	Reduced Features	Accuracy %(J48)
Seed(7)	4	92.20	SBC	5	79.12
Wine(13)	7	**95.03**	FRNN-FS TCbGA	6 9	94.77 99.6
Zoo(16)	8	96.18	GARIPPER TCbGA	7 5	96.00 98.03
Pima(8)	4	77.8	SBC	5	79.00
BC(10)	5	96.7	SBC FRNN-FS	4 4	97.00 93.8
WQW(11)	8	62.68	- -	–	–
Ionosphere(34)	10	94.40	GARIPPER TCbGA	10 14	94.61 98.32
Lungcancer(56)	5	79.58	FCBF TCbGA	5 9	80.83 96.30
Isolet(617)	127	72.45	FCBF	137	80.70

The results obtained using GACC are compared with the methods that are available

in the literature and are tabulated in Table 3.13. The choice of the methods is based on the availability of the results for the specific datasets considered for our experimentation. TCbGA is one of the latest methods that gives better accuracy in most of the cases but with more features. For datasets of group (i) with less features and less instances, GACC is performing very well and specially for the seed dataset and gives higher accuracy of 92.20% comapared to the literature. In both the cases of groups (ii) and (iii), the results are on par with those reported in the literature both in terms of accuracy and reduction in the number of features. Finally for group (iv) having dataset Isolet, it can be seen that GACC is not performing very well.

3.4.3 Discussion

There are many methods in the literature for dimensionality reduction. GACC method is compared with other methods from the literature which have used the same data sets. Bayesian network method is converting high-dimensional data to 2 or 3 dimensions. They have used unsupervised Bayesian distance metric to find the pairwise distance between two vectors. Using unsupervised distance metric requires the prediction of distribution, which is a difficult problem. Selective Bayesian Classifier algorithm(SBC) always selects a set of attributes that appear only in the first three levels of simplified decision tree which was constructed using C4.5. Selecting only from 3 levels may not work always.

The results achieved using GACC method is comparable with methods in the literature. In GACC method, a genetic algorithm is used to generate feature subsets. With the experiment of GACC method on Wine dataset in which 95.03% classifier accuracy is achieved with 7 features(total features are 13). Both the methods are able to pick up feature subsets which are giving comparable performance reported in the literature. For wine-quality(white), breast cancer and Pima, GACC performance is better than random method. From the above experiments it can be concluded that GACC selects the feature subsets better than random method.

From Table 3.14 it can be observed that most of the features (4 out of 7) are commonly selected in both of the methods. The results obtained using randomized method and GACC are plotted as in Figure 3.4.

Table 3.14: Features selected using the GACC method and random method for Wine dataset

GACC method:(7 features)	Random method:(7 features)
alchohol **magnisium** ash **proanthocyanis** **color intensity** malic acid Flavonoids	**alchohol** **magnisium** total phenols **proanthocyanis** **color intensity** 0D280/0D315 of diluted wines Hue
95.03% accuracy	92.1% accuracy

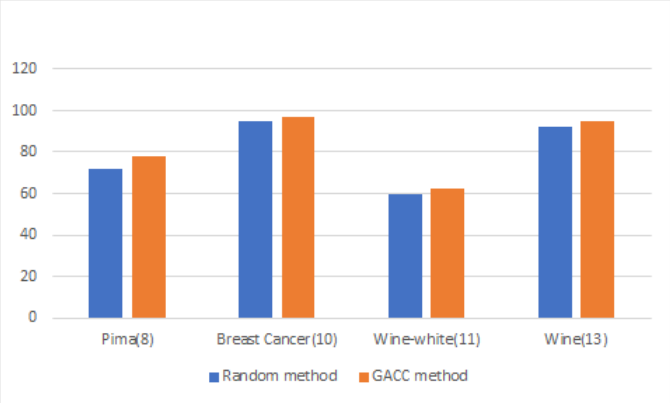

Figure 3.4: Comparison of Random method and GACC

3.5 Conclusions

Clustering high dimensional data is a challenging problem in data mining. By reducing the number of dimensions, we can group the data set into well separated clusters. The most popular method used for dimensionality reduction is PCA which does not select the feature subset from whole set of features, instead it constructs new features. This method is not feasible for very high-dimensional data. With the aim of better visualization of data using reduced feature subset a novel genetic algorithm based feature selection using consensus clustering algorithm is proposed (GACC). To prove the feasibility of consensus clustering for feature selection, initially a random method is proposed. Due to his high computational complexity, GACC is proposed. Random approach is tested on synthetic dataset of size 200 and results obtained showed that consensus clustering works well for feature selection problem. Random approach is further implemented on benchmark datasets from UCI and achieved comparable results with literature. Further GACC is tested on benchmark

datasets from UCI. GACC is proved to be better than random approach. Dissimilarity measure is used as a fitness function in GACC which need to be minimized. The stopping criteria used in GACC is also proved as good one. But, the time taken by this algorithm is high compared to the methods in the literature and may not be scalable for very high dimensional datasets.

Chapter 4

Community discovery based feature selection using consensus clustering

Graph-based methods have been studied in the field of cluster analysis in recent years. In general, when using a graph-based method, a graph is constructed on instances as nodes and defining an 'interaction' among the instances as edges. For clustering instances, edges are removed from the graph based on some specific criterion. The end result is a forest, with each tree in the forest representing a cluster. On the basis of this concept, Song et al. developed an algorithm known as FAST [98]. Instead of using data instances as points, FAST creates a minimum spanning tree using well known Prim's algorithm [83] based on the feature space using symmetric uncertainty value between the features as edge weight. Using T-relevance and F-correlation definitions some edges are removed from the tree. The final output is the forest of trees(clusters). From each cluster, the most representative feature that is highly correlated to the target class is chosen to form the feature subset. Since it operates in feature space, this algorithm is extremely fast. On the basis of this concept, a novel and efficient consensus clustering-based algorithm for feature subset selection is proposed here.

4.1 Motivation

With the inspiration from the FAST algorithm a new algorithm based on consensus clustering is proposed that works on feature space. The aim of the approach is that time can be saved by working on feature space rather than original data space, as the number of features is generally very small in comparison to the number of data instances. Feature subset selection can be achieved by selecting the most "representative" features that are highly correlated to the target class. Since different community discovery algorithms give different partitions of the feature space, consensus among these clusterings may find the 'best' subset of features that "describes" the data the well. Further, the subsets can be pruned further with the correlation between features that can be used to identify redundant features, and the correlation between a feature and a class variable (class correlation) that can be used to identify irrelevant features. Using these ideas, a method is proposed by constructing a complete graph on feature space using (i) Pearson's correlation values and (ii) Symmetric uncertainty as edge weights, and then partitioning the graph using algorithms from social network literature [31]. These are ultimately reconciled through consensus clustering to generate a suitable feature selection.

4.2 Background

Many graph partitioning algorithms have been proposed in the literature of social networks which are also referred to as community discovery algorithms. The problem of detecting communities in a social network is referred to as community detection. Essentially, the goal is to partition the graph so that dense edges are there within each group or community and sparse edges between groups. We present here a few of these algorithms which have been used in the proposed method for feature subset selection.

4.2.1 Community discovery algorithms

Extensive literature is available on community discovery algorithms in social network analysis [28]. Community discovery problem is an NP-hard problem. Hence a lot greedy approaches have been proposed in the literature. A few community discovery algorithms that are available in the 'igraph' package of Rstudio[1] have been considered for this work

which are explained below. For the purpose of evaluating the performance of these algorithms, Newman and Girvan[74] have proposed a quantitative measure known as ”modularity(Q)”.

4.2.1.1 Edge betweenness:

The main idea of the algorithm is based on the Edge betweenness centrality measure [74] that is defined by the number of shortest paths that are passing through the edge. Many other centrality measures are given in the book by Freeman [29]. The time complexity of this algorithm is O(mN) with ‘m’ edges and ‘N’ vertices.

4.2.1.2 Fast greedy:

This method tries to find the dense subgraphs, also called communities, by greedily optimizing the modularity score. This algorithm has linear time complexity with O($MdlogN$) with ‘N’ vertices, ‘M’ edges and ‘d’ depth of dendrogram [2].

4.2.1.3 Leading eigen vector:

This algorithm tries to find communities in a graph by calculating the leading positive eigenvector of the modularity matrix of the graph. The time complexity is O(N^2) with ‘N’ vertices [75].

4.2.1.4 Label propagation:

It works by labelling the vertices with unique labels, which are then updated by majority voting in the neighbourhood of vertex. This is a fast approach for determining community structure in networks that runs in almost linear time [85].

4.2.1.5 Spinglass:

Spin-glass model and simulated annealing are used to find communities in graphs. The network’s community structure is represented by the spin configuration that minimises energy of the spin glass, with the spin states serving as community indices [88]. This algorithm takes O(CN^2) time to form C number of communities with N vertices.

4.2.1.6 Multilevel:

The multi-level modularity optimization approach is used to determine community structure in this method. It is based on a hierarchical method and the modularity metric [103].

4.2.1.7 Optimal:

This function estimates the ideal community structure of a graph, by maximising the modularity measure over all possible partitions [12].

4.2.1.8 Walktrap:

This method finds the dense subgraphs, also called communities in a graph using random walks. A measure of similarity between vertices based on random walks is used and the idea that short random walks tend to stay in the same community [81] is used to discover communities. The time complexity of this method is O(N^2) with 'N' vertices.

4.2.1.9 Infomap:

This is an information theoretic technique that shows community structure in weighted and directed networks. The network is decomposed into modules by compressing a description of the probability of flow of random walks on a network [89].

4.2.2 Definitions and Heuristics

To find relevant features and redundant features, the following definitions and heuristics are used from the literature[110].

Definition 1: If a feature is not correlated with class or is weakly correlated with class, it is said to be **irrelevant**.

Definition 2: When a feature has a high degree of correlation with one or more other features, it is referred to as being **redundant**.

Definition 3: When a feature has a high correlation with a class, it is referred to as being **representative**.

Heuristic 1: Redundant and irrelevant features need to be eliminated.

Heuristic 2: Feature subset is formed by a set of representative features.

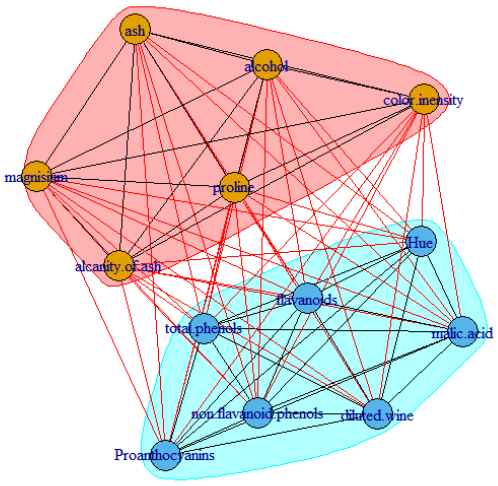

Figure 4.1: Example for community discovery in social networks for Wine dataset. Two communities are discovered here using fastgreedy algorithm.

As per the above definitions, the correlation between features (feature correlation) can be used to identify redundant features, and the correlation between a feature and a class variable (class correlation) can be used to identify irrelevant features. There are a variety of measures available in the literature for determining the relationship between two features or between a feature and a target class. Symmetric uncertainty(SU) is a non-linear correlation measure that has been used in FCBF and FAST. Here, both Pearson's correlation coefficient abbreviated as r and Symmetric uncertainty(SU) have been used in the proposed algorithm to measure the correlations.

4.2.2.1 Pearson's correlation coefficient:

Pearson's correlation between two variables x and y is defined as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4.2.1}$$

Here, x and y are two random variables $\bar{x}$, $\bar{y}$ denotes mean values of x and y respectively. This produces a value ranging from -1 to +1, inclusive. The value of -1 indicates total negative correlation, the value of 0 indicates no correlation, and the value of +1 indicates total positive correlation. This measure works well if all of the features have numerical

values associated with them.

In the algorithm edge weights are computed based on the Pearson's correlation value(r) between the features. This method has two limitations.

1. r value can be negative, which is not allowed by community discovery algorithms.

2. all features in the dataset must contain numerical values.

To overcome the above limitations, a mutual information(MI) based measure called *Symmetric Uncertainty* (SU) [82] is used instead of Pearson's correlation coefficient r to compute the edge weights between the features (f_i, f_j). The definition of SU is as follows:

4.2.2.2 Symmetric Uncertainty(SU):

It is a correlation measure that is based on the information-theoretic concept of *uncertainty* or *entropy* of a random variable. The entropy of a random variable X can be expressed as follows:

$$H(X) = -\sum_i P(x_i) log_2(P(x_i)), \tag{4.2.2}$$

and after observing the values of another random variable Y, the entropy of X is:

$$H(X|Y) = -\sum_j P(y_j) \sum_i P(x_i|y_j) log_2(P(x_i|y_j)), \tag{4.2.3}$$

where P(x_i) is the prior probabilities of variable X, and P($x_i|y_j$) is the posterior probabilities of X given the values of Y. And *information gain* is defined as

$$InfoGain(X|Y) = H(X) - H(X|Y) \tag{4.2.4}$$

According to information gain(InfoGain), if $InfoGain(X|Y) > InfoGain(Z|Y)$, it means feature Y is more correlated to feature X. Information gain is symmetrical, and values must be normalised in order to assure compatibility. Now, *symmetric uncertainty* can be defined as

$$SU(X,Y) = 2\frac{InfoGain(X|Y)}{H(X)+H(Y)} \tag{4.2.5}$$

The range of SU is [0, 1], with the number 1 indicating that knowledge of one variable totally predicts the other variable, whereas the value 0 indicates that X and Y are independent of one another.

4.3 Framework

We propose an algorithm called Community based consensus clustering(CDCC) which adopts the following framework. The process begins with the construction of a complete graph with features serving as vertices. The edge weight is calculated using two measures (i) Pearson's correlation coefficient r and (ii) Symmeteric uncertainty(SU) between features (f_i, f_j). Then the graph is partitioned using a few popular community discovery algorithms that are available in *igraph* package of the Rstudio [1] software package, namely, Fast greedy, Edge betweenness, Label propagation, Leading eigen vector, Multilevel, Spinglass, Walktrap, Optimal, and Infomap community discovery algorithms. These algorithms divide the graph into groups or communities of nodes that are connected by edges in such a way that edges within the community are more dense and edges between the communities are sparse[18]. The performance of these algorithms is measured by maximizing "modularity(Q)" as proposed by Newman and Girvan [74]. Since different graph partitioning algorithms partition the graph in different ways, it is necessary to use a consensus clustering algorithm to select the best partitioning for the graph. Here, BoK(Best-of-k) consensus clustering is used to find the best partitioning, which is a 2-approximation algorithm [26]. Then choose a representative feature from each community of the best partitioning or clustering to form the feature subset and evaluate the accuracy of the classifier. The algorithm follows a wrapper approach by continuing to take the top representative features till the accuracy of the classifier does not decrease. Further, if a feature is correlated with the already selected features, then it does not get added to the feature subset.

4.4 Proposed algorithm

Community discovery based consensus clustering(CDCC) algorithm is given in Algorithm 4.2. Three variants using absolute PCC, square PCC and symmetric uncertainty are denoted as CDCC-$|r|$, CDCC-r^2 and CDCC-SU respectively.

Algorithm 4.2 Community discovery based consensus clustering(CDCC)

Input: Dataset is X=<**x**,**t**>. N: Dataset size and feature set is F, where F={f_1, f_2, f_3,.....f_M} and t is the target vector, CD={*Fast greedy, Edge betweenness, Label propagation, Leading eigen vector, Multilevel, Spinglass, Walktrap, Optimal, and Infomap* }.
Result: Feature subset FS.

$FS \leftarrow \emptyset$
Construction of a complete graph G=(V, E) where, $V \leftarrow F$
for all $i \leftarrow 1$ *to* M *in* F **do**
    for all $j \leftarrow 1$ *to* M *in* F **do**
        $W(e_{ij}) \leftarrow correlation(f_i, f_j)$
    end for
end for
Apply community discovery algorithms to partition graph G
for all $k \leftarrow 1$ *to* n *in* CD **do**
    $partition(k) = community\_discovery(G,k)$
end for
Apply BoK algorithm to find best partitioning from step 6
$best\_partition \leftarrow BoK(partition(1), \ldots, partition(n))$
for all $c \leftarrow 1$ *to* C *of* $best\_partition$ **do**
    $f_r(c) \leftarrow max_i\{correlation(f_i,t) : f_i \in c\}$
    $FS \leftarrow FS \cup f_r(c)$
    $c \leftarrow c - f_r(c)$
end for
for all $c \leftarrow 1$ *to* C *of* $best\_partition$ **do**
    while (accuracy is not decreasing) **do**
        $f_r(c) \leftarrow max_i\{correlation(f_i,t)$ AND NOT $((correlation(f_i,FS) : f_i \in c\}$
        $FS \leftarrow FS \cup f_r(c)$
        $c \leftarrow c - f_r(c)$
    end while
end for
return FS ▷ Final feature subset

4.4.1 Complexity Analysis

Initially, the time required to find the correlation between every pair of features is O(NM^2), where N is the number of instances and M is the number of features. The second part of the algorithm is the construction of a complete graph, which takes O(M^2) time to complete. It takes O(PM) time to partition a graph using community discovery algorithms, where

'P' represents the number of edges in the graph. In the end, the time required to apply a consensus clustering algorithm best-of-k(BOK) is O(Mk^2), where 'k' is the number of input clusterings . As a result, the overall complexity of implementing CDCC and CDCC-SU is O(NM^2), which is quadratic in number of features.

4.5 Experiments and Results

4.5.1 Datasets

The CDCC algorithm is implemented on benchmark datasets from UCI machine learning repository. Here, four types of datasets are considered. Wine and PIMA ($<$ 15 features), Zoo and Ionosphere($>$15 and $<$ 50 features), Lungcancer($>$ 50 and $<$ 100 features), Musk1 and Musk2($>$ 100 features). Table 4.1 gives the summary of datasets used in CDCC algorithm experiment. As CDCC-SU can handle categorical data, experiment is also carried out on other three categorical datasets namely Sonar, Spambase and Waveform datasets from UCI repository in addition to the datasets mentioned in Table 4.1.

Table 4.1: Benchmark datasets chosen from UCI

Dataset	# Features	# Instances	#Classes
Wine	13	178	3
PIMA	8	768	2
Zoo	16	101	7
Ionophere	34	352	2
Lungcancer	56	32	2
Musk1	168	476	2
Musk2	168	6598	2

Table 4.2: Description of catergorica datasets chosen from UCI to implement CDCC-SU in addition to datasets mentioned in Table 4.1.

Dataset	# Features	# Instances	#Classes
Sonar	60	500	2
Spambase	57	4601	2
Waveform	21	5000	3

4.5.2 Implementation

In CDCC algorithm, Pearson's correlation(r) value between a pair of features is used as edge weight. But, as the range of r is [-1, 1], edge weight also may become negative. Many of the community discovery algorithms do not allow negative edge weights. To make the edges positive, two ways are considered.

- using square of correlation value as edge weight.
- using absolute value of correlation as edge weight.

.

The results for the small dataset named *Wine* having 13 features are shown to illustrate the method. Number of communities generated by CDCC algorithm with square of correlation as edge weight is shown in Table 4.3, and absolute value of correlation as edge weight is shown in Table 4.4.

Table 4.3: Number of communities generated for Wine data using CDCC-r^2 with different community discovery algorithms.

Algorithm	**#Communities**
Fast Greedy	1
Edge Betweenness	1
Spinglass	3
Leading Eigen Vector	3
Multilevel	3
Optimal	3
Label Propagation	1
Walktrap	13
Consensus(BOK)	**3**

It can be seen in Table 4.3 that the number of communities generated by different community discovery algorithms for the Wine dataset ranges from 1 to 13 depending on the algorithm used. In addition, four out of eight algorithms provide three communities. The best-of-k(BoK) consensus clustering algorithm is used to find the optimal number of communities, and it results in three communities being selected. In Table 4.4, majority of the algorithms produce two communities, and the best-of-k algorithm produces two communities as well. A 'representative' feature is selected from each community that has

Table 4.4: Number of communities generated for Wine data using CDCC-$|r|$ with different community discovery algorithms.

Algorithm	**#Communities**
Fast Greedy	2
Edge Betweenness	3
Spinglass	2
Leading Eigen Vector	1
Multilevel	2
Optimal	2
Label Propagation	1
Walktrap	2
Consensus(BOK)	**2**

the highest correlation with the target class, and these features are used to create the final feature subset.

Let *cor(f,c)* denote correlation between feature and target class. Highlighted features in Table 4.5 and Table 4.6 show 'representative' features with highest *cor(f,c)* from each community.

Table 4.5: Representative features are highlighted from each community for Wine data set using square of correlation as edge weight.

Community 1		**Community 2**		**Community 3**	
Feature	corr(f,c)	Feature	corr(f,c)	Feature	corr(f,c)
Malic acid	0.437	Alcohol	0.32	Total phenols	0.71
Hue	**0.61**	Ash	0.049	**Flavanoids**	**0.84**
		Alcanity of ash	0.51	Non flavanoid phenols	0.48
		Magnisium	0.209	Proanthocyanins	0.499
		Color intensity	0.26	0D280/315 diluted wine	0.265
		Proline	**0.63**		

The accuracy of the classifier was first evaluated using a representative feature from each cluster. In Wine data set initially 2 representative features (one from each cluster) were selected as shown in Table 4.5, and accuracy with these features is 93.2%.

The next top representative feature from each community, that has a lower correlation with the previously chosen feature was added to the feature subset. The accuracy of the classifier slightly improved as a result of this. Whenever there is a reduction in the accuracy of the classifier, stop adding features to the feature set. By adding one more feature, accuracy improved to 94.9%, then for 6 features it is 96.16%. After that, it was observed

Table 4.6: Representative features from each community for Wine data set using absolute value of correlation as edge weight

Community 1		**Community 2**	
Feature	corr(f,c)	Feature	corr(f,c)
Malic acid	0.437	Alcohol	0.32
Hue	0.61	Ash	0.049
Total phenols	0.71	Alcanity of ash	0.51
Flavanoids	**0.84**	Magnisium	0.209
Non flavanoid phenols	0.48	Color intensity	0.26
Proanthocyanins	0.499	**Proline**	**0.63**
0D280/315 diluted wine	0.265		

that there was no change in accuracy and there after accuracy decreased to 93.01%. Same procedure is used to find the feature subsets from the communities given in Table 4.6 also. It is observed that 90% of the features are commonly selected using both the correlations.

The experiments are carried out to find representative features for all data sets given in Table 5.1 using CDCC. The largest data set considered is Musk2 having 167 features and 6598 number of instances. Table 4.7 shows the results obtained using square of correlation as edge weight. Table 4.8 shows the results obtained using absolute value of correlation as edge weight. From Table 4.7 and Table 4.8 it can be seen that, the number of features selected using our method is less compared to the number of features selected by other methods in literature.

Table 4.7: Results for CDCC-r^2.

Dataset	**Our method**		**Literature**	
	#selected features	**accuracy%** (J48)	**#selected features**	**accuracy%**
Pima(8)	2	73.9	5	79(SBC)
Wine(13)	6	96.16	7	97.4(GARIPPER)
Zoo(16)	5	92.07	7	96(GARIPPER)
Ionosphere(34)	5	92.3	10	94.6(GARIPPER)
Lung cancer(56)	5	60	5	87(ReliefF)
Musk1(168)	6	76.8	25	74(WBFS)
Musk2(168)	4	93.08	2 2 10 25	91.33(FCBF) 94.6(ReliefF) 95.5(CFS-SF) 96.35(FCBF-P)

Table 4.8: Results for CDCC-$|r|$.

Dataset	Our method		Literature	
	#selected features	accuracy% (J48)	#selected features	accuracy%
Pima(8)	4	74.08	5	79(SBC)
Wine(13)	4	94.9	7	97.4(GARIPPER)
Zoo(16)	4	91.08	7	96(GARIPPER)
Ionosphere(34)	5	92.3	10	94.6(GARIPPER)
Lung cancer(56)	5	65.6	5	87(ReliefF)
Musk1(168)	4	75.4	25	74(WBFS)
Musk2(168)	2	91.88	2	91.33(FCBF)
			2	94.6(ReliefF)
			10	95.5(CFS-SF)
			25	96.35(FCBF-P)

4.5.2.1 CDCC with symmetric uncertainty as edge weight (CDCC-SU):

Here, symmetric uncertainty(SU) value is used to compute the edge weight in place of Pearson's correlation. Features having categorical values cannot be handled by Pearson's correlation and hence Symmetric uncertainty(SU) can be considered. SU is calculated between each feature and class and also between every pair of features. Using features as nodes and SU-value between features as edge weights a complete graph is constructed, then graph is partitioned using community discovery algorithms. BOK is applied to find the best partitioning.

Table 4.9: Performance of CDCC-SU compared to latest literature TCbGA.

Dataset	CDCC-SU		TCbGA[61](2017)	
	Features	Accuracy	Features	Accuracy
Wine(13)	6	96.08	9	99.60
Zoo(16)	**5**	93.54	**5**	98.03
Ionosphere(34)	**6**	92.30	14	98.32
Lungcancer(56)	**5**	78.10	9	96.30
Musk1(166)	**4**	75.40	97	94.27
Musk2(166)	**4**	93.08	86	99.23
Spambase(57)	**7**	84.72	19	91.85
Sonar(60)	**3**	75.40	9	84.62
Waveform(40)	**13**	83.86	18	85.43

Table 4.9 shows that proposed method CDCC-SU is selecting very less number of features compared to literature. Number of features selected using CDCC-SU method is

Table 4.10: Number of features selected by CDCC-SU compared to latest literature.

Dataset	CDCC-SU	GCACO [71] (2015)	GCNC [70] (2015)	UFSACO [92] (2014)	RRFS [24] (2012)
Wine(13)	6	6	7	**5**	**5**
Ionosphere(34)	**6**	15	17	20	20
Spambase(57)	**7**	24	27	30	30
Sonar(60)	**3**	24	25	30	30

Table 4.11: Classifier accuracy of CDCC-SU compared to latest literature.

Dataset	CDCC-SU	GCACO [71]	GCNC [70]	UFSACO [92]	RRFS [24]
Wine(13)	96.08	95.73	95.08	93.76	94.42
Ionosphere(34)	92.30	90.24	89.91	86.80	89.40
Spambase(57)	84.72	88.22	88.11	86.48	82.71
Sonar(60)	75.40	77.60	74.36	75.34	72.53

Table 4.12: Comparison of CDCC-SU with classical methods SBC, GARIPPER, ReliefF, FCBF and NMIFS (Datasets represented with blue colour indicates categorical datasets.

Dataset	CDCC-SU		Literature	
	#Features	Accuracy%	#Features	Accuracy%
Pima(8)	**4**	75.6	5	79(SBC[14])
Wine(13)	**6**	96.08	7	97.40(GARIPPER[109])
Zoo(16)	**5**	93.54	7	96(GARIPPER[109])
Ionosphere(34)	**6**	92.30	10	94.60(GARIPPER[109])
Lung cancer(56)	5	78.10	5	87.00(ReliefF[64])
Musk1(168)	**4**	**75.4**	25	74.00(WBFS[56])
Musk2(168)	4	**93.08**	2	91.33(FCBF[110])
			2	94.6(ReliefF[46])
			10	95.5(CFS-SF[35])
			25	96.35(FCBF-P[110])
Waveform(40)	13	**83.86**	13	81.52(NMIFS[23])
Spambase(57)	3	**84.72**	3	75.8(NMIFS[23])
Sonar(60)	**7**	75.4	11	86.36(NMIFS[23])

very less compared to TCbGA [61] method for Wine, Ionosphere, Lungcancer, Spambase, Sonar and Musk1, Musk2. Musk1, Mus2 are selecting even less than 10% features compared to TCbGA. Except for Wine dataset, other methods are selecting more than twice the number of features selected by CDCC-SU and classifier accuracies of TCbGA are slightly high compared to the proposed methods. This is possible, as CDCC-SU selects very less number of features compared to TCbGA. From Table 4.10 and Table 4.11 it can be

seen that, classifier accuracy of Wine and Ionosphere is more compared to GCACO [71], GCNC [70], UFSACO [92] and RRFS [24] in spite of selecting least number of features. Classifier accuracies are on par with the results from the literature for Spambase and Waveform datasets. Even though TCbGA is achieving slightly more accuracy than CDCC-SU, computational complexity of TCbGA is very high compared to the proposed method. Time taken by TCbGA to find optimal feature subset of Sonar dataset having moderate number of features(i.e,. 60) is in hours where as maximum time taken by CDCC-SU for any high-dimensional dataset is in minutes.

4.5.3 Robustness of CDCC algorithm

CDCC uses BoK consensus clustering algorithm to find the best partitioning among all input partitionings. Initially BOK is used due to its linear time complexity. Recently Huang et al. [41] and Banerjee et al. [9] have proposed new consensus clustering algorithms namely LWEA and WHAC respectively. CDCC-SU has been tested with LWEA and WHAC also to check the sensitivity of consensus clustering algorithm. It can be seen from the results presented in Table 4.13, Table 4.14 that BoK, LWEA and WHAC all are giving nearly the same results.

Table 4.13: Number of Features selected by CDCC-SU, with combination BOK, LWEA and WHAC consensus clustering algrotihms.

Dataset	CDCC-SU+BOK	CDCC-SU+LWEA	CDCC-SU+WHAC
Wine(13)	6	5	5
Zoo(16)	5	5	5
Ionosphere(34)	6	6	7
Lungcancer(56)	5	5	5
Musk1(166)	4	6	6
Musk2(166)	4	4	4
Waveform(40)	13	9	8
Spambase(57)	7	6	6
Sonar(60)	3	3	3

4.5.4 Discussion

There are many methods in the literature for feature subset selection. CDCC-r, CDCC-r^2 and CDCC-SU algorithms are compared with classical methods from the literature. Relief

Table 4.14: Classifier accuracy of CDCC-SU, with combination BOK, LWEA and WHAC consensus clustering algrotihms

Dataset	CDCC-SU+BOK	CDCC-SU+LWEA	CDCC-SU+WHAC
Wine(13)	96.08	97.19	97.19
Zoo(16)	93.54	93.54	93.54
Ionosphere(34)	92.30	92.30	93.54
Lungcancer(56)	78.10	78.10	78.10
Musk1(166)	75.40	80.46	80.46
Musk2(166)	93.08	95.30	95.30
Waveform(40)	83.8	84.62	82.56
Spambase(57)	84.72	84.69	84.69
Sonar(60)	75.40	75.40	75.40

and ReliefF are faster to implement, but they do not identify redundant features. FCBF and FAST both can identify irrelevant and redundant features. But, both are sensitive to initial parameter θ that is the threshold of feature relevance. Classification results vary with θ value. Selective Bayesian Classifier algorithm(SBC) always selects a set of attributes that appear only in the first three levels of simplified decision tree which was constructed using C4.5. Selecting only from 3 levels may not work always.

In Table 4.7 and Table 4.8, the number of features selected using CDCC method corresponds to the most representative features from each of the communities given by best-of-k consensus clustering algorithm. The algorithm follows a wrapper approach by continuing to take the top representative features till the accuracy of the classifier does not decrease. Further, if a feature is correlated with the already selected features, then it does not get added to the feature subset. The number of features selected by CDCC-r and CDCC-r^2 are different as can be seen in Table 4.7 and Table 4.8. The number of features selected is more when edge weight is considered as square of correlation value instead of absolute of correlation value in 4 out of 7 data sets. But, 80% of the features are same in both the methods. Both the variants of CDCC-PCC could not achieve comparable accuracy of 87% for lung cancer dataset as reported in the literature. We investigated further by running ReliefF available in Weka to obtain the top features and tested the accuracy with J48 classifier. But we found the accuracy to be only 75% not matching with the 87% obtained by ReliefF as reported in the literature.

CDCC-r is performing better than CDCC-r^2. CDCC-SU shows better accuracy with similar feature reduction. Further, CDCC-SU is compared with classical methods pre-

sented in Table 5.6 and it is clear that performance of CDCC-SU is high on most of the UCI repository datasets. Except for Lungcancer, in most of the cases the number of selected features is less compared to literature and classifier accuracies are on par with the existing methods. Computational complexity of the proposed method is less compared to the methods in the literature. CDCC-SU achieves good results for Musk 2, Spambase and Waveform datasets compared to the current literature. For most of the datasets, results are on par or less but with less number of features. Reason for low accuracy for Musk1 and Lungcancer datasets could be that the instance to feature ratio is quite low for both the datasets which implies that the correlations between the features may not be adequately captured.

Further, the sensitivity of the CDCC algorithm is verified by replacing BoK consensus clustering algorithm with two recent consensus clustering algorithms namely LWEA and WHAC. The results obtained establish that the CDCC approach is not sensitive to the choice of consensus clustering algorithm.

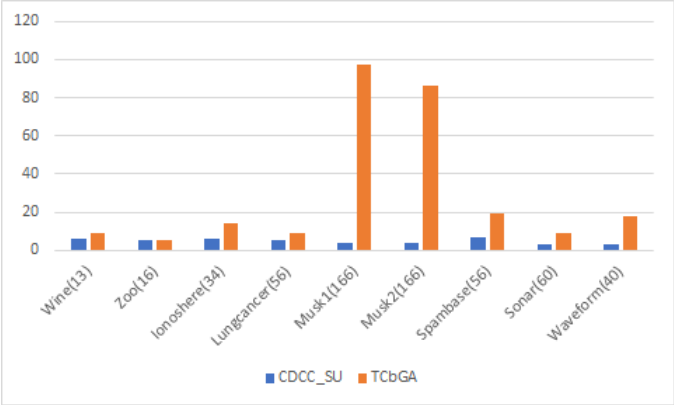

Figure 4.2: Number of features selected by CDCC-SU compared to latest literature.

4.6 Conclusions

In this chapter, an algorithm based on consensus clustering(CDCC) for feature subset selection with different correlation measures is proposed. This method is tested on benchmark data sets from UCI. A few selected community discovery algorithms are used from *R*studio to partition the graph constructed on feature space. To find the best partitioning, best-of-k(BOK) consensus clustering algorithm is used. Representative features that are highly correlated to the target class are selected from each community to form the fea-

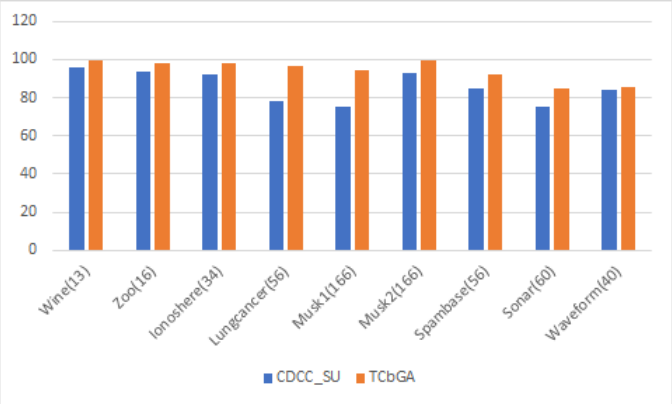

Figure 4.3: Classifier accuracy obtained by CDCC-SU compared to latest literature.

ture subset. Number of features selected using CDCC algorithm is less than the number of features selected by other methods in the literature. Accuracies obtained are on par with the accuracies obtained in the literature for most of the datasets. Time complexity of CDCC algorithm depends on the number of features, not on number of instances. Since, Pearson's correlation coefficient is used to find the correlation values, this algorithm works only for numerical data sets. In order to deal with all types of data sets further symmetric uncertainty(SU) value is used as edge weight.

Contributions in this chapter are summarized as follows: A novel method CDCC for feature subset selection using consensus clustering is proposed. constructing the communities based on the features where by features in a community are taken to be more closely correlated. The highlight of this method is using consensus clustering to retrieve highly representative features that are in common with the partitionings obtained by various community discovery algorithms.

Chapter 5

Feature subset selection for high-dimensional data and big data

In connection with the rapid development of information technology and internet, the scale of data that needs to be processed has been continuously increasing, resulting in the emergence of problems such as ”curse of dimensionality”. Feature selection is an essential step for high-dimensional datasets to achieve a good classification accuracy. Feature selection is used to reduce overall number of dimensions while simultaneously improving classification accuracy and efficiency. This can be accomplished by detecting and removing irrelevant and redundant features.

Feature selection is generally performed in the search space composed of all possible combinations of data features, using the feature subset search algorithm to identify a subset of features that are highly correlated with class variable. The dimensionality of the datasets in real-time applications like gene expression datasets has increased tremendously in last few years. Finding the best feature subset that describes the original high-dimensional dataset is very difficult and time consuming.

Feature ranking is a simple technique used for feature subset selection. This method is preferable over other methods due to less computational complexity, since most of the feature ranking algorithms use greedy approach. There are a number of different feature

ranking algorithms that are already available and every algorithm generates a ranking of features, starting with the most significant feature and progresses to the least important feature in the list. In order to rank the features, each algorithm employs a unique set of ranking criteria such as feature weight, information theoretic measures, statistical measures, and so on. As a result, the ranking of features by different algorithms may differ as well. When features are selected using a variety of ranking algorithms, the issue of stability or robustness becomes the most critical consideration.

Even though the algorithms presented in Chapter chapter 3 and chapter 4 are robust, they are not scalable for high-dimensional data. To address this problem, a new robust scalable feature selection algorithm (FRCC) is presented here which is based on the idea of consensus clustering.

5.1 Motivation

In large dimensional datasets, many features may be irrelevant and redundant. Such features can also affect the classifier performance. There exist many feature ranking algorithms that can be used to pick the top most relevant features. Therefore consensus clustering is applied to obtain the most relevant features and removing redundant features. A novel approach called FRCC is proposed to implement this idea.

5.2 Background

5.2.1 Feature ranking algorithms

First, a few standard filter-based feature ranking methods from the literature [57], [36], [42], [64], [60], [50], [87], [82] mentioned below are used to generate feature rankings along with feature weights.

1. **Chi-squared(χ^2):** This statistic is based on the χ^2-statistic, and it evaluates features in a way that is independent of how they are classified. The greater the value of the Chi-square, the more relevant the feature is in relation to the class. First, the feature values must be distributed into a number of intervals using an entropy-based discretization method, before the rest of the process can begin.

2. **Information Gain(IG):** Information Gain (IG) is a metric that is commonly used in the fields of machine learning and information theory. In the context of class prediction, IG is defined as the number of bits of information gained about the class prediction by knowing the value of a given feature when predicting the class. Before calculating the information gain of a given feature X with respect to a given class attribute Y, it is necessary to understand both the uncertainty about the value of class attribute Y and the uncertainty about the value of class attribute Y when the value of X is known. In the former, the entropy of Y is measured by H(Y), whereas in the latter, it is measured by H(Y—X), which is the conditional entropy of Y when given X. Information Gain can be defined as:

$$IG(X) = H(Y) - H(Y|X) \tag{5.2.1}$$

3. **Gain Ratio(GR):** In comparison to Information Gain, the Gain Ratio (GR) is a refinement. While IG prefers features that have a large number of values, GR's approach is to maximise the information gain from a feature while keeping the number of its values to a bare minimum. The following is a description of the intrinsic value of $X = (X_1, X_2, \ldots X_M)$

$$IV(X) = -\sum_{i=1}^{M} (|X_i|/N) log(|X_i|/N) \tag{5.2.2}$$

where'M' is the number of distinct values in X, and N is the total number of instances. Then, Gain Ratio of attribute X is as follows:

$$GR(X) = IG(X)/IV(X) \tag{5.2.3}$$

4. **OneR:** OneR uses Holte's rule-based classification algorithm to rank the features. Essentially, the method finds a simple rule for each feature by identifying the majority class for each feature's value. Then, each rule's correctness is assessed, and the features are ranked according to the quality of the related rules.

5. **ReliefF:** Kononenko et al. suggest a few changes to Relief. To begin, they use the Manhattan (L1) norm instead of the Euclidean (L2) norm to locate the near-hit and near-miss, but the rationale is not given. While updating the weight vector, they found that the absolute differences between x_i and near-hit_i, as well as x_i and near-

$miss_i$, was sufficient (instead of square of those differences). Rather than repeating the method 'm' times, complete it for a small number of 'n' (up to one thousand). ReliefF also searches for 'k' nearby hits and misses and averages their contributions to the weights of each feature, rather than identifying the single nearest hit and single nearest miss, which may cause redundant and noisy features to affect the selection of the nearest neighbours.

6. **Symmetric Uncertainty(SU):** It is a correlation measure based on information-theoretic concept of entropy or uncertainty of a random variable.

5.3 Framework for feature ranking based feature subset selection

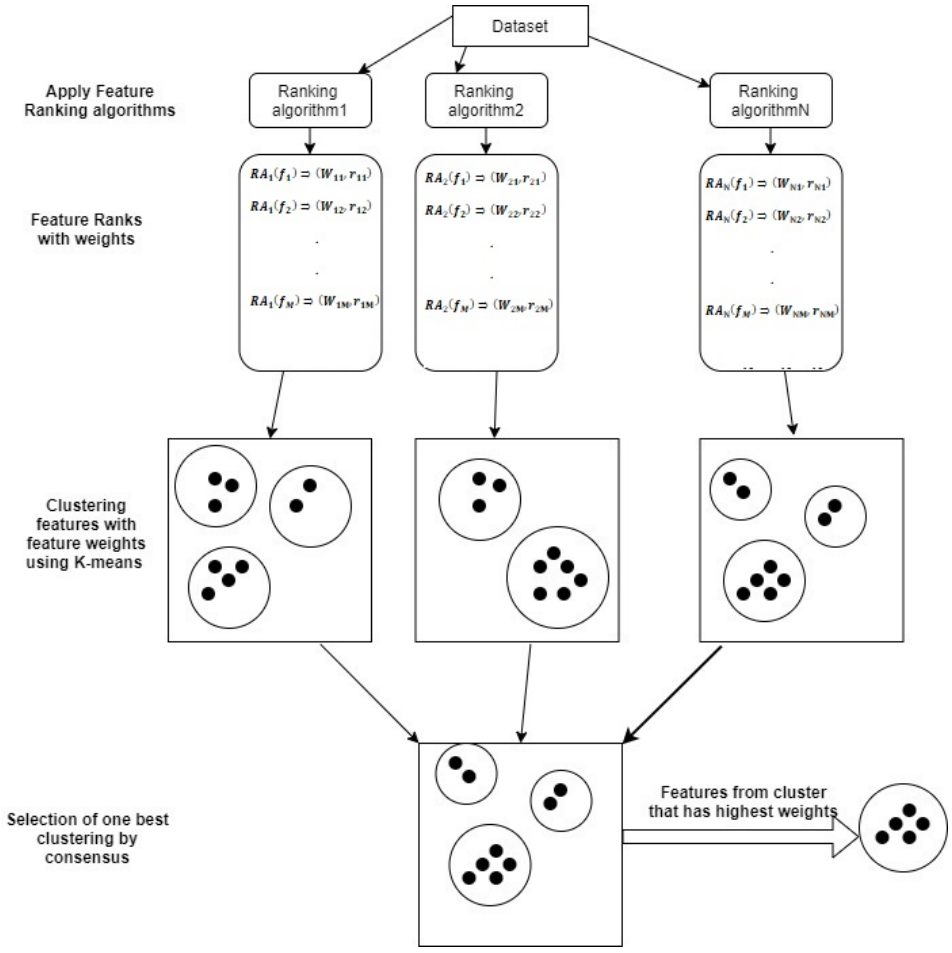

Figure 5.1: Feature ranking based on consensus clustering method.

K-means clustering is performed with the feature weight considered as the single dimension for each ranking algorithm *RA*. K-means clustering is applied to obtain the cluster labels for the features. An input file with *M* rows and *n* columns is created where M is the number of features and *n* is number of ranking algorithms. Each column contains the cluster labels assigned to the features by the K-means algorithm. This matrix is provided as input to the consensus clustering algorithm. For the purpose of finding the best clustering, the BOK consensus algorithm is used. Finally, features from the best cluster are chosen based on the fact that the total of the weights of the features is the greatest. Figure 5.1 reflects the FRCC approach.

5.4 Proposed algorithm

The FRCC algorithm is described in detail in Algorithm 5.3.

Algorithm 5.3 Feature ranking based consensus clustering (FRCC)

Input: Data set is X=<**x**,**t**>. Size of the dataset is N and feature set is F, where F=$\{f_1, f_2, f_3, \ldots, f_M\}$ and t is the target vector, W=$\{w_1, w_2, w_3, \ldots, w_M\}$, FRA={*Information gain, Gain ratio, ReleifF, OneR and Symmetric Uncertainty*}

Output: Feature subset FS

1: $FS \leftarrow \emptyset$;
2: **for all** $k \leftarrow 1$ *to n in FRA* **do** ▷ Apply feature ranking algorithms on dataset D
3: $(F,W)_k$ = Feature_Ranking(F,k);
4: **end for**
5: Apply K-Means clustering on each ranking output by considering feature weight as one dimension.
6: **for all** $i \leftarrow 1$ *to* $|FRA|$ **do**
7: $partition(i) = K\text{-}means(W_i)$;
8: **end for**
9: Apply BoK algorithm to find best partitioning.
10: $best\_partition \leftarrow BoK(partition(1), \ldots, partition(|FRA|))$;
11: Select features from the cluster that has highest weights
12: $C_B = best\_cluster(best\_partition)$;
13: $FS \leftarrow C_B$; ▷ Add each feature f from cluster C_B
14: **return** FS; ▷ Final feature subset

5.4.1 Time complexity of FRCC

The time required to compute using *n* feature rankings for *M* number of features is equal to O(nNM). The time required to partition *M* features using K-means clustering techniques is

O(KMI), with I representing the number of iterations. Best-of-k (BoK) consensus clustering will take O(n^2M) time to find best partitioning. As a result, the overall time complexity of this approach is in the order of O($(nNM + nKMI + n^2M)$) $= O(N + KI + n)nM =$ $O(NM)$.

5.4.2 Variations of the FRCC

1. **Low-dimensional datasets (fewer than 100 features):** The FRCC algorithm is employed for datasets with dimensions ranging from 8 (Wine) to 60 (Sonar). The final feature subset is determined by selecting the best cluster (for which the sum of weights is the greatest).

2. **Medium dimensional datasets (100 to 1000 features):** For medium dimensional datasets with features greater than 100, such as Musk1(168) and Musk2(168), initially algorithm FRCC is applied, and to reduce the subset of features, the K-means clustering algorithm is applied iteratively on the output cluster of the first iteration. The accuracy of the classifier is evaluated after each iteration. This step is repeated until there is no further decrease in the accuracy of the classifier when the selected features are used.

3. **High dimensional datasets(more than 1000 features):** In this scenario, variation 1 of the algorithm FRCC yields a feature subset with a significant number of features, whereas variation 2 of this approach necessitates additional repetitions and, as a result, increases the complexity of the algorithm. Hence, the top 1% of features from the best cluster containing all of the top-ranked features after one iteration are chosen. Then delete any features that are redundant. Colon(2000), Lymphoma(4026), and Leukemia(7129) datasets are used in the implementation of this method.

5.5 Experiments and Results

5.5.1 Datasets for FRCC

The proposed method FRCC is tested on benchmark datasets obtained from the UCI machine learning repository. PIMA, which has a modest number of features(8), and the

Table 5.1: Bench-mark data sets chosen from UCI.

Data set	# Features	# Instances	#Classes
PIMA	8	768	2
Wine	13	178	3
Zoo	16	101	7
Ionophere	34	352	2
Waveform	40	5000	3
Lungcancer	56	32	2
Spambase	57	4601	2
Sonar	60	500	2
Musk1	168	476	2
Musk2	168	6598	2

Table 5.2: Microarray high-dimensional datasets.

Data set	# Features	# Instances	#Classes
Colon	2000	62	2
Lymphoma	4026	66	3
Leukemia	7129	72	2

medium-dimensional data sets Musk1 and Musk2, which have a total of 168 dimensions, are among the data sets picked. Further, additional high-dimensional micro-array gene expression cancer datasets, such as Colon, which has 2000 features, Lymphoma, which has 4076 features, and Leukemia, which has 7129 features are used. Table 5.1 and Table 5.2 provide a summary of the data sets that are used in this study, respectively.

5.5.2 Implementation of FRCC

Five feature ranking algorithms from *Weka* are applied to generate different ranking criteria along with feature weights. These algorithms are InfoGain, Gainratio, ReliefF, OneR, and Symmetric Uncertainty evaluators. Then, K-means clustering algorithm is used to partition the features into groups based on their weights. The Dunn Index [52] is used to determine the optimal number of clusters. BOK is employed in order to determine the optimal partitioning. The final feature subset is chosen from the cluster that contains all of the features with the highest weight. 10-fold Cross validation is carried out for classification with the selected features from the data sets.

Figure 5.2 depicts the implementation of FRCC algorithm on Wine dataset as an ex-

ample.

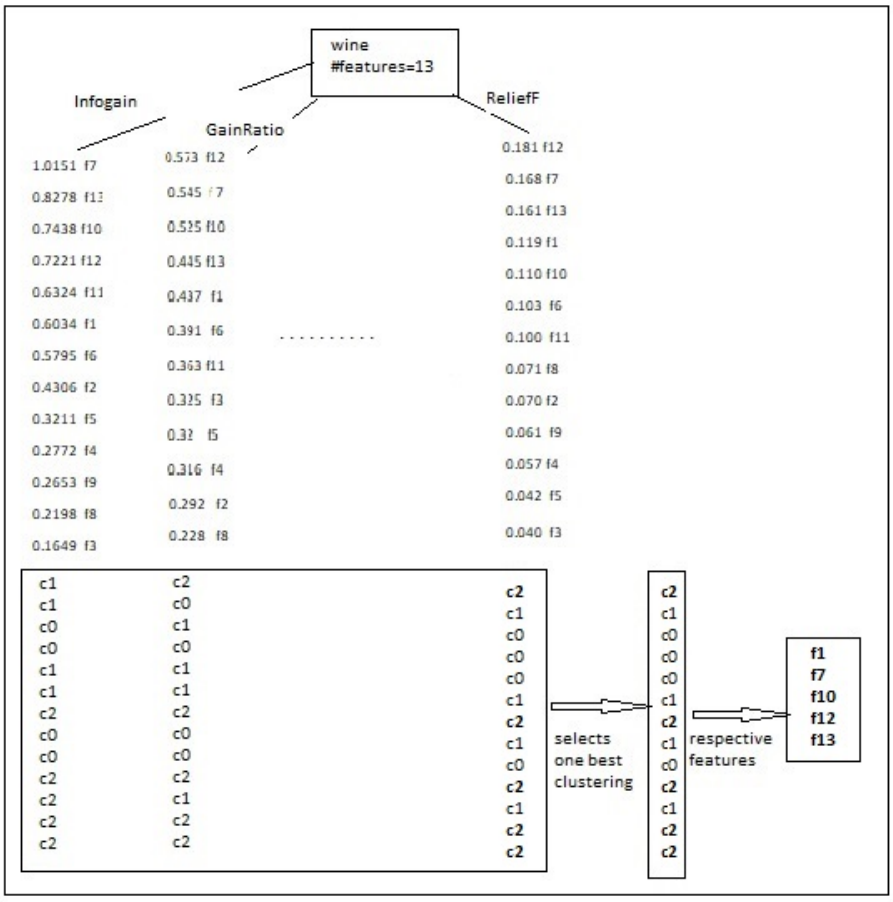

Figure 5.2: Steps of FRCC for Wine dataset.

However, for the medium dimensional datasets such as Musk1 and Musk2, the K-means clustering algorithm is applied repeatedly on the output of the previous iteration, resulting in a smaller number of features being selected each time. The accuracy of the classifier is evaluated after each iteration. Since this is a time-consuming process, to apply to the high dimensional micro-array datasets, only the top 1% of features are chosen from the output of the first iteration of the process. Further, redundant features are removed from the top 1% of features, and the accuracy of the classifier is evaluated using the Random Forest algorithm from *Weka*.

5.5.3 Results on UCI datasets

Performance of the FRCC method is shown in Table 5.3 and Table 5.4, which are all compared to a few recent methods available in the literature. Blanks in the tables indicate that

the results are not available for the corresponding data sets using those algorithms.

Table 5.3: Number of features selected by FRCC compared to latest literature

Dataset	**FRCC**	**GCACO** [71](2015)	**GCNC** [70](2015)	**UFSACO** [92](2014)	**RRFS** [24](2012)
Wine(13)	**5**	6	7	**5**	**5**
Ionosphere(34)	**8**	15	17	20	20
Spambase(57)	**4**	24	27	30	30
Sonar(60)	**7**	24	25	30	30
Colon(2000)	**5**	40	40	50	50

Table 5.4: Classifier accuracy of FRCC compared to latest literature

Dataset	**FRCC**	**GCACO** [71](2015)	**GCNC** [70](2015)	**UFSACO** [92](2014)	**RRFS** [24](2012)
Wine(13)	97.19	95.73	95.08	93.76	94.42
Ionosphere(34)	92.30	90.24	89.91	86.80	89.40
Spambase(57)	86.10	88.22	88.11	86.48	82.71
Sonar(60)	75.40	77.60	74.36	75.34	72.53
Colon(2000)	87.02	79.04	82.47	71.44	70.96

In the case of Wine and Ionosphere datasets, FRCC achieves lesser number of features as well as higher accuracy when compared to GCACO [71], GCNC [70], UFSACO [92], and RRFS [24] as shown in Table 5.3 and Table 5.4. With the exception of Wine and Zoo datasets, the number of features selected by other methods is more than twice that of FRCC. For Spambase and Sonar datasets, since the number of features selected by FRCC is far lesser, the accuracies are 2% lesser than those obtained by the other methods being compared.

Also the results are shown in Figure 5.3 and Figure 5.4.

5.5.3.1 Comparison with TCbGA

Table 5.5 shows the comparative performance of FRCC with a competitive GA based algorithm, namely TCbGA. As the results are available for all the datasets chosen by us, these are shown in a separate table.

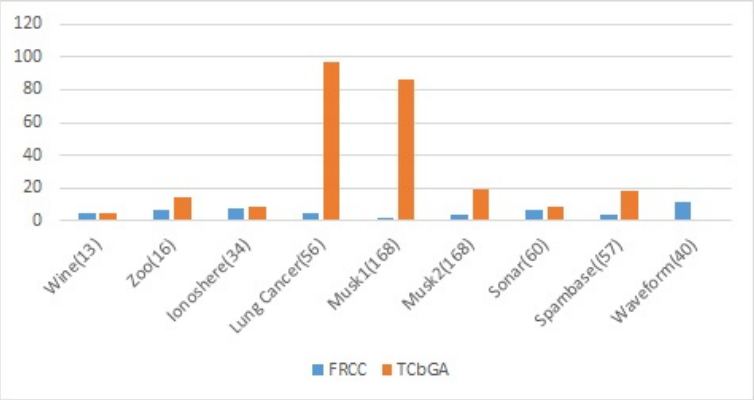

Figure 5.3: Number of features selected by FRCC method compared to literature.

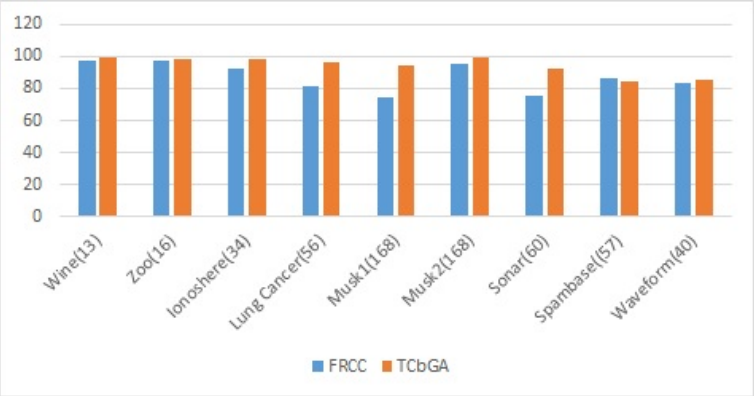

Figure 5.4: Classifier accuracy of FRCC method compared to literature.

Table 5.5: Performance of FRCC compared latest method TCbGA

Dataset	**FRCC**		**TCbGA**[61](2017)	
	Features	Accuracy	Features	Accuracy
Wine(13)	**5**	97.19	9	99.60
Zoo(16)	**5**	97.02	**5**	98.03
Ionosphere(34)	**8**	92.30	14	98.32
Lungcancer(56)	**5**	81.10	9	96.30
Musk1(166)	**6**	80.46	97	94.27
Musk2(166)	**4**	95.30	86	99.23
Spambase(57)	**4**	86.10	19	91.85
Sonar(60)	**7**	75.40	9	84.62
Waveform(40)	**13**	83.80	18	85.43

TCbGA is one of the latest algorithms with which we compare the performance of the proposed algorithm FRCC. When compared to TCBGA, in all cases, FRCC selects a significantly fewer number of features, as shown in Table 5.5. For low dimensional datasets like Wine, Ionosphere, Lung cancer, Spambase, Sonar, and medium dimensional

datasets like Musk1 the number of features selected by FRCC is significantly lesser than that of TCbGA[61]. For Musk2, number of features selected using FRCC is even lesser than 10%. Even though the classifier accuracy obtained by TCbGA is slightly higher than FRCC, as shown in the Table 5.5, the features chosen by FRCC are far fewer in number. It is to be noted that computational complexity of TCbGA is significantly higher than our algorithm. For example, when searching for the optimal feature subset of Sonar dataset with a moderate number of features (i.e. 60), the time taken by TCbGA was in hours, whereas the maximum time taken by FRCC for any high-dimensional dataset is in minutes.

5.5.3.2 Comparison with classical methods

Table 5.6: Comparison of FRCC with classical methods like SBC, GARIPPER, ReliefF, FCBF and NMIFS.

Dataset	**FRCC**		**Literature**	
	#Features	**Accuracy%**	**#Features**	**Accuracy%**
Pima(8)	**3**	76.80	5	79(SBC[14])
Wine(13)	**5**	97.19	7	97.40(GARIPPER[109])
Zoo(16)	7	**97.02**	7	96(GARIPPER[109])
Ionosphere(34)	**8**	92.30	10	94.60(GARIPPER[109])
Lung cancer(56)	5	81.10	5	87.00(ReliefF[64])
Musk1(168)	**2**	**74.30**	25	74.00(WBFS[56])
Musk2(168)	4	**95.30**	2 2 10 25	91.33(FCBF[110]) 94.6(ReliefF[64]) 95.5(CFS-SF[35]) 96.35(FCBF-P[110])
Waveform(40)	13	**83.80**	13	81.52(NMIFS[23])
Spambase(57)	4	**86.10**	3	75.8(NMIFS[23])
Sonar(60)	**7**	75.40	11	86.36(NMIFS[23])

Table 5.6 gives the comparative results of FRCC with some of the classical feature subset selection algorithms. With the exception of Lung cancer, the number of selected features is lower in most cases when compared to the literature, and the classifier accuracies are comparable. Further, the computational complexity of the proposed method is much lower.

In the majority of the datasets, the reduction in the number of features selected is about 50% compared to those reported in the latest literature. Further, the classifier accuracy obtained with these features is on par with the literature.

5.5.4 Results on microarray datasets

Results are available in the latest literature for the high dimensional microarray datasets. Hence performance of FRCC is compared with some of these latest algorithms and the results are tabulated in Table 5.7. Also the results are plotted in Figure 5.5 and Figure 5.6.

It can be clearly seen from the results that the proposed FRCC algorithm outperforms all the other algorithms on micro array datasets. The accuracy of the FRCC classifier is higher when compared to the literature for the Colon and Leukemia datasets, and it is nearly the same for the Lymphoma dataset. However, when compared to the literature, the number of features selected by the FRCC for Colon and Lymphoma cancer is very small. On Lymphoma dataset, using only 12 features, FRCC was able to achieve nearly the same accuracy as the ensemble ranking (EnsRank) approach [80] which selects 80 features. In the case of leukaemia dataset, FRCC selects 26 features, which is higher when compared to FDT [96] method. To summarize, the number of features obtained is less than 0.5% to the total features and classifier accuracy is higher than the literature. The computational complexity of FDT is extremely high when compared to the computational complexity of FRCC. To summarise, the number of features that were ultimately chosen is very small when compared to the literature, accounting for less than 1 percent of the total number of features.

Table 5.7: Number of features selected by FRCC and accuracy on microarray datasets compared to the latest literature.

Dataset	**FRCC**		**EnsRank [80]**		**FDT [96]**	
	#F	Acc%	#F	Acc%	#F	Acc%
Colon(2000)	5	87.02	-	-	6	80.20
Lymphoma(4096)	12	96.96	80	97.20	-	-
Leukemia(7129)	26	95.83	-	-	4	87.50

In terms of competitiveness, the FRCC algorithm outperforms other methods on microarray data sets. A predefined threshold value is required by the majority of the methods described in the literature in order to select the final feature subset, and this threshold value varies depending on the total number of dimensions or features. For FRCC, however, there

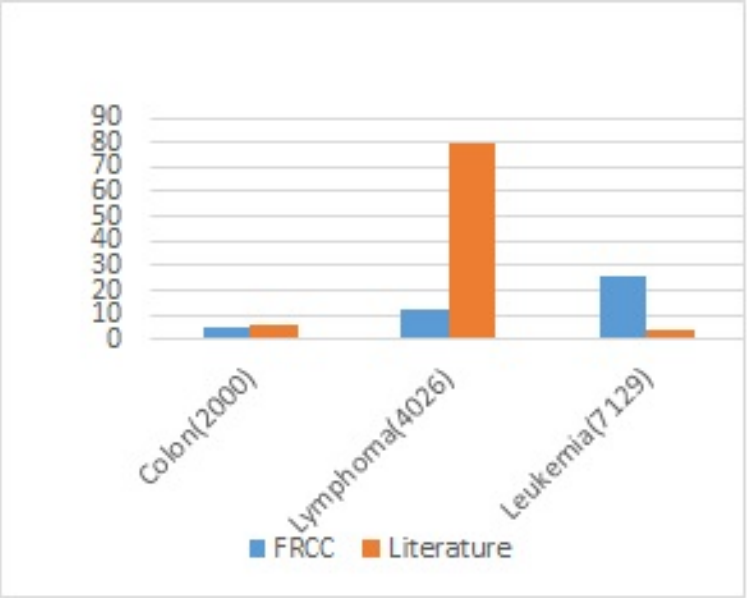

Figure 5.5: Number of features selected by FRCC on microarray datasets compared to literature.

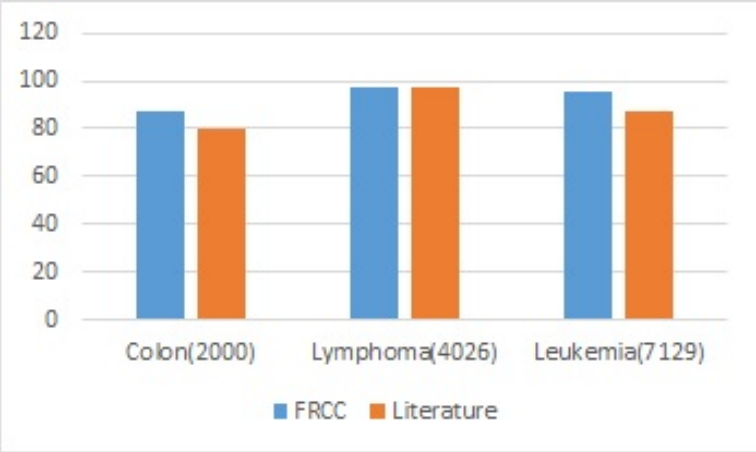

Figure 5.6: Classification accuracy of FRCC on microarray datasets compared to literature.

is no requirement for a pre-defined threshold value. The algorithm will determine the final feature subset to be used. We only use thresholds when dealing with extremely large datasets. The results clearly demonstrate that the features selected by the FRCC method are far superior to those found in the literature.

5.5.5 Robustness of FRCC

To verify the robustness of FRCC algorithm with different consensus clustering algorithms, FRCC has been re-implemented using two new recent consensus clustering algorithms namely LWEA and WHAC presented by Huang et al., [41] and Banerjee et al., [9]

respectively.Results are presented in Table 5.8. It is validated from the results obtained that FRCC is robust to the choice of the consensus clustering algorithm used.

Table 5.8: Results obtained by FRCC using BoK, LWEA and WHAC consensus clustering algorithms.

Dataset	FRCC+BoK		FRCC+LWEA		FRCC+WHAC	
	#Features	Accuracy%	#Features	Accuracy%	#Features	Accuracy%
Wine(13)	5	97.19	6	98.23	6	98.23
Zoo(16)	5	97.02	5	97.02	5	97.02
Ionosphere(34)	8	93.30	7	93.12	7	93.12
Lung cancer(56)	5	81.10	5	81.10	4	79.87
Musk1(168)	6	80.46	4	80.10	6	84.45
Musk2(168)	4	95.30	4	95.30	4	95.30
Spambase(57)	4	86.10	4	86.10	4	86.10
Sonar(60)	7	75.40	4	75.40	4	75.40
Waveform(40)	13	83.80	8	81.53	8	81.53
Colon(2000)	5	87.02	5	87.02	5	87.02
Lymphoma(4096)	12	96.96	12	96.96	12	96.96
Leukemia(7129)	26	95.83	26	95.83	26	95.83

5.6 Application of FRCC to Big data

Big data has the following three characteristics: a large amount of data, a wide variety of data, and a rapid change in data [55, 90, 84, 63]. It is important to note that the magnitude and complexity of the data that constitutes big data evolves over time.

In terms of diversity, it is found that there is a huge data to be mined with a wide variety of features in a variety of domains, including astronomy [22], Internet [77][13], geo-informatics [73], biomedicine [59], wireless sensor networks [54], crowdsensing [108] and the Internet of things [53]. To extract knowledge from a multi-dimensional data set when dealing with a large amount of data, feature selection becomes essential. However, the ever-increasing volume of data presents significant issues for feature selection. As a result, in recent years many feature selection algorithms have been proposed [101][105].

A number of feature selection methods have been developed, including filter, wrapper, and embedding approaches [4]. Scalability is a major concern for big data processing systems. The other challenges are due to the massive redundancy or irrelevance, which not only consumes computing resources but also affects processing performance. If such features are deleted while valuable features are retained, the dimension of big data will be

substantially reduced, and as a result, the performance of algorithms on big data will be enhanced, in addition to achieving computational efficiency. According to Yu et al. [44], a good feature selection algorithm should select a subset of features that are highly correlated with class variable and must give optimal classification results. The FRCC algorithm is a suitable candidate that can be used for big data by incorporating the diversity at both data level and at the algorithm level leading to a hybrid approach.

5.6.1 Motivation

Due to the scalability of FRCC algorithm, we apply this consensus method to the large scale domains like big data. As suggested by Barbara Pes[80], in this hybrid feature selection algorithm diversity is incorporated both at data level and algorithm level. Data level diversity is shown by dividing dataset into samples and algorithm level diversity is achieved by applying different feature ranking algorithms on each sample. Final feature subset is obtained by performing union and intersection operations on feature subsets obtained from each sample .

5.6.2 Related work

Recently, Kong et al. [49] proposed a distributed fuzzy rough set (DFRS) method in cloud computing to meet the growth of big data. The main idea of DFRS is to break down large amounts of data into smaller partitions, each of which is assigned to a cloud node to process the fuzzy rough set. The main challenge is to sensibly distribute processing jobs to different nodes while maintaining and sharing global data with separate memory resources. To solve this problem, they developed a data decomposition algorithm called in DFRS that dynamically allocates samples based on the processing resources of each distributed node. The algorithm ensures that all essential interrelations are traversed, which means that any interrelation inside a sub-dataset is evaluated by at least one distributed node. Data decomposition module was followed by update positive region and reduct-mergence modules. They implemented DFRS on 17 datasets from UCI machine learning repository. Memory and run time are the two major concerns in the experiments to assess the performance of the DFRS algorithm. The number of selected features and classifier accuracy are also used as performance measures and DFRS is compared with non-DFRS(centralized)

method.

5.6.3 Hybrid feature subset selection (HFS) algorithm for Big data

Step 1: Initially the dataset D with large number of instances(N) and high dimensions(M) is divided into smaller samples D_1, D_2,....D_P, where every sample contains nearly same number of instances that covers all classes in the dataset.

Step 2: On each sample D_i, FRCC algorithm which is described in detail in section 5.4 is applied as described below:

- Five popular feature ranking algorithms namely, Information Gain(IG), Gain Ratio(GR), OneR, ReliefF and Symmetric Uncertainty(SU) are used to get the ranking of features along with the feature weights.
- Each ranking output is clustered using K-Means clustering algorithm by considering feature weight as a single dimension (K-value is chosen using Dunn Index(DI)).
- As different ranking algorithms are applied on the sample, the output clusterings may also differ. To find the best clustering from the ensemble of five clusterings, two experiments have been done with two different consensus clustering algorithms. One is the state of the art algorithm Best-of-K(BoK) and the other one is latest consensus clustering algorithm WHAC [9] suggested by Banerjee et al(2021). The optimal and robust feature subset of the sample D_i is selected from the cluster(which has largest total weight) of the best clustering.

Step 3: After generating 'P' stable or robust feature subsets FS_1, FS_2,....FS_P from each sample D_i $(i \leq P)$, union or intersection is performed to get the final feature subset.

Figure 5.7 and Figure 5.8 depict the Hybrid feature selection algorithm. Hybrid feature selection algorithm is given in Algorithm 5.4.

To test the performance of this algorithm with the selected final feature subset, k-Nearest Neighbour(kNN) classifier is used.

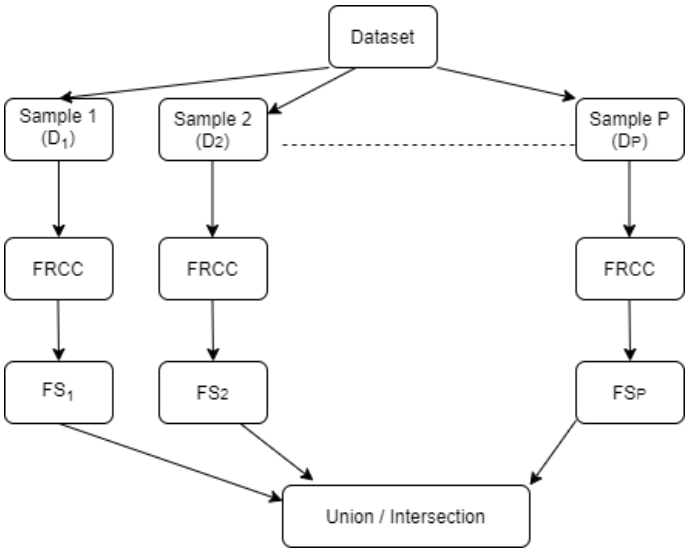

Figure 5.7: Diagram of Hybrid Feature Selection Algorithm

Algorithm 5.4 Hybrid feature selection algorithm for big data

Input: Data set is X=<**x**,**t**>. Size of the dataset is N and feature set is F, where F=$\{f_1, f_2, f_3, \ldots, f_M\}$ and t is the target vector, W=$\{w_1, w_2, w_3, \ldots, w_M\}$, FRCC

Output: Final Feature subset FFS

1: Divide the dataset D into P number of samples with nearly equal number of instances in each sample D_x
2: $FFS \leftarrow \emptyset$;
3: **for all** $x \leftarrow 1$ *to* P **do**
4: Apply FRCC(D_x); ▷ returns $FS(x)$
5: $FFS \leftarrow FFS \cup FS(x)$;
6: **end for**
7: **return** FFS; ▷ return final feature subset

5.6.4 Complexity analysis

As HFS approach repeats FRCC for P number of times, and the time complexity of FRCC is O(NM), thus the total time required to implement hybrid feature selection is O(PNM).

5.6.5 Experiments and results for HFS

5.6.5.1 Datasets used to implement HFS

To evaluate the performance of Hybrid Feature Selection algorithm, experiments are conducted on real-world datasets from UCI repository. Datasets are selected with number of features ranging from 21 to 561 and number instances are ranging between 5000 and 58509. K-Nearest Neighbour classifier is used to evaluate performance of the algorithm. Table 5.9 gives the information about datasets used in this experiment.

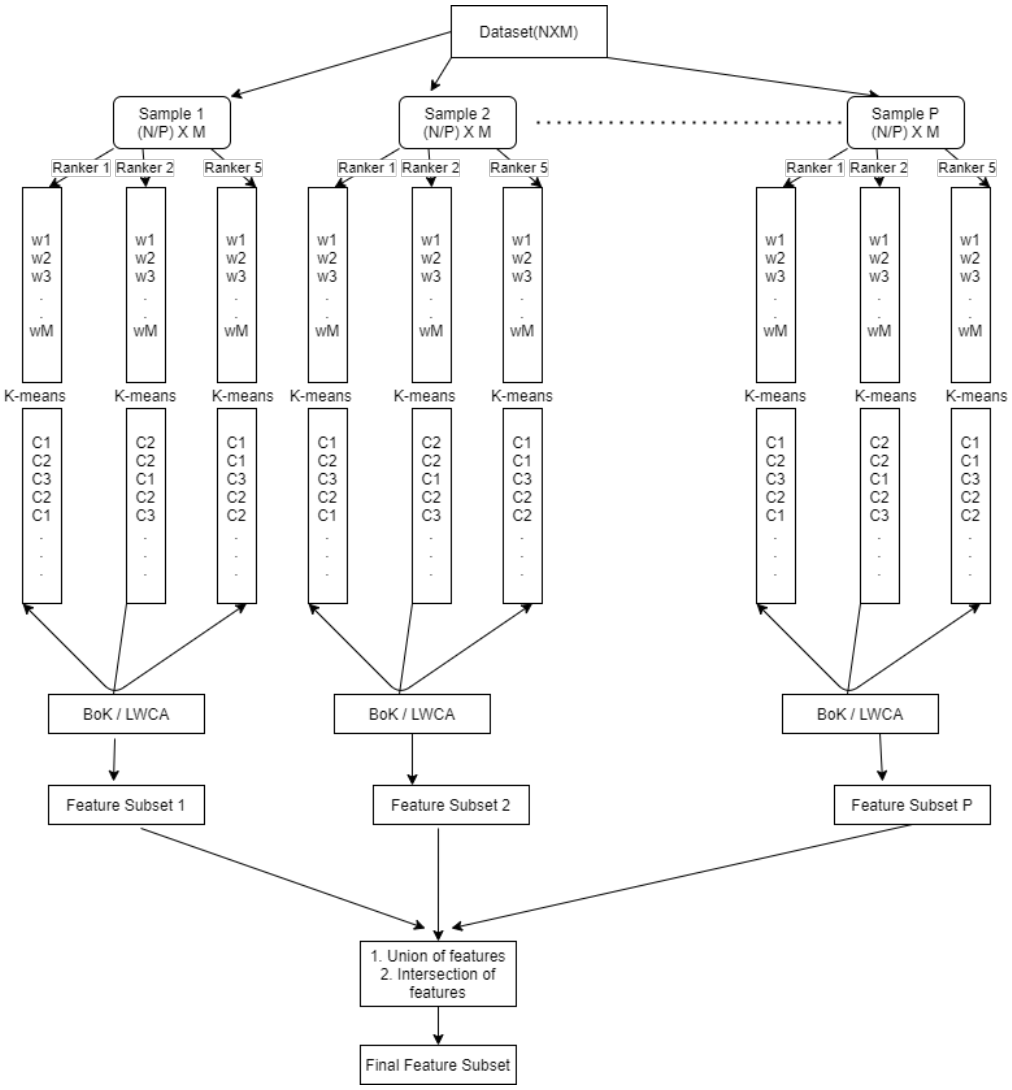

Figure 5.8: Diagram of Hybrid Feature Selection Method

5.6.5.2 Implementation of HFS

Initially, each dataset D is partitioned into P samples with nearly same number of instances in each sample. Then, on each sample FRCC is performed. In particular, Waveform dataset has only 5000 instances, it is divided into two samples, each with 2500 instances. The two partitions of HAPT dataset have 5465 and 5464 number of instances. Diagnosis dataset is partitioned into 10 samples, where each sample contains 5851 instances.

FRCC is applied on each partition of the dataset. As the number of features of HAPT is 561, we adopt the high-dimensional variation of FRCC as given in subsection 5.4.2 by applying the threshold of 10% of total features is selected from the best cluster. Then union or intersection is applied on the feature subsets obtained on each partition to get the final feature subset. The accuracy obtained by k-Nearest Neighbour(kNN) classifier is used to

Table 5.9: Description of datasets from UCI

S.No	Data set	# Instances	# Features	#Classes
1	Waveform	5000	21	3
2	HAPT	10929	561	12
3	Diagnosis	58509	49	11

Table 5.10: Results of HFS-BoK using Union and intersection operations compared to DFRS

Dataset	HFS+Union		HFS+Intersection		DFRS	
	#Features	Accuracy	#Features	Accuracy	#Features	Accuracy
Waveform	12	79.26%	10	78.24%	17	77.78%
HAPT	66	94.30%	42	94.93%	347	94.49%
Diagnosis	24	99.92%	13	99.83%	26	–

test the performance of the algorithm.

Table 5.10 shows the results obtained by HFS algorithm.

Additional experimentation is carried out by replacing BoK with other consensus clustering algorithms in the module of FRCC in HFS. The algorithm is termed as HFS+WHAC. The results are given in Table 5.11.

The proposed HFS algorithm is applied on three UCI machine learning datasets with instances varying from 5000 to 58509. Here, hybrid feature selection algorithm is combined with Best-of-K(BoK) consensus algorithm and also with Weighted Hierarchical Agglomerative Clustering(WHAC) consensus algorithm. The results are given in Table 5.10 and Table 5.11. When compared to one of the latest algorithms on big data, namely DFRS method [49], HFS-BOK and HFS-WHAC are selecting very less number of features for all the three data sets. Reduction obtained by HFS for Waveform data having 21 features

Table 5.11: Results of HFS-WHAC using Union and intersection operations compared to DFRS

Dataset	(HFS-WHAC)+Union		(HFS-WHAC)+Intersection		DFRS	
	#Features	Accuracy	#Features	Accuracy	#Features	Accuracy
Waveform	13	81.54%	10	78.24%	17	77.78%
HAPT	75	94.56%	48	94.28%	347	94.49%
Diagnosis	24	99.92%	13	99.83%	26	–

Table 5.12: Results of FRCC compared to DFRS

Dataset	FRCC		DFRS	
	#Features	Accuracy	#Features	Accuracy
Waveform(21)	11	78.84%	17	77.78%
HAPT(561)	56	95.83%	347	94.49%
Diagnosis(49)	19	99.92%	26	–

Table 5.13: Result of HFS on Cencus income dataset.

Dataset	HFS		Complete dataset	
	Reduced Features	Accuracy	Full Features	Accuracy
Census income(41)	13 (Union) 5 (Intersection)	94.75% 94.71	41	94.85%

is about 50%; for HAPT having 561 features, obtained is as low as 8% and for Diagnosis data set having 49 features about 25%; And with respect to the literature, the reductions obtained are 60%, 15% and 50% respectively. It is observed from the Table 5.10, Table 5.11 that, there is a significant improvement in the classifier accuracy for Waveform data with less number of features compared to literature. On the other data sets, the accuracy obtained is on par with the literature. Further it is to be noted that the DFRS algorithm which adopts a fuzzy rough set approach takes much higher time than our proposed greedy approach.

FRCC is a scalable approach. Further, is is applied on datasets described in Table 5.9 and the results are compared with latest method DFRS. The results are tabulated in Table 5.12.

The results clearly show that FRCC achieves a significant feature reduction for all the data sets as compared to DFRS, and also shows a slight improvement in the accuracy.

5.6.6 HFS on large scale dataset

HFS is further implemented on large scale dataset *Adult census income dataset* [21] which has 299285 samples and 41 features. This contains numerical and categorical features.

The results obtained by HFS with BoK consensus clustering are shown in Table 5.13.

This is a binary class dataset having 75% Class A (income 50K$) instances and 25% Class B (income 50K$) instances. We divide the dataset into 50 samples, with each sample having the number of instances having same class ratio.Then FRCC is applied on each

sample to obtain feature subset from each sample. Finally union and intersection is performed on all feature subsets to obtain final feature subset. Results from Table 5.13 shows that, HFS is giving 70% feature reduction and intersection is giving 80% feature reduction on *Cencus income* dataset with nearly same classifier accuracy. 10-fold cross validation is performed using random forest classifier to test the classifier accuracy. Further, results shows that HFS can be considered as a scalable approach and is applicable to largescale datasets.

5.7 Conclusions

Two methods are proposed to deal with the problems of high-dimensionality and bigdata. FRCC is a feature selection algorithm that is working well on datasets having instances upto 5000 and features upto 7129. FRCC method comes under heterogeneous ensemble approach (multiple algorithms implemented on same data). FRCC is selecting less number of features compared to many recent methods in the literature and accuracy is high for most of the datasets. It is giving significantly better feature reduction as well as accuracy on the high-dimensional microarray datasets. However, large datasets that are common to bigdata problems definitely pose a great challenge for feature selection. There is a need for decomposition of data into smaller samples and to develop a robust and stable feature selection algorithm. Recent study in the field of big data has revealed that the present algorithms are often insufficient in terms of stability with respect to changes in the input data. Because of the robustness, it will have practical consequences for distributed applications in which the algorithm must give reliable results across a large number of different data samples. To address this problem, we extend FRCC algorithm to a hybrid feature selection (HFS) algorithm that has diversity at both data level(sampling) and algorithm level. HFS algorithm shows good performance on three UCI machine learning data sets that are taken from the literature, namely, Waveform, HAPT and Diagnosis. Feature reduction of HFS on Census income dataset is very high with HFS-intersection while maintaining nearly same accuracy as original dataset.

Chapter 6

Conclusions and future scope

6.1 Conclusions

We address the problem of feature subset selection using consensus clustering method. So far, consensus clustering has been used in the literature to find the best clustering among various input clusterings. As per our knowledge, we are the first to use consensus clustering method to find the best feature subset. Based on this idea, we validate that, a robust feature subset can be obtained by applying consensus among various feature subsets. To check the feasibility of applying consensus clustering to feature selection problem, a novel feature selection algorithm called genetic algorithm based feature selection using consensus clustering (GACC) is devised.

Next, to deal with irrelevant and redundant features that greatly affect the classifier accuracy a graph based approach is proposed. By modeling the feature space as a graph, we apply community discovery algorithms for graph partitioning in order to obtain relevant and non-redundant features using consensus clustering (CDCC).

Most of the feature selection algorithms are not scalable for high dimensional datasets. To address this issue, a fast and scalable approach called feature ranking based feature selection algorithm using consensus clustering (FRCC) is proposed. This method uses various feature ranking algorithms to rank the features and then features are clustered using K-means algorithm in the single dimensional space of feature weights given by each feature ranking algorithm. The intuition behind this clustering is that all the top ranked features may form one cluster. Optimal feature subset is selected from the best partitioning

obtained by the consensus clustering.

Among these three approaches FRCC is more scalable. So, we devise an algorithm for big data by dividing the dataset into manageable samples and applying FRCC on each sample. Then applying consensus clustering algorithm to obtain an optimal feature set from all the samples. This leads to a hybrid feature selection algorithm (HFS).

Experimentation using all these approaches is carried out on benchmark datasets from UCI machine learning repository. To check the performance of our approaches, we divide the datasets into four categories as follows:

1. Low number of features with low number of instances. Eg: Wine and Zoo.
2. Low number of features with high number of instances. Eg: Pima.
3. High number of features with low number of instances. Eg: Ionosphere and Lung-cancer.
4. High number of features with high number of instances. Eg: Musk1, Musk2 and Isolet.

Table 6.1 shows the performance of GACC, CDCC and FRCC. CDCC is implemented with three variations using $|r|$, r^2 and SU as edge weights. Due to its scalability FRCC is also implemented on high-dimensional micro-array datasets. Further, HFS is implemented on big datasets to test the efficiency of the algorithm.

6.2 Comparison of all approaches

From the Table 6.1 it can be concluded that, overall FRCC shows a superior performance both in terms of feature reduction and classifier accuracy on all the data sets. Specifically, for category 1 datasets, FRCC is giving better classifier accuracy, but CDCC-$|r|$ is selecting less number of features with comparable accuracy. In category 2, CDCC-r^2 is giving better reduction in number of features compared to other methods. As FRCC is selecting more features than CDCC-r^2, accuracy is also high. In category 3, CDCC-$|r|$, CDCC-r^2 and CDCC-SU are giving same result for Ionospere with less features and high accuracy. FRCC is performing better in case of lungcancer dataset. In category 4, GACC

Table 6.1: Comparison of all three approaches

Dataset	GACC		CDCC						FRCC	
			$\|r\|$		r^2		SU			
	#F	Acc	#F	Acc	#F	Acc	#F	Acc	#F	Acc
Wine(13)	7	95.69	4	94.90	6	96.16	5	96.08	5	97.19
Zoo(16)	8	96.18	4	95.3	5	95.2	5	93.5	7	97.02
Pima(8)	4	75.2	4	76.8	2	75.9	4	75.6	3	76.8
Ionosphere(34)	10	94.4	5	93.3	5	92.3	6	92.3	5	92.3
Lungcancer(56)	8	79.5	5	65.6	5	65	5	78.1	5	81.1
Musk1(168)	–	–	4	74.4	6	76.8	4	75.4	2	74.3
Musk2(168)	–	–	2	93.8	4	94.08	4	93.8	4	95.3
Isolet(617)	127	72.45	113	71.85	107	68.36	118	78.1	62	81.1

is selecting very high number of features compare to all methods. FRCC is performing better compared to all methods.

The effectiveness of FRCC can be seen by its performance on high dimensional microarray datasets belonging to category 3. For example, on Leukemia dataset having 7129 dimensions, FRCC selects 0.36% of the features and achieves classifier accuracy of 95.83% which is superior to the accuracies reported in the literature. The scalability of HFS is demonstrated on a big data set namely, *census income* with HFS selecting only 12% of the features and achieving an accuracy of 94.7% which is on par with the accuracy attained when all features are used.

6.3 Future work

In future we want to work on high-dimensional datasets which have large number of instances. Further, feature subset selection using consensus clustering can be used in the problems like anomaly detection, opinion polling, biclustering etc. In the recent literature, anomaly detection [7, 107] is addressed using ensemble methods, but they require predefined thresholds and cannot be applied on large scale datasets. Application of HFS on such datasets may produce more robust results. Biclustering is another area in which HFS can be applied, as the recent work proposed in this area is not scalable [62].

Bibliography

[1] Rstudio: Integrated development environment for r. 2012.

[2] C Moore A Clauset, MEJ Newman. Finding community structure in very large networks. *Physical Review E, 70(6)*, 2004.

[3] H. Mannila A. Gionis and P. Tsaparas. Clustering aggregation. In *Proceedings of 21st International Conference on Data Engineering (ICDE)*, page 341–352, 2005.

[4] N. Abd-Alsabour. A review on evolutionary feature selection. *In Proceedings of the 2014 European Modelling Symposium. IEEE Computer Society*, page 20–26, 2014.

[5] A.Gionis, H.Mannila, and P.Tsaparas. Clusetering aggregation. In *Proceedings of International Conference on Data Engineering*, pages 341–352. ACM, 2005.

[6] A.Goder and V. Filkov. Consensus clustering algorithms : Comparision and refinement. In *Proc. SIAM International Conference on Data Mining*, pages 109–118. SIAM, 2008.

[7] Soulib Ghosh Neeraj Kumar and Ram Sarkar Akash Saha, Agneet Chatterjee. An ensemble approach to outlier detection using some conventional clustering algorithms, multimedia tools and applications. *Multimedia tools and applications*, 2020.

[8] Bhadra. T. Mktra P. Bandyopadhyay, S. and U. Maulik. Integration of dense subgraph finding with feature clustering for feature selection. *Pattern Recognition Letters*, 40:104–112, 2014.

[9] Arko Banerjee, Arun K. Pujari, Chhabi Rani Panigrahi, Bibudhendu Pati, Suvendu Chandan Nayak, and Tien-Hsiung Weng. A new method for weighted ensem-

ble clustering and coupled ensemble selection. *Connection Science*, 33(3):623–644, 2021.

[10] R Bellman. Adaptive control processes: A guided tour. In *Princeton University Press.*, 1961.

[11] Michael Bertolacci and Anthony Wirth. Are approximation algorithms for consensus clustering worthwhile? pages 437–442. SIAM, 2007.

[12] Delling D. Gaertler M. Gorke R. Hoefer M. Nikoloski Z.- Wagner D Brandes, U. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering, 20(2)*, page 172–188, 2008.

[13] N. Cullot C. K. Emani and C. Nicolle. Understandable big data: A survey. *Computer Science Review*, 17:70–81, 2015.

[14] C.A.Ratanamahatana and D. Gunopulos. Feature selection for the naive bayesian classifier using decision trees. In *applied artificial intellegence*, volume 17, pages 475–488. Taylor and Francis, 2006.

[15] William Eberle Chih-Fong and Chi-Yuan Chu. A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing*, 74:2914–2928, 2011.

[16] William Eberle Chih-Fong and Chi-Yuan Chu. Genetic algorithms in feature and instance selection. *Knowledge Based Systems*, 39:240–247, 2013.

[17] huan-Yu Chen Chuen-Horng Lin and Y-S. Wua. Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection. *Expert System Applications*, 41:6611–6621, 2014.

[18] Michele Coscia. Discovering communities of community discovery. In *International Conference on Advances in Social Networks Analysis and Mining*. ACM, 2019.

[19] P. H. Dubois D. Rough fuzzy sets and fuzzy rough sets. *International Journal of General System*, 17:191–209, 1990.

[20] S. Zhao Q. Hu D. Chen, L. Zhang and P. Zhu. A novel algorithm for finding reducts with fuzzy rough sets. *IEEE Transactions on Fuzzy Systems*, 20:385–389, 2012.

[21] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[22] K. Jay Edwards and M. Gaber. Astronomy and big data: A data clustering approach to identifying uncertain galaxy morphology. *Springer*, 2014.

[23] Pablo A. Estevez, Michel Tesmer, Claudio A. Perez, and Jacek M. Zurada. Normalized mutual information feature selection. In *IEEE Transactions on pattern analysis and machine intelligence*, volume 20, pages 189–201. IEEE Computer Society, 2009.

[24] Artur J. Ferreira and M.A.T¿ Figueiredo. An unsupervised approach to feature descritization and selection. *Pattern recognition*, 45:3048–3060, 2012.

[25] Vladimar Filkov and Steven Skiena. Heterogeneous data integration with the consensus with the consensus clustering formalism. In *Proceedings of Data Integration in the Life Sciences*, pages 110–123, 2004.

[26] Vladimar Filkov and Steven Skiena. Integrating microarray data by consensus clustering. *Journal of Artificial Intellegence Tools*, pages 863–880, 2004.

[27] G. Forman. An extensive empirical study of feature selection metrics for text classification. *ournal of Machine Learning Research*, 3:1289–1305, 2003.

[28] S. Fortunato. Community detection in graphs. *Physics Reports, 486(3-5)*, page 75–174, 2010.

[29] L.C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1:215–239, 1979.

[30] Joydeep Ghosh and Ayan Acharya. Cluster ensembles. In *WIRE's Datamining Knowledge discovery*, volume 1, pages 305–315. John Wiley and Sons, 2011.

[31] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. In *Proceedings of National Academic Sci.USA*, pages 7821–7826, 2002.

[32] Li Zhenhui Gu, Quanquan and J. Han. Generalized fisherscore for feature selection. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence.*, 2011.

[33] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Machine Learning Research*, 3:1157– 1182, 2003.

[34] K-J. Kim H. Ahn. Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach. *Appl. Soft Comput.*, 59:599–607, 2009.

[35] M. Hall. Correlation-based feature selection for machine learning. *PhD thesis, Citeseer*, 1999.

[36] M. A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering, 15(6)*, page 1437–1447, 2003.

[37] Jiawei Han and Micheline Kamber. *Data mining Concepts and techniques*. Morgan Kaufmann Series in Data Management Systems, 1999.

[38] Cai-Deng He, Xiaofei and Niyogi1. Laplacian score for feature selection. *Adv. Neural Inf. Process*, 18:507–514, 2005.

[39] J. H. Holland. Genetic algorithms and adaptation. *Adaptive Control of Ill-Defined Systems*, page 317–333, 1984.

[40] Yi Hong, Sam Kwong, Yuchou Chang, and Qingsheng Ren. Consensus unsupervised feature ranking from multiple views. *Pattern Recognition Letters*, 29(5):595–602, 2008.

[41] Dong Huang, Chang-Dong Wang, and Jian-Huang Lai. Locally weighted ensemble clustering. *IEEE Transactions on Cybernetics*, 48:1460–1473, 05 2018.

[42] Witten IH and Frank E. Data mining: practical machine learning tools and techniques. *Morgan Kaufmann, 2nd Edition, Burlington*, 2005.

[43] Hua Zhong Jaehong Yu and Seoung Bum Kim. An ensemble feature ranking algorithm for clustering analysis. *Journal of Classification, Springer*, 2019.

[44] W. Ding K. Yu, X. Wu and J. Pei. Towards scalable and accurate online feature selecion for big data. *2014 IEEE International Conference on Data Mining*, pages 660–669, 2014.

[45] Antoine Cornuejols Elena Marchiori Kees Jong, Jeremie Mary and Michele Sebag. Ensemble feature ranking. *Knowledge discovery in databases:PKDD*, pages 267–278, 2004.

[46] Kenji Kira and Larry A Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of AAAI*, 1992.

[47] R. Kohavi and John. Wrappers for feature subset selection. *Artificial Intelligence*, pages 273–324, 1997.

[48] Ron kohavi. Scaling up accuracy of naive bayes classifier: a decision tree. *Knowledge Discovery and Datamining(KDD)*, pages 202–207, 1996.

[49] Linghe Kong, Wenhao Qu, Jiadi Yu, Hua Zuo, Guihai Chen, Fei Xiong, Shirui Pan, Siyu Lin, and Meikang Qiu. Distributed feature selection for big data using fuzzy rough sets. *IEEE Transactions on Fuzzy Systems*, PP:1–1, 2019.

[50] I. Kononenko. Estimating attributes: Analysis and extensions of relief pages. *In Proceedings of the European conference on Machine Learning, ECML, Secaucus, NJ, USA, Springer-Verlag*, page 171–182, 1994.

[51] Petri Kontkanen, J.Lahtinen, P.Millymaki, and Henry Tirri. Unsupervised bayesian visualization of high-dimensional data. *In Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining(KDD)*, pages 325–329, 2000.

[52] Kovacs, F. Legany, and A Babos. Cluster validity measurement techniques. In *Proceedings of the 6th International Symposium of Hungarian Researchers on Computational Intelligence*, pages 18–19, 2005.

[53] F. Wu G. Chen L. Kong, M. K. Khan and P. Zeng. Millimeter-wave wireless communications for iot-cloud supported autonomous vehicles: Overview, design, and challenges. *IEEE Communications Magazine*, 55:62–68, 2017.

[54] X.-Y. Liu G. Chen Y. Gu M.-Y. Wu L. Kong, M. Xia and X. Liu. Data loss and reconstruction in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 25:2818–2828, 2014.

[55] Z. He Q. Xiang J. Wan L. Kong, D. Zhang and M. Tao. Embracing bigdata with compressive sensing: A green approah in industrial wireless networks. *IEEE communications magazine*, 54:53–59, 2016.

[56] J Leng, C Valli, and L Armstong. A wrapper-based feature selection for analysis of large data sets. In *Proceedings of 3rd International Conference on Computer and Electrical Engineering(ICCEE)*, pages 167–170. IEEE, 2010.

[57] H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes. *Proceedings of the Seventh International Conference on Tools with Artificial Intelligence, IEEE*, pages 388–391, 1995.

[58] H. Liu and R. Setiono. A probabilistic approach to feature selection: A filter solution. *Proc. 13th Int'l Conf. Machine Learning*, pages 319–327, 1996.

[59] C. Lynch. Big data: How do your data grow?". *Nature*, 455:28–29, 2008.

[60] Robnik-Sikonja M and Kononenko I. Theoretical and empirical analysis of relief and rrelieff. *Journal of Machine Learning*, 53:23–69, 2003.

[61] B. Ma and Y. Xia. A tribe competition-based genetic algorithm for feature selection in pattern classification. *Appl. Soft Comput.*, 58:328–338, 2017.

[62] Julieta Sol Dussaut Ignacio Ponzoni Maria Jimena Martinez. Biclustering as strategy for improving feature selection in consensus qsar modeling. *Electronic notes in Discrete Mathematics*, 69:117–124, 2018.

[63] V. Mayer-Schönberger and K. Cukier. Big data: A revolution that will transform how we live, work, and think. *Houghton Mifflin Harcourt*, 2013.

[64] W. Megchelenbrink, Elena Marchiori, and Peter Lucas. Relief-based feature selection in bio-informatics:detecting functional specificity residues from multiple sequence alignments. *Master Thesis, Radboud University, Nijmegen*, 2010.

[65] T.M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[66] M. Modrzejewski. Efficiently inducing determinations: A complete and systematic search algorithm that uses optimal pruning. *Proc. 10th Int'l Conf. Machine Learning*, pages 284–290, 1993.

[67] M. Modrzejewski. Feature selection using rough sets theory. *Proc. European Conf. Machine Learning*, pages 213–226, 1993.

[68] Mandal Monalisa and Mukhopadhyay A. Unsupervised non-redundant feature selection: a graph-theoretic approach. In *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications(FICTA).*, pages 373–380, 2013.

[69] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, pages 91–118, 2003.

[70] Parham Moradi and Mehrdad Rostami. A graph theoretic approach for unsupervised feature selection. *Engineering Applications of Artificial Intelligence*, 44:33–55, 2015.

[71] Parham Moradi and Mehrdad Rostami. Integration of graph clustering with ant colony optimization for feature selection. *Knowledge based systems*, 84:144–161, 2015.

[72] M. Charikar N. Ailon and A. Newman. Aggregating inconsistent information: ranking and clustering. *In 37th Symposium on Theory of Computing (STOC)*, page 684–693, 2005.

[73] A. Nara. Big data: techniques and technologies in geoinformatics. *Intenational Journal of Geographical Information Science*, 29:694–696, 2015.

[74] M Newman, M. Girvan Kira, and Larry A Rendell. Finding and evaluating community structure in networks. In *Phys. Rev E, 69:026113*, 2004.

[75] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E, 74(3)*, 2006.

[76] Nam Nguyen and Rich Caruana. Consensus clusterings. In *Proceedings of the Sixth International Conference on Data Mining(ICDM)*, pages 607–612. IEEE Computer Society, 2007.

[77] E. Dogdu O. B. Sezer and A. M. Ozbayoglu. Context-aware computing, learning, and big data in internet of things: A survey. *IEEE Internet of Things Journal*, 5:1–27, 2018.

[78] Karl Pearson. On lines and planes of closest fit to systems of points in space. In *Philosophical Magazine 2*, pages 559–572, 1901.

[79] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. In *IEEE Transactions on pattern analysis and machine intelligence*, volume 27, pages 1226–1238. IEEE Computer Society, 2005.

[80] Barbara Pes. Ensemble feature selection for high-dimensional data: a stability analysis across multiple domains. *Neural Computing and Applications*, pages 5951–5973, 2020.

[81] P. Pons and M Latapy. Computing communities in large networks using random walks. *Lecture Notes in Computer Science*, pages 284–293, 2005.

[82] Flannery B. P. Teukolsky S. A. Press, W. H. and W. T. Vetterling. Numerical recipes in c. *Cambridge University Press, Cambridge*, 1988.

[83] R.C. Prim. Shortest connection networks and some generalizations. *Bell System Technical J*, 36:1389–1401, 1957.

[84] D. Talia R. Elshawi, S. Sakr and P. Trunfio. Big data systems meet machine learning challenges: Towards big data science as a service. *Big Data Research*, 14:1–11, 2018.

[85] U.N. Raghavan, R. Albert, and S Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys Rev E 76*, 2007.

[86] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association. American Statistical Association*, 66:846–850, 1971.

[87] Holte RC. Very simple classification rules perform well on most commonly used datasets. *Journal of Machine Learning*, 11:63–91, 1993.

[88] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E, 74(1)*, 2006.

[89] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences, 105(4)*, page 1118–1123, 2008.

[90] S. El-Masri K. Kim A. Ali S. El-Sappagh, F. Ali and K. Kwak. Mobile health technologies for diabetes mellitus: Current state and future challenges. *IEEE Access*, pages 1–1, 2018.

[91] J.d.E.S. Batista Neto C. Traina-Jr S. F. da Silva, M.X. Ribeiro and A.J.M. Traina. Improving the ranking quality of medical image retrieval using a genetic feature selection method. *Decision Support Systems*, 51:810–820, 2011.

[92] P. Moradi S. Tabakhi and F. Akhlaghian. An unsupervised approach to feature selection algorithm based on ant colony optimization. *Engineering Applications of Artificial Intelligence*, 32:112–123, 2014.

[93] S.Acharya. Transductive de-noising and dimensionaity reduction using total bregman regression. pages 514–518. SIAM, 2006.

[94] Sandro Saitta, Benny Raphael, and Ian F.C. Smith. A bounded index for cluster validity. *Lecture notes in computer science*, 2007.

[95] C.E Shannon. A mathematical theory of communication. *Bell Syst. Tech. J*, page 379–423, 1948.

[96] Stjepan Picek Simone A. Ludwig and Domagoj Jakobovic. Chapter 13: Classification of cancer data: Analyzing gene expression data using a fuzzy decision tree algorithm. In *Operations Research Applications in Health Care Management, International Series in Operations Research Management Science 262*. Springer International Publishing, 2018.

[97] Ivica Slavkov, Bernard Ženko, and Sašo Džeroski. Evaluation method for feature rankings and their aggregations for biomarker discovery. *Journal of Machine Learning Research - Proceedings Track*, 8:122–135, 01 2010.

[98] Qinbao Song, Jingjie Ni, and Guangtao Wang. A fast clustering-based feature subset selection algorithm for high-dimensional data. In *IEEE Transactions on Knowledge and data engineering*, volume 25, pages 1–14. IEEE Computer Society, 2013.

[99] Alexander Strehl and Joydeep Ghosh. Cluster ensembles-a knowledge reuse framework for combing multiple partitions. *Journal of Machine Learning*, pages 583–617, 2002.

[100] Alexander Topchy, Anil K. Jain, and William Punch. A mixture model for clustering ensembles. In *Proceedings of SIAM international conference on datamining*, pages 379–390. SIAM, 2004.

[101] N. Snchez-Maro V. Boln-Canedo and A. Alonso-Betanzos. Recent advances and emerging challenges of feature selection in the context of big data. *Knowledge-Based Systems*, 86:33–45, 2015.

[102] T Varin, N Saettel, J Villain, A Lesnard, F Dauphin, R Bureau, and SJ Rault. 3d pharmacophore, hierarchical methods, and 5-ht4 receptor binding data. In *Enzyme Inhib Med Chem*, pages 593–603, 2008.

[103] Renaud Lambiotte Etienne Lefebvre Vincent D. Blondel, Jean-Loup Guillaume. Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008.

[104] C. T. Lin W. Ding and Z. Cao. Deep neuro-cognitive co-evolution for fuzzy attribute reduction by quantum leaping pso with nearest-neighbor memeplexes. *IEEE Transactions on Cybernetics*, 99:1–4, 2018.

[105] S. Chen X. Zhang W. Ding, C. T. Lin and B. Hu. Multiagent consensus-mapreduce-based attribute reduction using co-evolutionary quantum pso for big data applications. *Neurocomputing*, 272, 2017.

[106] Wei Wang and Jiong Yang. Mining high-dimensional data. In *Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook*, pages 793–799. Springer, 2005.

[107] Biao Wanga and Zhizhong Mao. A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule. *Information fusion*, 63:30–40, 2020.

[108] L. Kong Y. Liu and G. Chen. Data-oriented mobile crowdsensing: A comprehensive survey. *IEEE Communications Surveys and Tutorial*, 1:1–31, 2019.

[109] Jihoon Yang, Asok Tiyyagura, Fajun Chen, and Vasant Hanover. Feature subset selection for rule induction using ripper. In *Proceedings of Genetic and evolutionary programming*, pages 117–136, 1998.

[110] Lei Yu and Huan Liu. Feature selection for high-dimensional data:a fast correlation-based filter solution. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, 2003.

[111] X.-M. Wu Z. Li and S.-F. Chang. Segmentation using superpixels: A bipartite graph partitioning approach. *In Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, page 789–796, 2012.

[112] Xiaoli Z.Fern and Wei Lin. Cluster ensemble selection. In *SDM*, pages 128–141, 2008.

[113] E.R. Zhang. Z, Hancock. Hypergraph based information theoretic feature selection. *Pattern Recognition Letters*, 33:1991–1999, 2012.

[114] Junyang Zhao and zhili zhang. Fuzzy rough neural network and its application to feature selection. *International Journal of Fuzzy Systems*, 4:270–275, 2011.

[115] Zheng Zhao and Huan Liu. Searching for interacting features. *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, page 1156–1161, 2007.

Total Similarity = 27% - (9+6+4)% = 8%

Feature Subset Selection Based on Consensus Clustering

ORIGINALITY REPORT

29/12/21

27%	13%	26%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

T. SOBHA RANI
Associate Professor
School of Computer & Information Sciences
University of Hyderabad,
Hyderabad-500 046.

PRIMARY SOURCES

1	D Sandhya Rani, T Sobha Rani, S Durga Bhavani. "Consensus clustering for dimensionality reduction", 2014 Seventh International Conference on Contemporary Computing (IC3), 2014 Publication	9%
2	xplorestaging.ieee.org Internet Source	6%
3	D Sandhya Rani, T Sobha Rani, S Durga Bhavani. "Feature subset selection using consensus clustering", 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR), 2015 Publication	4%
4	pt.scribd.com Internet Source	<1%
5	Artificial Intelligence Foundations: Theory and Algorithms, 2015. Publication	<1%
6	link.springer.com Internet Source	<1%

This is to certify that this is from student's paper.

T. SOBHA RANI
Associate Professor
School of Computer & Information Sciences
University of Hyderabad,
Hyderabad-500 046.

This is to certify that this is from Students paper

T. SOBHA RANI
Associate Professor
School of Computer & Information Sciences
University of Hyderabad,
Hyderabad-500 046.

This is to certify that this is from Students paper.

T. SOBHA RANI
Associate Professor
School of Computer & Information Sciences
University of Hyderabad,
Hyderabad-500 046.

7	Linghe Kong, Wenhao Qu, Jiadi Yu, Hua Zuo, Guihai Chen, Fei Xiong, Shirui Pan, Siyu Lin, Meikang Qiu. "Distributed Feature Selection for Big Data using Fuzzy Rough Sets", IEEE Transactions on Fuzzy Systems, 2019 Publication	<1 %
8	www.inderscience.com Internet Source	<1 %
9	en.wikipedia.org Internet Source	<1 %
10	"Simulated Evolution and Learning", Springer Science and Business Media LLC, 2014 Publication	<1 %
11	Parham Moradi, Mozhgan Gholampour. "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy", Applied Soft Computing, 2016 Publication	<1 %
12	eprints.nottingham.ac.uk Internet Source	<1 %
13	rdrr.io Internet Source	<1 %
14	"Feature Extraction, Construction and Selection", Springer Science and Business Media LLC, 1998 Publication	<1 %

15	Submitted to Pontificia Universidad Catolica del Peru Student Paper	<1%
16	Jaehong Yu, Hua Zhong, Seoung Bum Kim. "An Ensemble Feature Ranking Algorithm for Clustering Analysis", Journal of Classification, 2019 Publication	<1%
17	Submitted to University of Bristol Student Paper	<1%
18	Yi Hong, Sam Kwong, Yuchou Chang, Qingsheng Ren. "Consensus unsupervised feature ranking from multiple views", Pattern Recognition Letters, 2008 Publication	<1%
19	www.tandfonline.com Internet Source	<1%
20	"Cloud Computing and Security", Springer Science and Business Media LLC, 2016 Publication	<1%
21	"Emerging Technologies in Data Mining and Information Security", Springer Science and Business Media LLC, 2019 Publication	<1%
22	Lecture Notes in Computer Science, 2004. Publication	<1%

23	"Exploratory Data Analysis in Empirical Research", Springer Science and Business Media LLC, 2003 Publication	<1 %
24	"Image Analysis and Recognition", Springer Nature, 2004 Publication	<1 %
25	Lecture Notes in Computer Science, 2005. Publication	<1 %
26	Submitted to Queen Mary and Westfield College Student Paper	<1 %
27	Verónica Bolón-Canedo, Amparo Alonso-Betanzos. "Recent Advances in Ensembles for Feature Selection", Springer Science and Business Media LLC, 2018 Publication	<1 %
28	"Computer Engineering and Networking", Springer Science and Business Media LLC, 2014 Publication	<1 %
29	Dalton Ndirangu, Waweru Mwangi, Lawrence Nderu. "An Ensemble Filter Feature Selection Method and Outlier Detection Method for Multiclass Classification", Proceedings of the 2019 8th International Conference on Software and Computer Applications, 2019	<1 %

Publication

30	Danial Hooshyar, Yeongwook Yang, Margus Pedaste, Yueh-Min Huang. "Clustering Algorithms in an Educational Context: An Automatic Comparative Approach", IEEE Access, 2020 Publication	<1 %
31	Hongbin Dong, Tao Li, Rui Ding, Jing Sun. "A novel hybrid genetic algorithm with granular information for feature selection and optimization", Applied Soft Computing, 2018 Publication	<1 %
32	Lecture Notes in Computer Science, 2016. Publication	<1 %
33	Submitted to Universiti Teknologi Malaysia Student Paper	<1 %
34	Wilker Altidor. "Ensemble Feature Ranking Methods for Data Intensive Computing Applications", Handbook of Data Intensive Computing, 2011 Publication	<1 %
35	research.gold.ac.uk Internet Source	<1 %
36	"ICDSMLA 2019", Springer Science and Business Media LLC, 2020 Publication	<1 %

37	"Recent Trends in Image Processing and Pattern Recognition", Springer Science and Business Media LLC, 2017 Publication	<1 %
38	Linghe Kong, Wenhao Qu, Jiadi Yu, Hua Zuo, Guihai Chen, Fei Xiong, Shirui Pan, Siyu Lin, Meikang Qiu. "Distributed Feature Selection for Big Data Using Fuzzy Rough Sets", IEEE Transactions on Fuzzy Systems, 2020 Publication	<1 %
39	Submitted to Unizin, LLC Student Paper	<1 %
40	orca.cf.ac.uk Internet Source	<1 %
41	www.tutorialspoint.com Internet Source	<1 %
42	Lianxi Wang, Shengyi Jiang, Siyu Jiang. "A feature selection method via analysis of relevance, redundancy, and interaction", Expert Systems with Applications, 2021 Publication	<1 %
43	Majdi Mafarja, Ibrahim Aljarah, Hossam Faris, Abdelaziz I. Hammouri, Ala' M. Al-Zoubi, Seyedali Mirjalili. "Binary Grasshopper Optimisation Algorithm Approaches for Feature Selection Problems", Expert Systems with Applications, 2018	<1 %

Publication

44	towardsdatascience.com Internet Source	<1%
45	"Rough Sets", Springer Science and Business Media LLC, 2018 Publication	<1%
46	open.library.ubc.ca Internet Source	<1%
47	tudr.thapar.edu:8080 Internet Source	<1%
48	Gema Bello-Orgaz, Julio Hernandez-Castro, David Camacho. "Detecting discussion communities on vaccination in twitter", Future Generation Computer Systems, 2017 Publication	<1%
49	"Advances in Knowledge Discovery and Data Mining", Springer Science and Business Media LLC, 2018 Publication	<1%
50	Huan Liu. "Efficiently handling feature redundancy in high-dimensional data", Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 03 KDD 03, 2003 Publication	<1%

51	Submitted to Imperial College of Science, Technology and Medicine Student Paper	<1%
52	Maoguo Gong, Qing Cai, Lijia Ma, Shanfeng Wang, Yu Lei. "Computational Intelligence for Network Structure Analytics", Springer Science and Business Media LLC, 2017 Publication	<1%
53	Moradi, Parham, and Mehrdad Rostami. "Integration of graph clustering with ant colony optimization for feature selection", Knowledge-Based Systems, 2015. Publication	<1%
54	Submitted to iGroup Student Paper	<1%
55	"Pattern Recognition and Machine Intelligence", Springer Science and Business Media LLC, 2015 Publication	<1%
56	Chuncheng Zhang, Sutao Song, Xiaotong Wen, Li Yao, Zhiying Long. "Improved sparse decomposition based on a smoothed L0 norm using a Laplacian kernel to select features from fMRI data", Journal of Neuroscience Methods, 2015 Publication	<1%

57	Jose M. Sallan, Oriol Lordan. "Complex Networks Theory", Elsevier BV, 2020 Publication	<1 %
58	S. Senthamarai Kannan, N. Ramaraj. "A novel hybrid feature selection via Symmetrical Uncertainty ranking based local memetic search algorithm", Knowledge-Based Systems, 2010 Publication	<1 %
59	Jian Ma, Jianping Fan, Feng Liu, Honghui Li. "A Community Detection Algorithm Based on Markov Random Walks Ants in Complex Network", Journal of Shanghai Jiaotong University (Science), 2019 Publication	<1 %
60	Submitted to National University of Ireland, Galway Student Paper	<1 %
61	Raymond, J.W.. "Comparison of chemical clustering methods using graph- and fingerprint-based similarity measures", Journal of Molecular Graphics and Modelling, 200303 Publication	<1 %
62	Recep Doga Siyli, Batuhan Gundogdu, Murat Saraclar, Lale Akarun. "Unsupervised Key Hand Shape Discovery of Sign Language	<1 %

Videos with Correspondence Sparse Autoencoders", ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020

Publication

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		