

OPTIMIZING STORAGE SPACE FOR HIGHER DIMENSIONAL DATA USING FEATURE SUBSET SELECTION APPROACH

A PROJECT REPORT

submitted by

DONIA AUGUSTINE
TJE15CSCE03

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Technology
in
Computer Science and Engineering



Department of Computer Science and Engineering

Thejus Engineering College
Vellarakkad, Thrissur

MAY 2017

DECLARATION

I undersigned hereby declare that the project report on "Optimizing Storage Space for Higher Dimensional Data using Feature Subset Selection Approach", submitted for partial fulfillment of the requirements for the award of degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Ms. Panchami V.U., Assistant Professor, Department of Computer Science and Engineering, Thejus Engineering College. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Vellarakkad

12/05/2017

DONIA AUGUSTINE

TJE15CSCE03

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
THEJUS ENGINEERING COLLEGE, VELLARAKKAD**



CERTIFICATE

This is to certify that the report entitled '**Optimizing Storage Space for Higher Dimensional Data using Feature Subset Selection Approach**' submitted by **Donia Augustine** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Technology in Computer Science and Engineering is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

INTERNAL SUPERVISOR

EXTERNAL SUPERVISOR

PG COORDINATOR

Mr. Valanto Alappatt

Assistant Professor

Dept. of CSE

Thejus Engineering College

HEAD OF THE DEPARTMENT

Dr. Saju P. John

Associate Professor, HOD

Dept. of CSE

Thejus Engineering College

ACKNOWLEDGEMENT

It is with great enthusiasm and the learning spirit that I am bringing out this project report. I also feel that it is the right opportunity to acknowledge the support and guidance that come in from various quarters during the course of completion of my project.

I have great pleasure in expressing my gratitude and obligations to **Ms. Panchami V.U.**, Assistant Professor, Department of Computer Science and Engineering, Thejus Engineering College, for her valuable guidance, constant encouragement and creative suggestions to make this work a great success.

I express my sincere thanks to **Dr. Saju P. John**, Associate Professor, Head of the Department, Department of Computer Science and Engineering, Thejus Engineering College, for his encouragement and support.

I extend my heartiest thanks to project coordinator **Mr. Valanto Alappatt**, Assistant Professor, Department of Computer Science and Engineering, Thejus Engineering College for his valuable help.

I also acknowledge my gratitude to other members of faculty in the Department of Computer Science and Engineering and all my friends for their whole hearted cooperation and encouragement.

Vellarakkad

DONIA AUGUSTINE

ABSTRACT

As applications producing data of higher dimensions has increased tremendously, clustering of data under reduced memory became a necessity. Feature selection is a typical approach to cluster higher dimensional data. It involves identifying a subset of most relevant features from the entire set of features. Our approach suggests a method to efficiently cluster higher dimensional data under reduced memory. An N-dimensional feature selection algorithm, NDFS is used for identifying the subset of relevant features. The concept of feature selection helps in removing the irrelevant and redundant features from each cluster. In the initial phase of NDFS algorithm, features are divided into clusters using graph-theoretic clustering methods. The final phase of the algorithm generates the subset of relevant features that are closely related to the target class. Features in different clusters are relatively independent. The clustering-based strategy of NDFS have a high probability of producing a subset of useful and independent features. Further, to efficiently handle the generated subset of features, a minimum spanning tree is constructed. We discuss the space and time complexity of maintaining the minimum spanning tree. Experiments on a wide range of synthetic and real data sets highlight that the NDFS approach manipulates higher dimensional data efficiently over other methods.

CONTENTS

ACKNOWLEDGEMENT	i
ABSTRACT	ii
LIST OF TABLES	vi
LIST OF FIGURES	vii
ABBREVIATIONS	viii
CHAPTER 1 INTRODUCTION	1
1.1 GENERAL BACKGROUND	1
1.1.1 Higher Dimensional Data	1
1.1.2 Feature Selection	3
1.2 OBJECTIVES	5
1.3 SCOPE OF FEATURE SELECTION	6
1.4 OUTLINE OF THE THESIS	7
CHAPTER 2 LITERATURE SURVEY	9
2.1 FEATURE SELECTION BASED ON COMBINING THE FEAT- TURES FOR EVALUATION	9
2.1.1 Feature subset-based methods	9
2.1.2 Feature ranking-based methods	12
2.2 FEATURE SELECTION BASED ON THE SUPERVISED LEARN- ING ALGORITHM USED	13
2.2.1 Wrapper-based methods	13

2.2.2	Embedded-based methods	14
2.2.3	Filter-based methods	15
2.2.4	Hybrid methods	17
CHAPTER 3	EXISTING SYSTEM	19
3.1	DBSTREAM ONLINE COMPONENT	19
3.1.1	Leader-Based Clustering	19
3.1.2	Competitive Learning	20
3.1.3	Capturing Shared Density	21
3.1.4	Fading and Forgetting Data	23
3.2	SHARED DENSITY RECLUSTERING	24
3.3	DRAWBACKS	25
CHAPTER 4	PROPOSED SYSTEM	26
4.1	PROBLEM DEFINITION	26
4.2	SYSTEM ARCHITECTURE	26
4.3	MODULE DESCRIPTION	27
4.4	ALGORITHM	28
4.5	ADVANTAGES	31
CHAPTER 5	SYSTEM SPECIFICATIONS	32
5.1	HARDWARE REQUIREMENTS	32
5.2	SOFTWARE REQUIREMENTS	32
5.3	SOFTWARE ENVIRONMENT	32
5.3.1	NetBeans IDE	32
5.3.2	Front End-Java EE	34
5.3.3	Back End :- MySQL	37

CHAPTER 6	RESULTS AND DISCUSSIONS	40
6.1	EXPERIMENTAL RESULTS	40
6.2	PERFORMANCE EVALUATION	46
CHAPTER 7	CONCLUSION	49
REFERENCES		50
LIST OF PUBLICATIONS		56

LIST OF TABLES

6.1 Comparison on Time and Space Utilization 48

6.2 Time and Space Complexity Analysis 48

LIST OF FIGURES

1.1	Different feature selection techniques	4
3.1	High and Low density MC regions	20
3.2	DBSTREAM Algorithm	22
3.3	Clean-up Algorithm	24
3.4	Reclustering Algorithm	25
4.1	System Architecture	27
4.2	Proposed System Framework	28
6.1	Loading the Dataset	40
6.2	Cluster Formation	41
6.3	Estimation of Standard Deviation	41
6.4	Density Estimation	42
6.5	Measuring Conditional Entropy	42
6.6	Individual Feature Relevance Measure	43
6.7	T-Relevance Estimation	43
6.8	Removal of Irrelevant Features and correlation Estimation	44
6.9	Minimum Spanning Tree Construction	44
6.10	Eliminating Redundant Features	45
6.11	Final Feature Subset Generation	45
6.12	Proportion of Features Selected	46
6.13	Analysis of Data handled	47
6.14	Average number of edges per cluster	47

ABBREVIATIONS

NDFS	N Dimensional F eature S election
COFS	Consistency based F eature S ubset S election
ACO	A nt C olony O ptimization
PSO	P article S warm O ptimization
MRMR	M aximum R elevancy M inimum R edundancy
CIFE	C onditional I nformax F eature E xtraction

CHAPTER 1

INTRODUCTION

Data mining, the extraction of hidden predictive information from large database, is a powerful technology to assist companies to focus on the most relevant information in their data warehouses. Data mining incorporated many techniques such as machine learning, pattern recognition, database and data warehouse systems, visualization, high performance computing, and many application domains. With the rapid growth of computational biology and ecommerce applications, high dimensional data becomes very common. The mining of high dimensional data is an urgent problem in day today life.

1.1 GENERAL BACKGROUND

1.1.1 Higher Dimensional Data

High-dimensional data, i.e., data described by a large number of attributes, pose specific challenges to clustering. The general increase in complexity of various computational problems as dimensionality increases, is known to render traditional clustering algorithms ineffective. The curse of dimensionality, means that with increasing number of dimensions, a loss of meaningful differentiation between similar and dissimilar objects is observed. As high-dimensional objects appear almost alike, new approaches for clustering are required. Consequently, recent research has focused on developing techniques and clustering algorithms specifically for high-dimensional data. Still, open research issues remain.

Clustering is a data mining task devoted to the automatic grouping of data based on mutual similarity. Each cluster groups objects that are similar to one another, whereas dissimilar objects are assigned to different clusters, possibly separating out noise. In this manner, clusters describe the data structure in an unsupervised manner, i.e., without the need for class labels. A number of clustering paradigms exist that provide different cluster models and different algorithmic

approaches for cluster detection. Common to all approaches is the fact that they require some underlying assessment of similarity between data objects.

The technologies present investigators with the task of extracting meaningful statistical and biological information from high dimensional data. A great deal of data from different domains such as medicine, business, science is high dimensional. Many objects can be represented under high dimensions such as speech signals, images, videos, text documents, hand writing letters and numbers. We often need to analyze large amount of data and efficiently process them. For e.g. need to identify person fingerprints, certain hidden patterns and images, to trace objects from videos. To complete these tasks, we develop the systems to process data suitably. However due to high dimension of data, direct processing of these data may be very complicated and unstable so that it is infeasible.

Challenges in Higher Dimensional Data

- As the number of possible values with each dimension grows exponentially, complete enumeration of all subspaces becomes intractable. This problem is known as the curse of dimensionality.
- As the dimensionality grows, the measure of distance becomes imprecise, since the distance between any two points in a given dataset converges. The discrimination of the nearest and farthest point in particular becomes meaningless.
- A cluster groups objects that are related, based on their attribute values. However, given a large number of attributes, some of the attributes will usually not be meaningful for a given cluster. This is termed as the local feature relevance problem. A global filtering of attributes is not sufficient as different clusters might be found in different subspaces.
- Given a large number of attributes, it is likely that some attributes are correlated. Hence, clusters might exist in arbitrarily oriented relative subspaces.

With the higher dimensions, machine learning methods have difficulty in dealing with the increased number of input features, which is posing a great deal of challenge for researchers. In order to make use of machine learning methods effective, preprocessing of the data is essential.

Feature selection is one of the most frequent and important techniques in data preprocessing, and has become an indispensable component of the machine learning process.

1.1.2 Feature Selection

In machine learning studies, feature selection is also known as variable selection, attribute selection or variable subset selection. It is the process of detecting relevant features and removing irrelevant, redundant or noisy data. Irrelevant features are those that provide no useful information, and redundant features provide no more information than the currently selected features. In terms of supervised learning, feature selection gives a set of candidate features with the best commitment among size and evaluation measure.

Feature selection algorithms improves learning, either in term of generalization capability, learning speed, or reducing the complexity of the induced model. In the process of feature selection, irrelevant and redundant features or noise in the data may be hinder in many situations, because they are not relevant and important with respect to the class concept. Machine learning methods gets particularly difficult when the number of samples is much less than the features, because the search space will be sparsely populated. Therefore, the model will find it difficult to differentiate between noise and relevant data.

Figure 1.1 shows the two main models that deal with feature selection: (a) filter methods, and (b) wrapper methods. Filter methods rely on the general characteristics of the training data to select features with independence of any learning classifier, which are usually computationally less expensive than the wrapper models, and have the ability to scale to large datasets. On the other side, wrapper methods involve optimising a learning classifier as part of the feature selection process. Wrapper models tend to give better results and the model is more precise than the filter model. However, wrapper models are very time consuming, which restricts application with some datasets.

The hybrid methods are based on a sequential approach where the first step is usually based on filter methods to reduce the number of features considered in the second stage. Afterwards, a wrapper method is employed to select the desired number of features using this reduced set in the second stage.

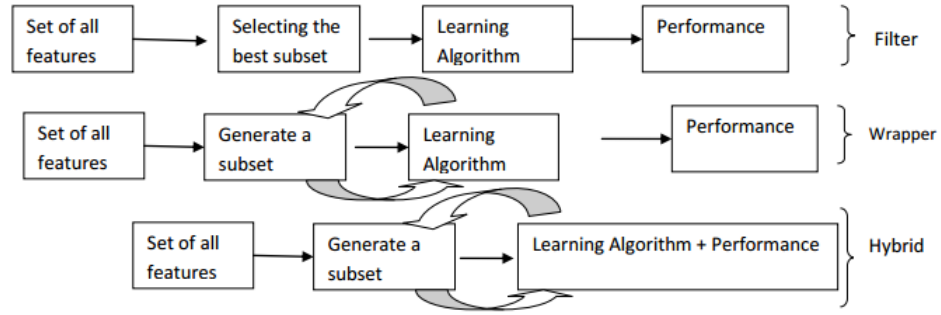


Figure 1.1: Different feature selection techniques

Structure of learning system:

From the perspective of label availability, feature selection methods can be broadly classified into supervised, unsupervised, and semi-supervised methods. In terms of different selection strategies, feature selection can be categorized as filter, wrapper, and embedded models.

Supervised feature selection is usually used for classification tasks. The availability of the class labels allows supervised feature selection algorithms to effectively select discriminative features to distinguish samples from different classes. Features are first generated from training data. Instead of using all the data to train the supervised learning model, supervised feature selection will first select a subset of features and then process the data with the selected features to the learning model. The feature selection phase will use the label information and the characteristics of the data, such as information gain or Gini index, to select relevant features. The final selected features, as well as with the label information, are used to train a classifier, which can be used for prediction.

Unsupervised feature selection is usually used for clustering tasks. The unsupervised feature selection is almost to supervised feature selection, except that there is no label information involved in the feature selection phase and the model learning phase. Without label information to define feature relevance, unsupervised feature selection relies on another alternative criterion during the feature selection phase. One commonly used criterion chooses features that can best preserve the manifold structure of the original data. Another frequently used method is to seek cluster indicators through clustering algorithms and then transform the unsupervised feature se-

lection into a supervised framework. There are two different ways to use this method. One way is to seek cluster indicators and simultaneously perform the supervised feature selection within one unified framework. The other way is to first seek cluster indicators, then to perform feature selection to remove or select certain features, and finally to repeat these two steps iteratively until certain criterion is met. In addition, certain supervised feature selection criterion can still be used with some modification.

Semi-supervised feature selection is usually used when a small portion of the data is labeled. When such data is given to perform feature selection, both supervised and unsupervised feature selection might not be the best choice. Supervised feature selection might not be able to select relevant features because the labeled data is insufficient to represent the distribution of the features. Unsupervised feature selection will not use the label information, while label information can give some discriminative information to select relevant features. Semi-supervised feature selection, which takes advantage of both labeled data and unlabeled data, is a better choice to handle partially labeled data. The general framework of semi-supervised feature selection is the same as that of supervised feature selection, except that data is partially labeled. Most of the existing semi-supervised feature selection algorithms rely on the construction of the similarity matrix and select features that best fit the similarity matrix. Both the label information and the similarity measure of the labeled and unlabeled data are used to construct the similarity matrix so that label information can provide discriminative information to select relevant features, while unlabeled data provide complementary information

1.2 OBJECTIVES

The project entitled "Optimizing storage space for higher dimensional data using feature subset selection approach" implements a method to achieve the following:

- Removal of irrelevant features
- Handling higher dimensions
- Reduce the memory requirement
- Reduce complexity

1.3 SCOPE OF FEATURE SELECTION

High-dimensional data is very ubiquitous in the real world, which makes feature selection a very popular and practical preprocessing technique for various real-world applications, such as text categorization, remote sensing, image retrieval, microarray analysis, mass spectrum analysis, sequence analysis, and so on.

Text Categorization

Text Clustering The task of text clustering is to group similar documents together. In text clustering, a text or document is always represented as a bag of words, which causes high-dimensional feature space and sparse representation. Obviously, a single document has a sparse vector over the set of all terms. The performance of clustering algorithms degrades dramatically due to high dimensionality and data sparseness. Therefore, in practice, feature selection is a very important step to reduce the feature space in text clustering.

Remote Sensing

Feature selection is one of the important tasks in the remote sensing image classification. Various challenges and issues in feature selection and hyper spectral remote sensing image analyzed. Recently pre-processing techniques have been proposed for hyper spectral images in which feature extraction and feature selection have been emphasized as important components in hyper spectral image classification. Feature selection guided by evolutionary algorithms has been proposed, and use a self-adaptive differential evolution for feature subset generation. Spatial and spectral informations are utilized to select a subset of bands from a hyper spectral image to improve the performance of the classification.

Intrusion Detection

In this modern age, information sharing, distribution, or communication is widely done by network-based computer systems. Therefore, the security of the system is an important issue protecting communication networks from intrusion by enemies and criminals. One of the ways to protect communication networks (computer systems) is intrusion detection. Feature selection plays an important role to classifying system activity as legitimate or an intrusion.

Genomic Microarray Data

Microarray data is usually short and fat data with high dimensionality having a small sample size, which poses a great challenge for computational techniques. Their dimensionality can be up to tens of thousands of genes, while their sample sizes can only be several hundreds. Furthermore, additional experimental complications like noise and variability render the analysis of microarray data an exciting domain. Because of these issues, various feature selection algorithms are adopted to reduce the dimensionality and remove noise in microarray data analysis.

Hyperspectral Image Classification

Hyperspectral sensors record the reflectance from the Earth's surface over the full range of solar wavelengths with high spectral resolution, which results in high-dimensional data that contains rich information for a wide range of applications. However, this high-dimensional data contains many irrelevant, noisy, and redundant features that are not important, useful, or desirable for specific tasks. Feature selection is a critical preprocessing step to reduce computational cost for hyperspectral data classification by selecting relevant features.

Sequence Analysis

In bioinformatics, sequence analysis is a very important process to understand a sequence's features, functions, structure, or evolution. In addition to basic features that represent nucleotide or amino acids at each position in a sequence, many other features, such as k-mer patterns, can be derived. By varying the pattern length k , the number of features grows exponentially. However, many of these features are irrelevant or redundant; thus, feature selection techniques are applied to select a relevant feature subset and essential for sequence analysis.

1.4 OUTLINE OF THE THESIS

The report is organized as follows: The second chapter, 'Literature Survey', presents a critical appraisal of the previous works published in the literature pertaining to the topic of the investigation. The third chapter, 'Existing System', provides a brief explanation about the clustering algorithm applicable to moderate datatypes. The fourth chapter, 'Proposed System', specifies

the features of the thesis. The fourth chapter gives the problem definition, detailed module wise description, a new system architecture model as a solution to the encountered problem. The fifth chapter, 'System Specifications', gives the hardware and software requirements. The sixth chapter, 'Results and Discussions', gives a thorough evaluation of the investigation carried out and brings out the contributions of the thesis. It also gives the current status of the work and displays the available results. Chapter six concludes the thesis and 'References', gives the list of publications and papers that contributed to this thesis.

CHAPTER 2

LITERATURE SURVEY

As the feature selection is employed in various machine learning applications, it has remarkable literature records made by the research community. Feature selection is a preprocessing technique to select the significant features from a dataset by removing the irrelevant and redundant features for improving the performance of the machine learning algorithms. The feature selection process can be categorized into various methods based on how the features are combined for evaluation in the feature selection process and how the supervised learning algorithm is used to evaluate the features in the features selection process. This chapter reviews the literature on various features selection methods.

2.1 FEATURE SELECTION BASED ON COMBINING THE FEATURES FOR EVALUATION

This section reviews various methods of feature selection based on how the features are combined for evaluation in order to select the significant features from a dataset. They are classified into feature subset-based and feature ranking-based methods.

2.1.1 Feature subset-based methods

In the feature subset-based method, the features are combined as possible combinations of feature subsets using any one of the searching strategies. Then, the feature subsets are evaluated using any one of the statistical measures or the supervised learning algorithms to observe the significance of each subset and the most significant subset is selected as the significant feature subset for a given dataset. If the subset is evaluated using the supervised learning algorithm, then this method is known as wrapper method.

The best example for the feature subset-based method is correlation-based feature subset selection (CRFS) developed by Hall. In this approach, two correlation measures are considered; one is feature-class correlation and another one is feature-feature correlation. Initially, N numbers of features are combined as possible combinations of feature subsets using heuristic-based best-first search, then each subset is evaluated with the two correlation measures as mentioned above. The subset that has lesser feature-feature correlation and higher feature-class correlation compared to other feature subsets is considered as the selected significant feature subset for the classification task.

Liu Setiono [5] proposed a feature subset-based feature selection method namely consistency based feature subset selection (COFS). This method uses the class consistency as an evaluation metric in order to select the significant feature subset from the given dataset. These methods are the filter-based methods since they do not use the supervised learning algorithm to validate the subsets and they use the statistical measure for evaluating the feature subsets. In general, the exhaustive or complete search has to generate 2^N number of subsets to produce the maximum number of possible combinations of feature subsets from the N number of features for evaluation. Therefore, this exhaustive searching strategy is computationally quite expensive hence the heuristic searching strategies such as simulated annealing (SA), tabu searching (TS), ant colony optimization (ACO), genetic algorithm (GA), particle swarm optimization (PSO), etc are used by some of the researchers to get the optimal solution by generating less number of feature subsets for evaluation. In the heuristic searching, the heuristic function obtains the prior knowledge to guide the search process to generate the subsets and these subsets are evaluated using supervised machine learning algorithm. These factors make the feature subset-based methods computationally expensive and also these methods seem to be the wrapper approach.

Some researchers used the simulated annealing search for generating the feature subset for evaluations. For example, Lin et al used the simulated annealing search to generate the feature subsets and evaluated them by supervised learning algorithm namely back-propagation network (BPN) to choose the better feature subset [6].

In several feature selection methods, the tabu search is used for subset generation such as Zhang Sun developed a tabu search-based feature selection. In this method, the subsets generated by tabu search are evaluated using the classification error criteria to find the better

feature subset [7]. Tahir et al formed the feature subsets using tabu search then these subsets are evaluated using K-nearest neighbor classifier (kNN) with the classification error as evaluation criteria to obtain the significant feature subset [8].

Aghdam et al employed the ant colony optimization search to form the feature subsets and they are validated by the nearest neighbor classifier for text classification application [9]. Sivagaminathan Ramakrishnan developed an ant colony optimization-based feature selection with artificial neural networks (ANN) for medical diagnosis system. In this method, the generated feature subsets are validated using ANN [10]. Sreeja Sankar presented an ant colony optimization-based feature selection with instancebased pattern matching-based classification (PMC) [11].

In certain feature selection research works, the genetic algorithm is adopted to generate the feature subsets for evaluation and the supervised machine learning algorithm is used to evaluate the generated subsets. Welikala et al presented a feature selection using genetic algorithm with support vector machine (SVM) for mining the medical dataset [12]. Erguzel et al used the genetic algorithm and artificial neural network for electroencephalogram (EEG) signal classification [13]. Oreski proposed a feature selection method based on genetic algorithm with neural networks for credit risk assessment [14]. Li et al developed a genetic algorithm with support vector machine for hyperspectral image classification [15]. Das et al formulated a genetic algorithm with support vector machine-based feature selection for handwritten digit recognition application [16]. Wang et al applied the genetic algorithm for subset generation with support vector machine in feature selection process for data classification applications [17].

In the literature, some researches employed the particle swarm optimization to generate the feature subsets and to validate them by supervised machine learning algorithm to identify the significant feature subset. Chen et al presented a feature selection method using particle swarm optimization search for sleep disorder diagnosis system [18]. Yang et al developed a particle swarm optimization-based feature selection for land cover classification [19].

From the subset-based feature selection literature, it is observed that the exhaustive or complete search leads to high computational complexity as it generates 2^N number of subsets from N number of features for evaluation. This searching strategy cannot be a better choice for high-dimensional space. The heuristic search methods also lead to more computational complexity,

because they need prior knowledge and each generated subset needs to develop a classification model for evaluating them to obtain the optimal feature subset in an iterative manner, hence these searching strategies are not suitable for high-dimensional space. However, these heuristic search methods follow a wrapper-based approach. Therefore, these methods are computationally expensive and they can only produce higher classification accuracy for the specific classification algorithm used to validate the subset, so they cannot achieve high generality.

2.1.2 Feature ranking-based methods

In the feature-ranking based approach, each feature of a dataset is weighted based on any one of the statistical or information-theoretic measures and the features are ranked based on their weight. Then the higher ranked features are selected as the significant features using a pre-defined threshold that determines the number of features to be selected from a dataset. The best example for the feature ranking-based method is chi-square-based feature selection (CQFS). In this method, Liu Setiono used the chi-square statistic measure to weight the features in order to rank them for selecting the significant features. In the similar way, the information-theoretic measures such as information gain, symmetric uncertainty, gain ratio, etc. are employed to weight the individual feature and rank them for selection.

Further, it is observed that the feature ranking-based methods use the statistical measures or information-theoretic measures to weight the individual feature only by observing the relevancy between the individual feature and the target-class. Hence, these methods take less runtime but fail to remove the redundant features. The feature ranking-based methods follow a filter-based approach since these methods do not involve the supervised learning algorithm to evaluate the significance of the features. Consequently, these methods are independent of the supervised learning algorithm hence they achieve more generality and less computational complexity. Thus, the feature ranking-based methods can be a good choice for selecting the significant features from the highdimensional space with suitable redundancy analysis mechanism.

2.2 FEATURE SELECTION BASED ON THE SUPERVISED LEARNING

ALGORITHM USED

This section reviews various methods of feature selection based on the machine learning algorithm used. They are categorized as wrapper, embedded, filter, and hybrid methods.

2.2.1 Wrapper-based methods

Wrapper-based approach generates the feature subsets using any one of the searching techniques and evaluates these subsets using the supervised learning algorithm in terms of classification error or accuracy. The wrapper method seems to be a "brute force" method. Kohavi John developed a wrapperbased feature selection method for selecting the significant features from the dataset [20]. This method consists of search engine for subset generation and classification algorithm to evaluate the subset. Further, they compare the performance of this method in terms of classification accuracy with hillclimbing and best-first searching strategies using decision tree and naive Bayes classifiers. However, they observed that wrapper method has the problems such as searching overhead, overfitting, and increased runtime.

In wrapper approach, the searching is an overhead since the searching technique does not have the domain knowledge. In order to overcome the searching time overhead, Inza et al used estimation of Bayesian network algorithm for feature subset selection using naive Bayes and ID3 (Iterative Dichotomiser 3) [21]. Dy Brodley developed a wrapperbased approach for unsupervised learning using order identification (recognizing the number of clusters in the data) with the expectation maximization (EM) clustering algorithm using maximum likelihood (ML) criterion [22]. Aha Bankert presented a wrapper-based method with beam search and IB1 classifier. Also, they compared its performance with the well known sequential search algorithms for feature selection such as forward sequential selection (FSS) and backward sequential selection (BSS). They observed that the beam search outperforms the FSS and BSS.

The Maldonado Weber developed a wrapper approachbased feature selection by combining support vector machine (SVM) with kernel functions. This method uses the sequential backward selection for feature subset generation and these subsets are validated in terms of classification error to identify the best subset [23]. In order to minimize the searching overhead,

GÃijtlein et al used the search algorithm namely ORDERED-FS that orders the features in terms of resubstitution error to identify their irrelevancy [24]. Kabir et al developed a wrapper-based constructive approach for feature selection (CAFS) using neural network (NN). In this method, the correlation measure is used to remove the redundancy in the searching strategy for improving the performance of NN [25]. Stein et al proposed an ant colony optimization-based feature selection with wrapper model. In this approach, the ant colony optimization is used as a searching method in order to reduce the searching overhead such as blind search or forward selection or backward elimination searching methods [26]. Furthermore, to minimize the searching overhead, Zhuo et al presented a wrapper-based feature selection using genetic algorithm with support vector machine for classifying the hyper-spectral images [27].

In the wrapper approach, overfitting can be overcome by postpruning, jitter, and early stopping methods. Post-pruning is carried out while developing the decision tree. In jitter method, the noisy data that make the learning process more difficult are eliminated in order to fit the training data thereby the overfitting is eliminated. In early stopping method, overfitting is eliminated using neural network by stopping the training process when performance on a validation set starts to deteriorate. The researchers have tried to reduce the overfitting by early stopping method using genetic algorithm based searching with early stopping (GAWES) [28].

Further, it is observed that the wrapper-based methods are suffered by the searching overhead, overfitting and have more computational complexity with less generality since they use the supervised learning algorithm for evaluating the generated subsets by the searching method. Therefore, these methods are not suitable choice for the high-dimensional space.

2.2.2 Embedded-based methods

The embedded-based methods use a part of the learning process of the supervised learning algorithm for feature selection. Embedded-based methods reduce the computational cost than the wrapper method. This embedded method can be roughly categorized into three namely pruning method, built-in mechanism, and regularization models. In the pruning-based method, initially all the features are taken into the training process for building the classification model and the features which have less correlation coefficient value are removed recursively using the support vector machine (SVM). In the built-in mechanism-based feature selection method, a

part of the training phase of the C4.5 and ID3 supervised learning algorithms are used to select the features. In the regularization method, fitting errors are minimized using the objective functions and the features with near zero regression coefficients are eliminated.

Neumann et al developed an embedded-based feature selection method for selecting the significant features from synthetic and real world datasets. In their approach, linear and non linear SVMs are employed in the selection process using the deference of convex functions algorithm (DCA) [29]. Xiao et al proposed an embedded-based method to select the significant features from audio signals for emotion classification. This method was implemented based on the principle of evidence theory with mass function and the identified most relevant features are added incrementally for classification. Maldonado et al developed an embedded method to select the significant features from imbalanced data for classification with several objective functions [30].

Further, it is observed that the embedded methods are computationally efficient than the wrapper methods and computationally costlier than the filter methods hence they cannot be suitable choice for high-dimensional space and they have poor generality since the embedded methods use the supervised learning algorithm.

2.2.3 Filter-based methods

The filter-based approaches are independent of the supervised learning algorithm therefore offer more generality and they are computationally cheaper than the wrapper and embedded approaches. For processing the high-dimensional data, the filter methods are suitable rather than the wrapper and embedded methods.

Generally, the process of feature selection aimed at choosing the relevant features. The best example is Relief [31] that was developed with the distance-based metric function that weights each feature based on their relevancy (correlation) with the target-class. However, Relief is ineffective as it can handle only the two-class problems and also does not deal with redundant features. The modified version of the Relief known as ReliefF [32] can handle the multi-class problems and deal with incomplete and noisy datasets too. However, it fails to remove the redundant features. Holte developed a rule based attribute selection known as OneR which forms one rule for each feature and selects the rule with the smallest error [33]. Yang Moody pro-

posed a joint mutual information-based approach (JMI) for classification. It calculates the joint mutual information between the individual feature and the target-class to identify the relevant features, and a heuristic search is adopted for optimization when the number of features is more. The features containing similar information and lesser relevancy to the target-class are treated as redundant features that are to be eliminated [34].

Lei Yu in their work proposed a feature selection algorithm which is specially used for high dimensional data which is called as fast correlation base filter. This algorithm is for removing irrelevant and redundant data. They applied FCBF, ReliefF, CorrF, and ConSF on four datasets and recorded the running time and number of features selected. Then they applied C4.5 and NBC classification on the data. Peng et al proposed a mutual information-based maxrelevancy min-redundancy (MRMR) feature selection. To identify the feature relevancy, the mutual information is computed between the individual feature and target-class, and to identify the redundant feature, the mutually exclusive condition is applied [35]. Battiti developed a mutual information-based feature selection method (MIFS). In this method, mutual information measure is used to determine the relevancy between the individual feature and the target-class. The features having similar information are considered as redundant features that are to be removed [36]. Fleuret presented a feature selection scheme namely conditional mutual info maximization (CMIM) that recursively chooses the features that have maximum mutual information with the target-class for classification [37].

Meyer Bontempi proposed a filter-based approach that uses double input symmetrical relevance (DISR) metric for feature selection. This approach returns the selected features that contain more information about the target-class than the information about other features [38]. Lin Tang introduced an information theory-based conditional infomax feature extraction (CIFE) algorithm to measure the class-relevancy and redundancy for feature selection [39].

In the recent past, the clustering technique is also adopted in feature selection. Song et al developed a feature selection framework and adopted the graph-based clustering technique to identify the similarity among the features for removing the redundant features [40]. Dhillon et al developed a feature selection algorithm based on information theory for text classification. In this approach, the hierarchical clustering is used to cluster the features or terms of documents for identifying their dependencies [41]. Li et al incorporated the clustering algorithm with

the chi-square statistical measure to select the features from statistical data [42]. Chow Huang employed the supervised clustering technique and mutual information for identifying the salient features from synthetic and real world datasets [43]. Mitra et al presented a feature selection approach by adopting the graph-based clustering approach to identify the similarity among the features for redundancy analysis. Sotoca Pla developed a feature selection method for classification based on feature similarity with hierarchical clustering [44].

Further, it is observed that the filter-based methods are computationally better than the wrapper and embedded methods. Therefore, the filter-based methods can be a suitable choice for high-dimensional space. The filter-based methods achieve high generality since they do not use the supervised learning algorithm.

2.2.4 Hybrid methods

The hybrid methods are the combination of filter and wrapper-based approaches. In general, processing the high-dimensional data is a difficult task with the wrapper method therefore the authors Bermejo et al developed a hybrid feature selection method known as filter-wrapper approach. In this approach, they used a statistical measure to rank the features based on their relevancy then the higher ranked features are given to the wrapper method so that the number of evaluations required for the wrapper method is linear. Thus, the computational complexity is reduced using hybrid method for medical data classification [45]. Ruiz et al developed a gene (feature) selection algorithm for selecting the significant genes for the medical diagnosis system. They used a statistical ranking approach to filter the features from high-dimensional space and the filtered features are fed into the wrapper approach. This combination of the filter and wrapper approach was used to distinguish the significant genes causing cancer disease in the diagnosis process [46].

Xie et al developed a hybrid approach for diagnosing the erythemato-squamous diseases. In this approach, F-score measure is used to rank the features to identify the relevant features (filter approach). The significant features are selected from the ranked features with the sequential forward floating search (SFFS) and SVM (wrapper method) [47]. Kannan Faez presented a hybrid feature selection framework. In this approach, ant colony optimization (ACO)-based local search (LS) is used with the symmetric uncertainty measure to rank the features [48]. Xie

et al designed a hybrid approach with F-score to identify the relevant attributes from a disease dataset. For feature subset generation from the relevant features, the searching strategies such as sequential backward floating search (SBFS), extended sequential forward search (ESFS), and sequential forward floating search (SFFS) are also employed [49]. Naseriparsa et al proposed a hybrid method using information gain and genetic algorithm-based searching method combined with a supervised learning algorithm [50]. Huda et al developed a hybrid feature selection method by combining the mutual information (MI) and artificial neural network (ANN) [51]. Gunal presented a hybrid feature selection method by combining filter and wrapper method for text classification. In this method, information gain measure is used for ranking the significant features and the genetic algorithm is used as the searching strategy with support vector machine [52].

Yang et al developed a hybrid method for classifying the micro array data[53]. In this method, the information gain and correlation metric are used for filter method and an improved binary particle swarm optimization (BPSO) method is used with the supervised learning algorithm as the wrapper method to improve the performance of the classification algorithm. The performance of this method is evaluated using kNN and SVM classifiers. To avoid the computational cost of the wrapper method, Bermejo presented a hybrid method by combining the filter and wrapper methods. In this method, the GRASP meta-heuristic based on stochastic algorithm is used as filter method for reducing the wrapper computation. Foithong et al also designed a hybrid feature selection method by combining the filter and the wrapper methods. In this method, the mutual information criterion is used for filtering the relevant features and the supervised learning algorithm is adopted as the wrapper method for evaluating features obtained from the filter method.

Further, it is observed that the hybrid methods are computationally intensive than the filter methods since they combine the wrapper and filter methods and have less generality compared to the filter methods since they use the supervised learning algorithm in feature selection process. These hybrid methods take more computational time than the filter-based methods.

CHAPTER 3

EXISTING SYSTEM

3.1 DBSTREAM ONLINE COMPONENT

Typical micro-cluster-based data stream clustering algorithms retain the density within each micro-cluster as some form of weight (e.g., the number of points assigned to the MC). Some algorithms also capture the dispersion of the points by recording variance. For reclustering, however, only the distances between the MCs and their weights are used. In this setting, MCs which are closer to each other are more likely to end up in the same cluster. This is even true if a density-based algorithm like DBSCAN [55] is used for reclustering since here only the position of the MC centers and their weights are used. The density in the area between MCs is not available since it is not retained during the online stage.

The basic idea of this work is that if we can capture not only the distance between two adjacent MCs but also the connectivity using the density of the original data in the area between the MCs, then the reclustering results may be improved. In the following we develop DBSTREAM which stands for density-based stream clustering.

3.1.1 Leader-Based Clustering

Leader-based clustering was introduced by Hartigan [56] as a conventional clustering algorithm. It is straight-forward to apply the idea to data streams.

DBSTREAM represents each MC by a leader (a data point defining the MC center) and the density in an area of a user-specified radius r (threshold) around the center. This is similar to DBSCAN concept of counting the points in an ϵ -neighborhood, however, here the density is not estimated for each point, but only for each MC which can easily be achieved for streaming data. A new data point is assigned to an existing MC (leader) if it is within a fixed radius of its center. The assigned point increases the density estimate of the chosen cluster and the MC

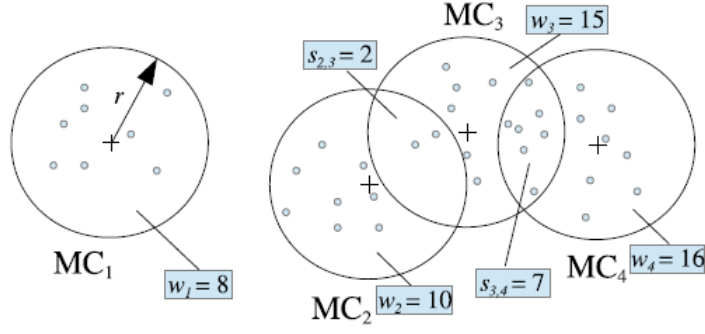


Figure 3.1: High and Low density MC regions

center is updated to move towards the new data point. If the data point falls in the assignment area of several MCs then all of them are updated. If a data point cannot be assigned to any existing MC, a new MC (leader) is created for the point. Finding the potential clusters for a new data point is a fixed-radius nearest-neighbor problem [57] which can be efficiently dealt with for data of moderate dimensionality.

The base algorithm stores for each MC a weight which is the number of data points assigned to the MC (see w_1 to w_4 in Figure 3.1). The density can be approximated by this weight over the size of the MC assignment area. Note that for simplicity the area here, however, the approach is not restricted to two-dimensional data. For higher-dimensional data volume is substituted for area.

3.1.2 Competitive Learning

New leaders are chosen as points which cannot be assigned to an existing MC. The positions of these newly formed MCs are most likely not ideal for the clustering. To remedy this problem, a competitive learning strategy introduced in [58] to move the MC centers towards each newly assigned point. To control the magnitude of the movement, a neighborhood function $h()$ similar to self-organizing maps. System implementation uses the popular Gaussian neighborhood function defined between two points, a and b , as $h(a, b) = \exp(-\|a - b\|^2 / (2\sigma^2))$ with $\sigma = r/3$ indicating that the used neighborhood size is 3 standard deviations. Since the stream is continuous, system do not use a learning rate to reduce the neighborhood size over time. This will accommodate slow concept drift and also has the desirable effect that MCs are drawn towards areas of higher density and ultimately will overlap, a prerequisite for capturing

shared density between MCs.

Note that moving centers could lead to collapsing MCs, i.e., the centers of two or more MCs converge to a single point. This will happen since the updating strategy makes sure that MCs are drawn to areas of maximal local density. Since several MCs representing the same area are unnecessary, many algorithms merge two converging MCs. However, experiments during development of the algorithm showed the following undesirable effect. New MCs are constantly created at the fringes of a dense area, then the MCs move towards the center and are merged while new MCs are again created at the fringes. This behavior is computationally expensive and degrades shared densities estimation. Therefore, the system prevent collapsing MCs by restricting the movement for MCs in case they would come closer than r to each other. This makes sure that the centers do not enter the assignment radius of neighboring MCs but will end up being perfectly packed together in dense areas giving us the optimal situation for estimating shared density.

3.1.3 Capturing Shared Density

The fact, that in dense areas MCs will have an overlapping assignment area, can be used to measure density between MCs by counting the points which are assigned to two or more MCs. The idea is that high density in the intersection area relative to the rest of the MCs area means that the two MCs share an area of high density and should be part of the same macro-cluster. In the example in Figure 3.1 we see that MC_2 and MC_3 are close to each other and overlap. However, the shared weight $s_{2,3}$ is small compared to the weight of each of the two involved MCs indicating that the two MCs do not form a single area of high density. On the other hand, MC_3 and MC_4 are more distant, but their shared weight $s_{3,4}$ is large indicating that both MCs form an area of high density and thus should form a single macro-cluster.

The shared density ρ_{ij} between two MCs can be estimate by $\rho_{ij} = s_{ij}/A_{ij}$, where s_{ij} is the shared weight and A_{ij} is the size of the overlapping area between the MCs. A shared density graph is an undirected weighted graph, where the set of vertices is the set of all MCs, and the set of edges represents all the pairs of MCs for which we have pairwise density estimates. Each


```

1: function UPDATE( $\mathbf{x}$ ) ▷ new data point  $\mathbf{x}$ 
2:    $\mathcal{N} \leftarrow \text{findFixedRadiusNN}(\mathbf{x}, \mathcal{MC}, r)$ 
3:   if  $|\mathcal{N}| < 1$  then ▷ create new MC
4:     add  $(\mathbf{c} = \mathbf{x}, t = t, w = 1)$  to  $\mathcal{MC}$ 
5:   else ▷ update existing MCs
6:     for each  $i \in \mathcal{N}$  do
7:        $mc_i[w] \leftarrow mc_i[w] \cdot 2^{-\lambda(t-mc_i[t])} + 1$ 
8:        $mc_i[\mathbf{c}] \leftarrow mc_i[\mathbf{c}] + h(\mathbf{x}, mc_i[\mathbf{c}])(\mathbf{x} - mc_i[\mathbf{c}])$ 
9:        $mc_i[t] \leftarrow t$  ▷ update shared density
10:    for each  $j \in \mathcal{N}$  where  $j > i$  do
11:       $s_{ij} \leftarrow s_{ij} \cdot 2^{-\lambda(t-s_{ij}[t])} + 1$ 
12:       $s_{ij}[t] \leftarrow t$ 
13:    end for
14:  end for ▷ prevent collapsing clusters
15:  for each  $(i, j) \in \mathcal{N} \times \mathcal{N}$  and  $j > i$  do
16:    if  $\text{dist}(mc_i[\mathbf{c}], mc_j[\mathbf{c}]) < r$  then
17:      revert  $mc_i[\mathbf{c}], mc_j[\mathbf{c}]$  to previous positions
18:    end if
19:  end for
20: end if
21:    $t \leftarrow t + 1$ 
22: end function

```

Figure 3.2: DBSTREAM Algorithm

edge is labeled with the pairwise density estimate

Note that most MCs will not share density with each other in a typical clustering. This leads to a very sparse shared density graph. This fact can be exploited for more efficient storage and manipulation of the graph. We represent the sparse graph by a weighted adjacency list S . Furthermore, during clustering we already find all fixed-radius nearest-neighbors. Therefore, obtaining shared weights does not incur any additional increase in search time.

Figure 3.2 shows the DBSTREAM approach and the used clustering data structures in detail. Microclusters are stored as a set \mathcal{MC} . Each micro-cluster is represented by the tuple (\mathbf{c}, w, t) representing the cluster center, the cluster weight and the last time it was updated, respectively. The weighted adjacency list S represents the sparse shared density graph which captures the weight of the data points shared by MCs. Since shared density estimates are also subject to fading, we also store a timestamp with each entry.

3.1.4 Fading and Forgetting Data

Cluster weights are faded in every time step by a factor of $2^{-\lambda}$, where $\lambda > 0$ is a user-specified fading factor. For example, if the current time-step is $t = 10$ and the weight w was last updated at $t_w = 5$ then we apply for fading the factor $2^{-\lambda(t-t_w)}$ resulting in the correct fading for five time steps. This approach keeps a timestamp with the time when fading was applied last for each value that is subject to fading.

The leader-based clustering algorithm only creates new and updates existing MCs. Over time, noise will cause the creation of low-weight MCs and concept shift will make some MCs obsolete. Fading will reduce the weight of these MCs over time and the reclustering has a mechanism to exclude these MCs. However, these MCs will still be stored in memory and make finding the fixed-radius nearest neighbors during the online clustering process slower. This problem can be addressed by removing weak MCs and weak entries in the shared density graph. In the following we define weak MCs and weak shared densities.

We define MC mc_i as a weak MC if its weight w_i increases on average by less than one new data point in a user-specified time interval t_{gap} . Also a weak entry in the shared density graph is defined as an entry between two MCs, i and j , which on average increases its weight s_{ij} by less than α from new points in the time interval t_{gap} . α is the intersection factor related to the area of the overlap of the MCs relative to the area covered by MCs. The rational of using α is that the overlap areas are smaller than the assignment areas of MCs and thus are likely to receive less weight. α will be discussed in detail in the reclustering phase.

Assume that every t_{gap} time step is checked and weak MCs and weak entries in the shared density graph are removed to recover memory and improve the clustering algorithms processing speed. To ensure that only weak entries are removed, the system uses the weight $w_{weak} = 2^{-\lambda T_{gap}}$. At any time, all entries that have a faded weight of less than w_{weak} are guaranteed to be weak. Noise entries (MCs and entries in the shared density graph) often receive only a single data point and will reach w_{weak} after t_{gap} time steps. Obsolete MCs or entries in the

Algorithm 2. Cleanup Process to Remove Inactive Micro-Clusters and Shared Density Entries from Memory

Require: $\lambda, \alpha, t_{gap}, t, \mathcal{MC}$ and S from the clustering.

```

1: function CLEANUP()
2:    $w_{weak} = 2^{-\lambda t_{gap}}$ 
3:   for each  $mc \in \mathcal{MC}$  do
4:     if  $mc[w] 2^{-\lambda(t-mc[t])} < w_{weak}$  then
5:       remove weak  $mc$  from  $\mathcal{MC}$ 
6:     end if
7:   end for
8:   for each  $s_{ij} \in S$  do
9:     if  $s_{ij} 2^{-\lambda(t-s_{ij}[t])} < \alpha w_{weak}$  then
10:      remove weak shared density  $s_{ij}$  from  $S$ 
11:    end if
12:  end for
13: end function

```

Figure 3.3: Clean-up Algorithm

shared density graph stop to receive data points and thus their weight will be faded till it falls below w_{weak} and then they are removed. It is easy to show that for an entry with a weight w it will take $t = \log_2(w)/\lambda + t_{gap}$ time steps to reach w_{weak} . Note that the definition of weak entries and w_{weak} is only used for memory management purpose. Reclustering uses the definition of strong entries. Therefore, the quality of the final clustering is not affected by the choice of t_{gap} as long as it is not set to a time interval which is too short for actual MCs and entries in the shared density graph to receive at least one data point. This clearly depends on the expected number of MCs and therefore depends on the chosen clustering radius r and the structure of the data stream to be clustered. A low multiple of the number of expected MCs is typically sufficient. The parameter t_{gap} can also be dynamically adapted during running the clustering algorithm. For example t_{gap} can be reduced to mark more entries as weak and remove them more often if memory or processing speed gets low. On the other hand, t_{gap} can be increased during clustering if not enough structure of the data stream is retained.

The cleanup process is shown in Figure 3.3. It is executed every t_{gap} time steps and removes weak MCs and weak entries in the shared density graph to recover memory and improve the clustering algorithms processing speed.

3.2 SHARED DENSITY RECLUSTERING

Reclustering represents the algorithms offline component which uses the data captured by the online component. For reclustering, MCs which are connected by areas of high density are

```

Require:  $\lambda, \alpha, w_{\min}, t, \mathcal{MC}$  and  $\mathbf{S}$  from the clustering.
1: function RECLUSTER
2:   weighted adjacency list  $\mathbf{C} \leftarrow \emptyset$   $\triangleright$  connectivity graph
3:   for each  $s_{ij} \in \mathbf{S}$  do  $\triangleright$  construct connectivity graph
4:     if  $\mathcal{MC}_i[w] \geq w_{\min} \wedge \mathcal{MC}_j[w] \geq w_{\min}$  then
5:        $c_{ij} \leftarrow \frac{s_{ij}}{(\mathcal{MC}_i[w] + \mathcal{MC}_j[w])/2}$ 
6:     end if
7:   end for
8:   return findConnectedComponents( $\mathbf{C} \geq \alpha$ )
9: end function

```

Figure 3.4: Reclustering Algorithm

joined together to form macro-clusters of arbitrary shape, while avoiding joining MCs which are close to each other but are separated by an area of low density.

Less dense clusters will also have a lower shared density. To detect clusters of different density correctly, the system defines connectivity relative to the densities (weights) of the participating clusters. That is, for two MCs, i and j , which are next to each other in the same macro-cluster connectivity is defined as $c_{ij} = s_{ij}/((w_i + w_j)/2)$ where s_{ij} is the weight in the intersecting area of MCs i and j and w_i and w_j are the MC weights. The connectivity graph is an undirected weighted graph with the micro clusters as vertices and edges are labelled with weights given by c_{ij} . Two MCs, i and j , are α -connected iff $c_{ij} \geq \alpha$, where α is the user-defined intersection factor.

Figure 3.4 shows the reclustering process. The parameters are the intersection factor α and the noise threshold w_{\min} . The connectivity graph \mathbf{C} is constructed using only shared density entries between strong MCs. Finally, the edges in the connectivity graph with a connectivity value greater than the intersection threshold are used to find connected components representing the final clusters.

3.3 DRAWBACKS

Although, complexity analysis and experiments reveal that the procedure can be effectively applied to data sets of moderate dimensionality, the worst-case memory requirements of the shared density graph grow extremely fast with higher dimensions.

CHAPTER 4

PROPOSED SYSTEM

4.1 PROBLEM DEFINITION

To improve the quality of clustering by efficiently removing irrelevant features to handle data of higher dimensions under reduced memory requirements.

4.2 SYSTEM ARCHITECTURE

With the aim of choosing a good subset of features with respect to the target concept, it was found that feature subset selection is an effective way to reduce dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility. Feature selection is also useful as part of the data analysis process, as it shows which features are important for prediction, and how these features are related. Irrelevant features, along with redundant features, severely affect the accuracy of the learning machines. Thus, feature subset selection should be able to identify and remove as much of the irrelevant and redundant information as possible. Moreover, "good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other."

The system architecture shown in figure 4.1 is composed of two connected components, irrelevant feature removal and redundant feature elimination. The former obtains features relevant to the target concept by eliminating irrelevant ones, and the latter removes redundant features from relevant ones via choosing representatives from different feature clusters, and thus produces the final subset. The irrelevant feature removal is straightforward once the right relevance measure is defined or selected, while the redundant feature elimination is a bit sophisticated.

In particular, the system adopt the minimum spanning tree (MST) based clustering algorithms, because they do not assume that data points are grouped around centers or separated by a regular geometric curve and have been widely used in practice. Based on the MST method, we

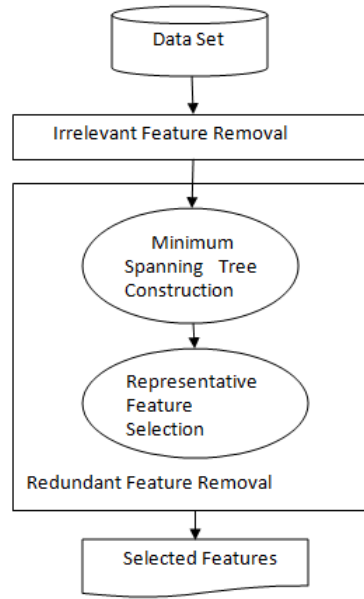


Figure 4.1: System Architecture

propose a N-Dimensional Feature Selection(NDFS) algorithm to effectively reduce the problem with higher dimensions.

4.3 MODULE DESCRIPTION

The system comprises of the following four modules:

Module 1. **Calculate Symmetric Uncertainty**

This module contains calculation of Symmetric Uncertainty to find the relevance of particular feature with target class

Module 2. **Minimum Spanning Tree Construction**

This module the constructs the minimum spanning tree and then partitioning of the MST into a forest with each tree representing a cluster.

Module 3. **Selection of Features**

In this module we do selection of most relevant features from the clusters which give us the reduced training dataset containing relevant and useful features only which improves efficiency.

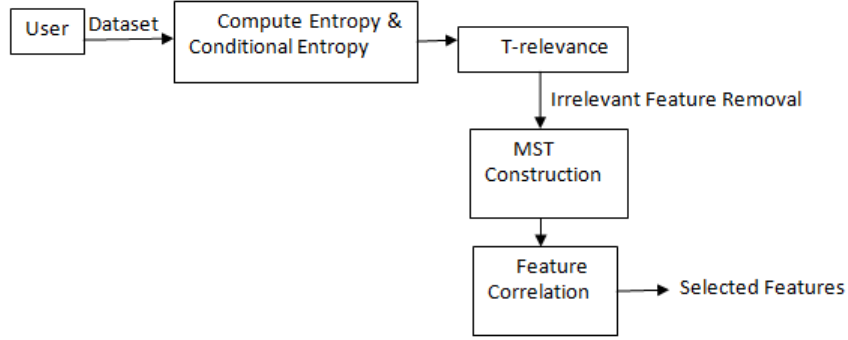


Figure 4.2: Proposed System Framework

Module 4. Feature Correlation

In this module we will use feature correlation measure for selecting most relevant features from cluster.

Figure 4.2 shows the structural flow of the proposed NDFS system. The system initially divides the features into clusters by using graph-theoretic clustering methods. Further the most representative feature that is strongly related to target classes is selected from each cluster to form the final subset of features. Features in different clusters are relatively independent, the clustering-based strategy of NDFS has a high probability of producing a subset of useful and independent features. The algorithm efficiently utilizes the memory required to store the higher dimensional data using the concept of minimum spanning trees.

4.4 ALGORITHM

Input : $D(F_1, F_2, \dots, F_m, C)$ - the given dataset.

Output : S - Selected feature subset

θ - the T-relevance threshold

- Part1 : Irrelevant Feature Removal

1. for $i=1$ to m do
2. $T\text{-Relevance} = SU(F_i, C)$

3. if T-Relevance $> \theta$ then
4. $S = S \cup F_i$;
- Part2 : Minimum spanning tree construction
5. $G = \text{NULL}$; // G is a complete graph
6. for each pair of features $F'_i, F'_j \subset S$ do
7. F-Correlation = $SU(F'_i, F'_j)$
8. Add F'_i and/or F'_j to G with F-Correlation as the weight of the corresponding edge;
9. minSpanTree = Kruskal(G); //Using Kruskal Algorithm to generate the MST
- Part3 : Tree Partition and Representative Feature Selection
10. Forest = minSpanTree
11. for each edge $E_{ij} \in \text{Forest}$ do
12. if $SU(F'_i, F'_j) < SU(F'_i, C) \wedge SU(F'_i, F'_j) < SU(F'_j, C)$ then
13. Forest = Forest - E_{ij}
14. $S = \Phi$
15. for each tree $T_i \in \text{Forest}$ do
16. $F_r^j = \text{argmax}_{F'k \in T_i} SU(F'k, C)$
17. $S = S \cup (F_r^j)$;
18. return S

The algorithm can be expected to be divided into 3 major parts:

1. The first part is concerned with removal of irrelevant features;
2. The second part is used for removing the redundant features and

3. The final part of the algorithm is concerned with representative feature selection.

Step 1: The data set 'D' with 'm' features $F = (F_1, F_2, \dots, F_m)$ and class 'C', compute the T-Relevance as symmetric uncertainty, $SU(F_i, C)$ value for every feature ($1 \leq i \leq m$). The symmetric uncertainty is defined as follows:

$$SU(F_i, C) = \frac{2 * Gain(F_i|C)}{H(F_i) + H(C)} \text{ where}$$

$$Gain(F_i|C) = H(F_i) - H(F_i/C) \text{ or } H(C) - H(F_i|C)$$

$H(F_i)$ and $H(F_i|C)$ denotes the feature entropies and conditional entropies respectively. In general they are given by:

$$H(X) = -\sum_x P(x) \log_2 P(x)$$

$$H(X|Y) = -\sum_y P(y) \sum_x P(x|y) \log_2 P(x|y)$$

where, $p(x)$ is the probability density function and $p(x|y)$ is the conditional probability density function.

Step 2: Here the first step is to calculate the F-Correlation $SU(F'_i, F'_j)$ value for each pair of features F'_i and F'_j . Then, seeing features F'_i and F'_j as vertices and $SU(F'_i, F'_j)$ the edge between vertices F'_i and F'_j a weighted complete graph $G = (V, E)$ is constructed which is an undirected graph. The complete graph reflects the correlations among the target-relevant features. A threshold value (θ) is defined to calculate the relevance among the selected features. If any feature exceeds a particular threshold value then that feature is treated as irrelevant.

Step 3: Here, unnecessary edges can be removed. Each tree $T_j \in \text{Forest}$ shows a cluster that is denoted as $V(T_j)$, which is the vertex set of T_j . For each cluster $V(T_j)$, select a representative feature whose T-Relevance $SU(F'_j, C)$ is the highest. All F'_j ($j = 1 \dots |\text{Forest}|$) consist of the final feature subset F_j^r .

4.5 ADVANTAGES

The feature selection model exhibits several advantages over the existing models as follows:

- Reduced memory requirement
- Retain good feature subset
- Eliminates irrelevant features
- Efficiently handles higher dimensional data

CHAPTER 5

SYSTEM SPECIFICATIONS

5.1 HARDWARE REQUIREMENTS

- Processor - Pentium IV
- Speed - 1.1 GHz
- RAM - 128 MB(min)
- Storage - 4 GB

5.2 SOFTWARE REQUIREMENTS

- Operating System - Windows
- Front End - JavaEE
- Back End - MySQL
- IDE - NetBeans

5.3 SOFTWARE ENVIRONMENT

5.3.1 NetBeans IDE

NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML5. It is also an application platform framework for Java desktop applications and others. The NetBeans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms that supporting a compatible JVM. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform (including the NetBeans IDE itself) can be extended by third party developers.

NetBeans IDE is an open-source integrated development environment. NetBeans IDE supports development of all Java application types (Java SE (including JavaFX), Java ME, web, EJB and mobile applications) out of the box. Among other features are an Ant-based project system, Maven support, refactorings, version control (supporting CVS, Subversion, Git, Mercurial and Clearcase).

NetBeans IDE 6.8 is the first IDE to provide complete support of Java EE 6 and the GlassFish Enterprise Server v3. Developers hosting their open-source projects on kenai.com additionally benefit from instant messaging and issue tracking integration and navigation right in the IDE, support for web application development with PHP 5.3 and the Symfony framework, and improved code completion, layouting, hints and navigation in JavaFX projects. It is a framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required. The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

- User interface management (e.g. menus and toolbars)
- User settings management
- Storage management (saving and loading any kind of data)
- Window management
- Wizard framework (supports step-by-step dialogs)
- NetBeans Visual Library

Modularity: All the functions of the IDE are provided by modules. Each module provides a well defined function, such as support for the Java language, editing, or support for the CVS versioning system, and SVN. NetBeans contains all the modules needed for Java development in a single download, allowing the user to start working immediately. Modules also allow NetBeans to be extended. New features, such as support for other programming languages, can be added by installing additional modules. For instance, Sun Studio, Sun Java Studio Enterprise, and Sun Java Studio Creator from Sun Microsystems are all based on the NetBeans IDE.

5.3.2 Front End-Java EE

Java technology is both a programming language and a platform. It is a high level language that can be characterized by all of the following buzzwords:

- Simple
- Object oriented
- Distributed
- Multithreaded
- Dynamic
- Architecture neutral
- Portable
- High performance
- Robust
- Secure

In the java programming language, all source code is first written in plain text files ending with .java extension. Those source files are then compiled into class files by the javac compiler. A .class file does not contain code that is native to your processor; it instead contains bytecodes, the machine language of the java virtual machine (Java VM).

Java Platform

The java platform differs from most other platforms in that it is a software-only platform that runs on top of other hardware-based platforms. The java platform has two components:

- The java virtual machine
- The java application programming interface (API)

The API is a large collection of ready-made software components that provide many useful capabilities. It is grouped into libraries of related classes and interfaces; these libraries are known as packages. As a platform-independent environment, the java platform can be a bit slower than native code. However, advances in compiler and virtual machine technologies are bringing performance close to that of native code without threatening portability. Java platform gives you the following features:

- **Development tools:** the development tools provide everything you will need for compiling, running, monitoring, debugging and documenting your applications. As a new developer, the main tools you will be using the javac compiler, the java launcher and the javadoc documentation tool.
- **Application programming interface (API) :** the API provides the core functionality of the java programming language. It offers a wide array of useful classes ready for use in your own applications.
- **Deployment technologies:** the JDK software provides standard mechanisms such as the java web start software and java plug-in software for deploying your applications to end users.
- **User interface toolkits:** the swing and java 2d toolkits make it possible to create sophisticated graphical user interfaces (GUIs).
- **Integration libraries:** integration libraries such as the java IDL, API, JDBC API, java naming and directory interface API, java RMI and java remote method invocation over internet inter-ORB protocol technology (java RMI-IIOP technology) enable database access and manipulation of remote objects.

Java technology will help you do the following:

- **Get started quickly:** although the java programming language is powerful object oriented language, it is easy to learn, especially for programmers already familiar with C or C++.
- **Write less code:** comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the java programming language can be four times smaller than the same program written in C++.

- Write better code: the java programming language encourages good coding practices, and automatic garbage collection helps you avoid memory leaks. Its object orientation, its javaBeans component architecture, and its wide ranging, easily extendible API let you reuse existing, tested code and introduce fewer bugs.
- Develop programs more quickly: the java programming language is simpler than C++, and as such, your development time could be up to twice as fast when writing in it. Your programs will also require fewer lines of code.
- Avoid platform dependencies: you can keep your program portable by avoiding the use of libraries written in other languages. This is a major function of java technology.
- Write once, run anywhere: because applications written in the java programming language are compiled into machine-independent bytecodes, they run consistently on any java platform.
- Distribute software more easily: with java web start software, users will be able to launch your applications with a single click of the mouse. An automatic version check at startup ensures that users are always upto date with the latest version of your software. If an update is available, the java web start software will automatically update their installation.

JAVA AND JAVA ENTERPRISE EDITION

JSP is now an integral part for developing web based applications using java because of its ability to separate presentation from logic implementation by combining standard markup text with scripting elements and object oriented components. JSP provides excellent front end technology for applications that are deployed over the web. Java Platform, Enterprise Edition (Java EE) is the standard in community-driven enterprise software. Java EE is developed using the Java Community Process, with contributions from industry experts, commercial and open source organizations, Java User Groups, and countless individuals. Each release integrates new features that align with industry needs, improves application portability, and increases developer productivity. The Java EE platform is designed to help developers create large-scale, multi-tiered, scalable, reliable, and secure network applications. A shorthand name for such applications is "enterprise applications," so called because these applications are designed to solve the problems encountered by large enterprises. The benefits of an enterprise application

are helpful, even essential, for individual developers and small organizations in an increasingly networked world. The features that make enterprise applications powerful, like security and reliability, often make these applications complex. The Java EE platform reduces the complexity of enterprise application development by providing a development model, API, and runtime environment that allow developers to concentrate on functionality. A proto typical web application can be composed from:

- Java runtime environment running in the server
- JSP page that handle request and generate the dynamic content
- Servlet that handle requests and generate dynamic content
- Server side java beans components that encapsulate behavior and state
- Static HTML, DHTML, XHTML, XML and similar pages.
- Client side java applets, java beans components and arbitrary java class files.
- Java runtime environments (downloadable via the plugin) running in the client

5.3.3 Back End :- MySQL

MySQL is the world's second most widely used open-source relational database management system(RDBMS). MySQL is an open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX, and Windows . Although it can be used in a wide range of applications, MySQL is most often associated with web-based applications and online publishing and is an important component of an open source enterprise stack called LAMP. Major features as available in MySQL 5.6:

- A broad subset of ANSI SQL 99, as well as extensions
- Cross-platform support
- Stored procedures, using a procedural language that closely adheres to SQL/PSM

- Triggers
- Cursors
- Updatable views
- Online DDL when using the InnoDB Storage Engine.
- Information schema
- Performance Schema
- A set of SQL Mode options to control runtime behavior, including a strict mode to better adhere to SQL standards.
- X/Open XA distributed transaction processing (DTP) support; two phase commit as part of this, using the default InnoDB storage engine
- Transactions with savepoints when using the default InnoDB Storage Engine. The NDB Cluster Storage Engine also supports transactions.
- ACID compliance when using InnoDB and NDB Cluster Storage Engines
- SSL support
- Query caching
- Sub-SELECTs (i.e. nested SELECTs)
- Full-text indexing and searching
- Embedded database library
- Partitioned tables with pruning of partitions in optimizer
- Shared-nothing clustering through MySQL Cluster
- Multiple storage engines, allowing one to choose the one that is most effective for each table in the application.

- Native storage engines InnoDB, MyISAM, Merge, Memory (heap), Federated, Archive, CSV, Blackhole, NDB Cluster.
- Commit grouping, gathering multiple transactions from multiple connections together to increase the number of commits per second.

Ensuring high availability requires a certain amount of redundancy in the system. For database systems, the redundancy traditionally takes the form of having a primary server acting as a master, and using replication to keep secondaries available to take over in case the primary fails. This means that the "server" that the application connects to is in reality a collection of servers, not a single server. In a similar manner, if the application is using a shared database, it is in reality working with a collection of servers, not a single server. In this case, a collection of servers is usually referred to as a farm. One of the projects aiming to provide high availability for MySQL is MySQL Fabric, an integrated system for managing a collection of MySQL servers, and a framework on top of which high availability and database sharing is built. MySQL Fabric is open-source and is intended to be extensible, easy to use, and to support procedure execution even in the presence of failure, providing an execution model usually called resilient execution.

CHAPTER 6

RESULTS AND DISCUSSIONS

The performance evaluation and experimental analysis of the proposed system are described in the following sections.

6.1 EXPERIMENTAL RESULTS

The figure 6.1 illustrates loading of dataset where when a user loads the dataset, the contents of the file are displayed in the text box.

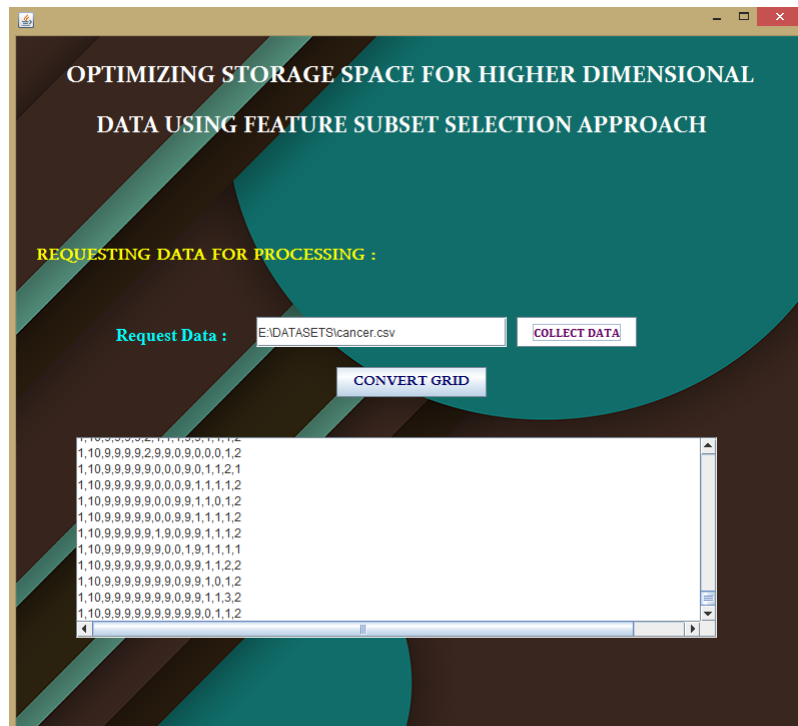


Figure 6.1: Loading the Dataset

Figure 6.2 shows the initial micro-clusters formed. Estimated standard deviation is shown in figure 6.3.

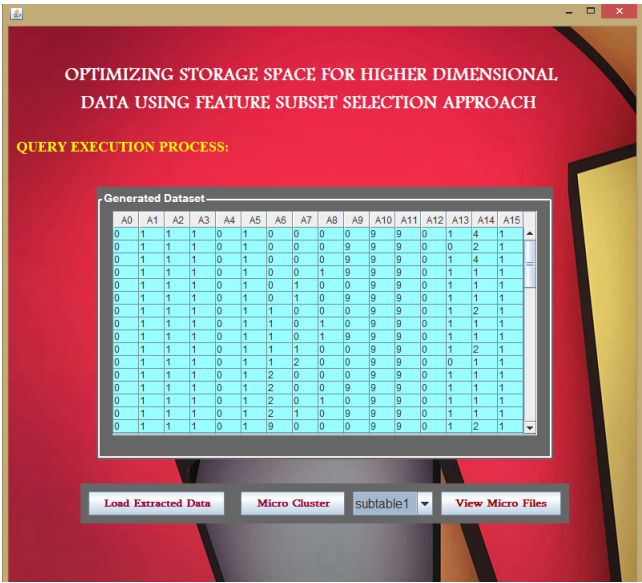


Figure 6.2: Cluster Formation

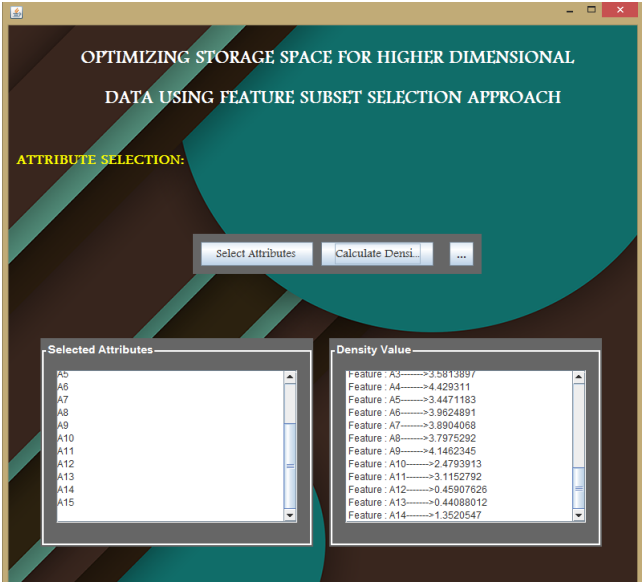


Figure 6.3: Estimation of Standard Deviation

Figure 6.4 shows the density estimation of each cluster. The entropy and conditional entropy of each cluster is evaluated in figure 6.5.

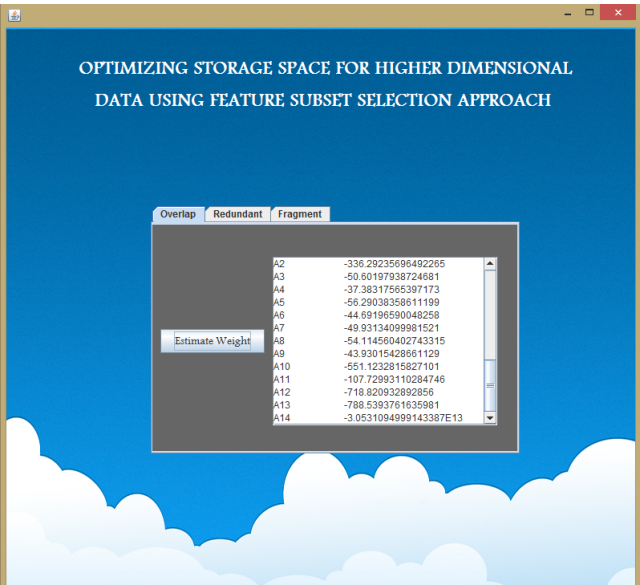


Figure 6.4: Density Estimation

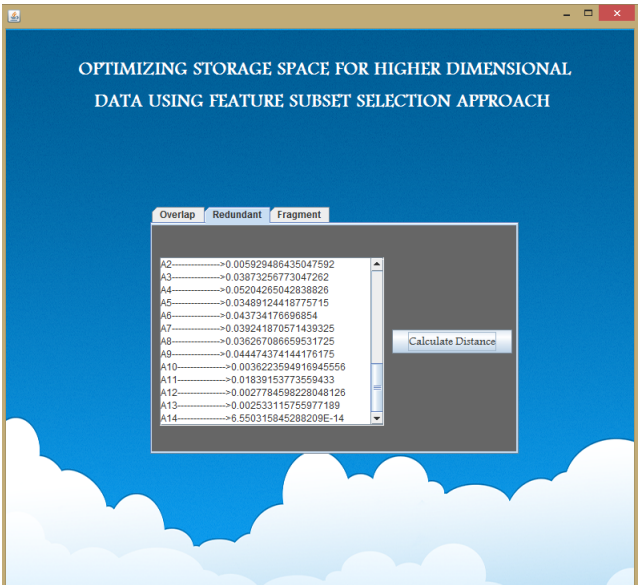


Figure 6.5: Measuring Conditional Entropy

Figure 6.6 and figure 6.7 shows the individual feature relevance and T-relevance measure respectively.

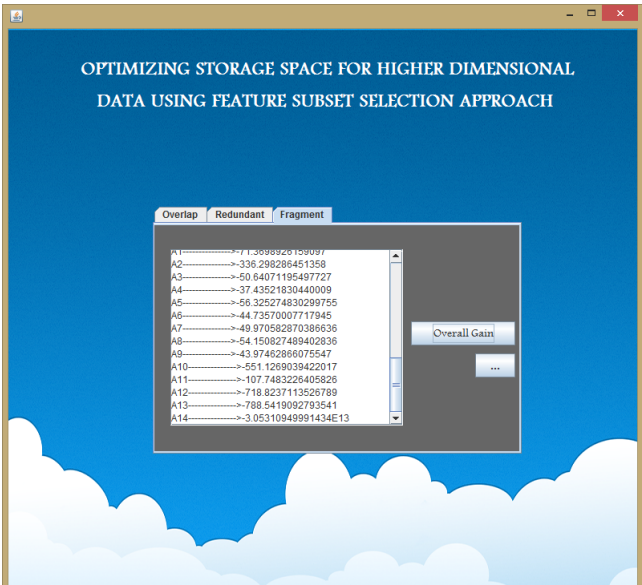


Figure 6.6: Individual Feature Relevance Measure

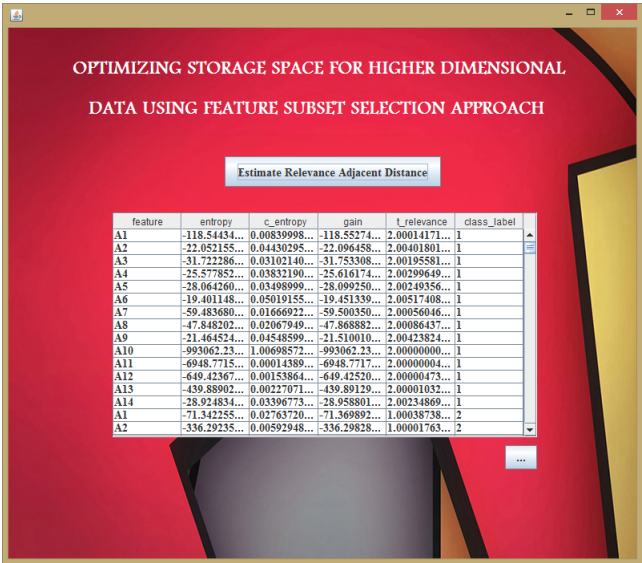


Figure 6.7: T-Relevance Estimation

In figure 6.8 irrelevant features are removed and feature correlation between each pair of relevant features are computed. Figure 6.9 shows the minimum spanning tree formed using the relevant set of features.

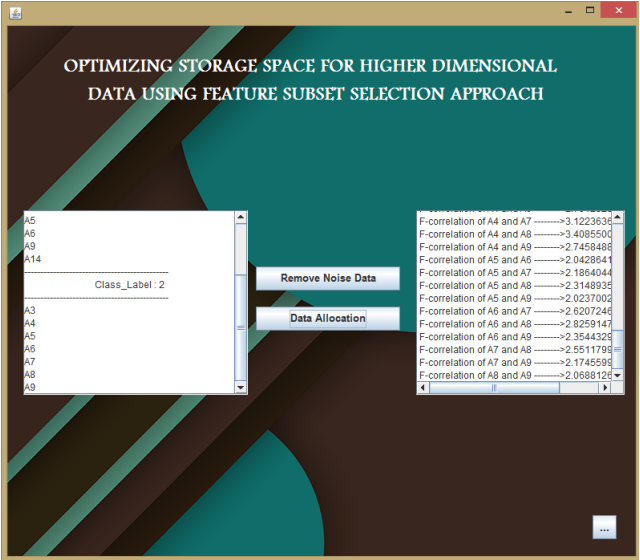


Figure 6.8: Removal of Irrelevant Features and correlation Estimation

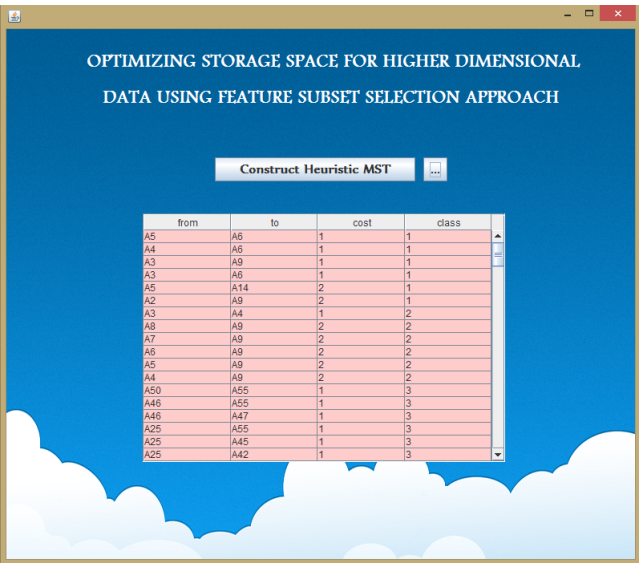
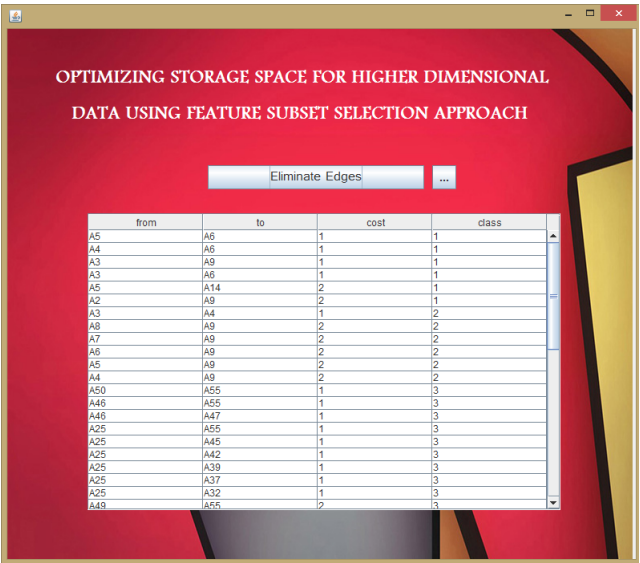


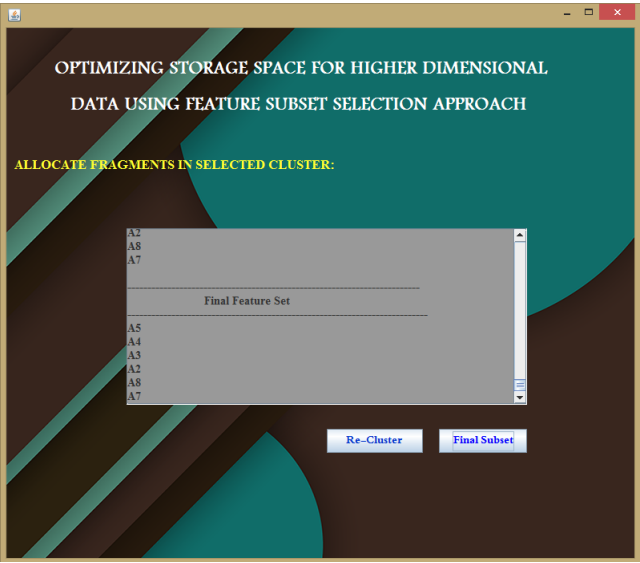
Figure 6.9: Minimum Spanning Tree Construction

Redundant features are removed in figure 6.10 by eliminating the edges that do not satisfy the threshold feature correlation. Figure 6.11 shows the final subset of features generated.



from	to	cost	class
A5	A6	1	1
A4	A6	1	1
A3	A9	1	1
A3	A6	1	1
A5	A14	2	1
A2	A9	2	1
A3	A4	1	2
A8	A9	2	2
A7	A9	2	2
A6	A9	2	2
A5	A9	2	2
A4	A9	2	2
A50	A55	1	3
A46	A55	1	3
A46	A47	1	3
A25	A55	1	3
A25	A45	1	3
A25	A42	1	3
A25	A39	1	3
A25	A37	1	3
A25	A32	1	3
A43	A55	2	3

Figure 6.10: Eliminating Redundant Features



ALLOCATE FRAGMENTS IN SELECTED CLUSTER:

Final Feature Set
A5
A4
A3
A2
A8
A7

Re-Cluster Final Subset

Figure 6.11: Final Feature Subset Generation

6.2 PERFORMANCE EVALUATION

The parameters used for performance evaluation includes:

1. Proportion of the features selected
2. The time and space Complexity
3. Type of Data Handled
4. Average no. of edges per cluster

- **Proportion of the features selected**

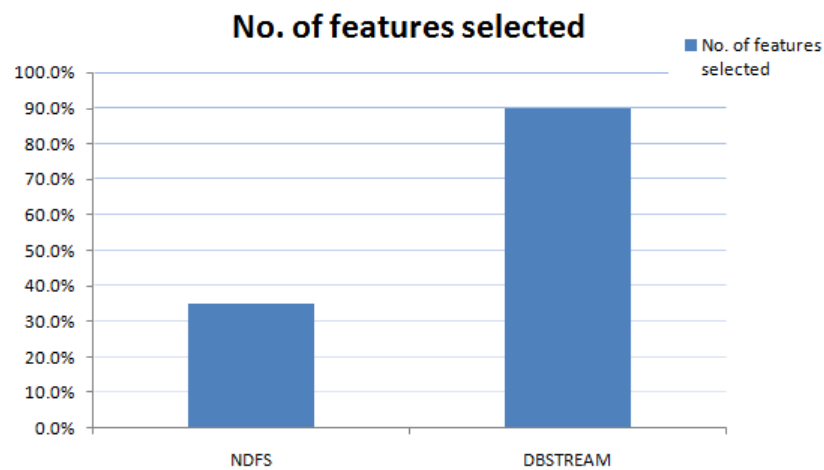


Figure 6.12: Proportion of Features Selected

The figure 6.2 depicts the the proportion of features used by both the existing and proposed systems. In X-axis the different algorithms are plotted and in Y-axis the number of features selected is plotted in terms of percentage measure. It can be clearly seen that the proposed system uses one-fourth of the entire features.

- **Type of Data Handled**

The figure 6.1 clearly analyses the type of data handles by the existing and proposed algorithms. The performance evaluation clearly shows that the NDFS algorithm efficiently handles the noisy and higher dimensional data better than DBSTREAM.

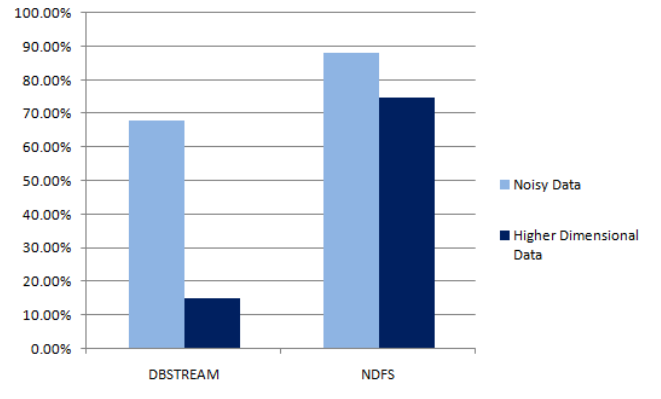


Figure 6.13: Analysis of Data handled

- **Average number of edges per cluster**

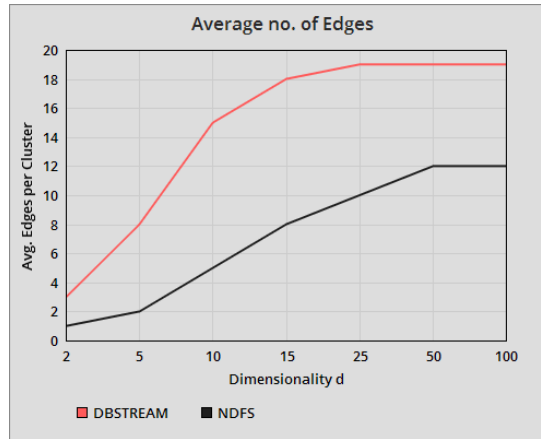


Figure 6.14: Average number of edges per cluster

Figure 6.3 shows that the average number of edges in the graph grows with the dimensionality of the data. However the no. of edges in the proposed system is comparatively less than the existing work.

- **The time and space Complexity**

The major amount of work for NDFS algorithm involves the computation of SU values for T-Relevance and F-Correlation, which has linear complexity in terms of the number of instances in a given data set.

Algorithm	Execution Time	Memory Utilization
DBSTREAM	Moderate	Increased storage
NDFS	Low	Optimal storage

Table 6.1: Comparison on Time and Space Utilization

The first part of the algorithm has a linear time complexity $O(m)$ in terms of the number of features m . Assuming $k(1 \leq k \leq m)$ features are selected as relevant ones in the first part, when $k = 1$, only one feature is selected. Thus, there is no need to continue the rest parts of the algorithm, and the complexity is $O(m)$. When $1 < k \leq m$, the second part of the algorithm firstly constructs a complete graph from relevant features and the complexity is $O(\log k)$, and then generates a MST from the graph using Kruskals algorithm whose time complexity is $O(\log k)$. The third part partitions the MST and chooses the representative features with the complexity of $O(k)$. Thus when $1 < k \leq m$, the complexity of the algorithm is $O(m + \log k)$. Thus, NDFS has a better runtime performance with high dimensional data. Table 2.2 and 2.3 compares and analyses the time and space complexity of the algorithms

Algorithm	Space Complexity	Time complexity
DBSTREAM	$O(t^2)$	$O(nt^2)$
NDFS	$O(m + \log k)$	$O(\log k)$

Table 6.2: Time and Space Complexity Analysis

CHAPTER 7

CONCLUSION

NDFS proposes a feature selection algorithm for high dimensional data. The algorithm includes (i) irrelevant features removal (ii) construction of a minimum spanning tree (MST), and (iii) selection of representative features. N-Dimensional Feature selection algorithm enables to recognize and remove as much of the unrelated and redundant information. In the proposed algorithm, a minimum spanning tree is constructed for faster searching of relevant data from high dimensional data. Removal of irrelevant features reduces the volume of data to be processed thereby enabling smooth handling of higher dimensional data. NDFS algorithm will obtain the best proportion of selected features, the best runtime, and the better space utilization. Overall the system will be effective in generating more relevant and accurate features which can provide faster results.

In future more challenging domains with more features and a higher proportion of irrelevant ones will require more sophisticated methods for feature selection. Although further increase in efficiency cannot eliminate problems caused by exponential growth in the number of feature sets, exploring different types of correlation measures for better halting criteria and inventing more intelligent techniques for selecting an initial set of features from which to start the search can improve the performance without sacrificing useful feature sets.

REFERENCES

- [1] **Michael Hahsler, Matthew Bolanos**,(2016) Clustering Data Streams Based on Shared Density between Microclusters, in *IEEE Transaction on Knowledge and Data Engineering*, Vol.28, No.6.
- [2] **Amineh Amini, Teh Ying Wah**, (2013)Leaden-Stream: A Leader Density-Based Clustering Algorithm over Evolving Data Stream, *Journal of Computer and Communications*, vol.1, pp.26-31.
- [3] **Ntoutsi I, Zimek A, Palpanas T**, (2012)Density-based projected clustering over high dimensional data streams, *Proc. the 12th SIAM Int. Conf. Data Mining*, pp.987-998.
- [4] **Cao F, Ester M, Qian W, Zhou A**, (2006)Density based clustering over an evolving data stream with noise, *Proc. the 2006 SIAM Conference on Data Mining*, pp.328-339.
- [5] **Liu, H Setiono, R**(1996) A probabilistic approach to feature selection-a filter solution, *Proceedings of Eighteenth International Conference on Machine Learning, Italy*, pp. 319-327.
- [6] **Lin, S.W, Tseng, TY, Chou, SY Chen, SC** (2008), A simulated-annealing-based approach for simultaneous parameter optimization and feature selection of backpropagation networks, *Expert Systems with Applications*, vol. 34, no.2, pp.1491-1499.
- [7] **Zhang. H Sun. G**(2002),Feature selection using tabu search method, *Pattern recognition*, vol. 35, no.3, pp.701-711.
- [8] **Tahir, MA, Bouridane, A Kurugollu, F**(2007), Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier *Pattern Recognition Letters*, vol. 28, no.4, pp.438-446.
- [9] **Aghdam, MH, Ghasem-Aghaee, N Basiri, ME**(2009), Text feature selection using ant colony optimization, *Expert systems with applications*, vol. 36, no.3, pp.6843- 6853.

- [10] **Sivagaminathan, RK Ramakrishnan, S**(2007), A hybrid approach for feature subset selection using neural networks and ant colony optimization, *Expert systems with applications*, vol. 33, no.1, pp.49-60.
- [11] **Sreeja, NK Sankar, A** (2015), Pattern Matching based Classification using Ant Colony Optimization based Feature Selection, *Applied Soft Computing*, vol. 31, pp.91-102.
- [12] **Welikala, R.A, Fraz, MM, Dehmeshki, J, Hoppe, A, Tah, V, Mann, S, Williamson, TH Barman, SA**(2015), Genetic algorithm based feature selection combined with dual classification for the automated detection of proliferative diabetic retinopathy, *Computerized Medical Imaging and Graphics*, vol. 43, pp.64-77.
- [13] **Erguzel, TT, Ozekes, S, Tan, O Gultekin, S**(2015), Feature Selection and Classification of Electroencephalographic Signals an Artificial Neural Network and Genetic Algorithm Based Approach, *Clinical EEG and Neuroscience*, vol. 46, no.4, pp.321- 326.
- [14] **Oreski, S Oreski, G** (2014), Genetic algorithm-based heuristic for feature selection in credit risk assessment, *Expert systems with applications*, vol. 41, no.4, pp.2052- 2064.
- [15] **Li, S, Wu, H, Wan, D Zhu, J,**(2011), An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine, *Knowledge-Based Systems*, vol. 24, no.1, pp.40-48.
- [16] **Das, N, Sarkar, R, Basu, S, Kundu, M, Nasipuri, M Basu, DK**(2012), A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application, *Applied Soft Computing*, vol.12, no.5, pp.1592-1606.
- [17] **Wang, Y, Chen, X, Jiang, W, Li, L, Li, W, Yang, L, Liao, M, Lian, B, Lv, Y, Wang, S Wang, S**(2011), Predicting human microRNA precursors based on an optimized feature subset generated by GA SVM, *Genomics*, vol. 98, no.2, pp.73-78.
- [18] **Chen, LF, Su, CT, Chen, KH Wang, PC**(2012), Particle swarm optimization for feature selection with application in obstructive sleep apnea diagnosis, *Neural Computing and Applications*, vol. 2, no. 8, pp.2087- 2096.

- [19] **Yang, H, Du, Q Chen, G**(2012), Particle swarm optimization-based hyperspectral dimensionality reduction for urban land cover classification, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5 no.2, pp.544- 554.
- [20] **Kohavi, R John, GH**(1997), Wrappers for feature subset selection, *Artificial intelligence*, vol. 97, no.1, pp.273-324.
- [21] **Inza, I, Larranaga, P, Etxeberria, R Sierra, B**(2000), Feature subset selection by Bayesian network-based optimization, *Artificial intelligence*, vol. 123, no. 1, pp.157-184.
- [22] **Dy, JG Brodley, CE**(2000), Feature subset selection and order identification for unsupervised learning, *proceedings In Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 247-254.
- [23] **Maldonado, S Weber, R**(2009), A wrapper method for feature selection using support vector machines, *Information Sciences*, 179(13), pp.2208-2217.
- [24] **Gutlein, M, Frank, E, Hall, M Karwath, A**(2009),Large-scale attribute selection using wrappers, *Proceeding of IEEE Symposium on Computational Intelligence and Data Mining, Nashville, TN, USA*, pp. 332-339.
- [25] **Kabir, MM, Islam, MM Murase, K**(2010), A new wrapper feature selection approach using neural network, *Neurocomputing*, vol. 73, no. 16, pp.3273- 3283.
- [26] **Stein, G, Chen, B, Wu, AS Hua, KA**(2005), Decision tree classifier for network intrusion detection with GA-based feature selection, *Proceedings of the forty-third ACM Annual Southeast regional conference, Kennesaw, GA, USA*, vol. 2, pp. 136-141.
- [27] **Zhuo, L, Zheng, J, Li, X, Wang, F, Ai, B Qian, J**(2008), A genetic algorithm based wrapper feature selection method for classification of hyperspectral images using support vector machine, *Proceedings of Geoinformatics and Joint Conference on GIS and Built Environment: Classification of Remote Sensing Images*, pp. 71471J- 71471J.
- [28] **Loughrey, J Cunningham, P**(2005), Overfitting in wrapper-based feature subset selection: The harder you try the worse it gets, *Proceedings of Research and Development in Intelligent Systems, Springer London*, pp. 33-43.

- [29] **Neumann, J, Schnorr, C Steidl, G**(2004), SVM-based feature selection by direct objective minimisation, *Proceeding of the twenty-sixth DAGM Symposium on Pattern Recognition, Germany*, pp. 212-219.
- [30] **Maldonado, S, Weber, R Famili, F**(2014),Feature selection for high-dimensional class-imbalanced data sets using Support Vector Machines, *Information Sciences*, vol. 286, pp.228-246.
- [31] **Kira, K Rendell, LA**(1992), A practical approach to feature selection, *Proceedings of the ninth international workshop on Machine learning, Aberdeen, Scotland, UK* (pp. 249-256).
- [32] **Kononenko, I**(1994), Estimating attributes: analysis and extensions of RELIEF, *Proceeding of European Conference on Machine Learning, Catania, Italy*, pp. 171-182.
- [33] **Holte, RC**(1993), Very simple classification rules perform well on most commonly used datasets, *Machine learning*, vol.11, no.1, pp.63-90.
- [34] **Yang, HH Moody, JE**(1999), Data Visualization and Feature Selection: New Algorithms for Nongaussian Data, *Advances in Neural Information Processing Systems*, vol. 99, pp. 687-693.
- [35] **Peng, H, Long, F Ding C**(2005), Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no.8, pp.1226-1238.
- [36] **Battiti, R**(1994), Using mutual information for selecting features in supervised neural net learning, *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp.537-550.
- [37] **Fleuret, F**(2004), Fast binary feature selection with conditional mutual information, *The Journal of Machine Learning Research*, vol. 5, pp.1531-1555.
- [38] **Meyer, PE Bontempi, G**(2006), On the use of variable complementarity for feature selection in cancer classification, *Applications of Evolutionary Computing*, pp. 91-102.

- [39] **Lin, D Tang, X**(2006), Conditional infomax learning: an integrated framework for feature extraction and fusion, *Proceeding of ninth European Conference on Computer Vision, Graz*, pp. 68-82.
- [40] **Song, Q, Ni, J Wang, G** (2013), A fast clustering-based feature subset selection algorithm for high-dimensional data, *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no.1, pp.1-14.
- [41] **Dhillon, IS, Mallela, S Kumar, R**(2003), A divisive information theoretic feature clustering algorithm for text classification, *The Journal of Machine Learning Research*, vol. 3, pp.1265-1287.
- [42] **Li, Y, Luo, C, Chung, SM** (2008), Text clustering with feature selection by using statistical data. *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no.5, pp.641-652.
- [43] **Chow, TW Huang, D**(2005), Estimating optimal feature subsets using efficient estimation of highdimensional mutual information, *IEEE Transactions on Neural Networks*, vol.16, no.1, pp.213-224.
- [44] **Sotoca, JM Pla, F**(2010),Supervised feature selection by clustering using conditional mutual information-based distances, *Pattern Recognition*, vol. 43, no.6, pp.2068- 2081
- [45] **Bermejo, P, Gamez, J Puerta, J**(2008), On incremental wrapper-based attribute selection: experimental analysis of the relevance criteria, *Proceedings of International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, France*, pp.638-645.
- [46] **Ruiz, R, Riquelme, JC Aguilar-Ruiz**, (2006), Incremental wrapper-based gene selection from microarray data for cancer classification *Pattern Recognition*, vol. 39, no. 12, pp.2383-2392.
- [47] **Xie, J, Xie, W, Wang, C Gao, X**(2010), A Novel Hybrid Feature Selection Method Based on IFSFFS and SVM for the Diagnosis of Erythemato-Squamous Diseases, *Proceedings of Workshop on Applications of Pattern Analysis, Cumberland Lodge, Windsor, UK*, pp. 142-151.

- [48] **Kannan, SS Ramaraj, N** (2010), A novel hybrid feature selection via Symmetrical Uncertainty ranking based local memetic search algorithm, *Knowledge-Based Systems*, vol. 23, no. 6, pp.580-585.
- [49] **Xie, J, Lei, J, Xie, W, Shi, Y Liu, X**(2013), Two-stage hybrid feature selection algorithms for diagnosing erythemato-squamous diseases, *Health Information Science and Systems*, vol.1, no.10, pp.2-14.
- [50] **Naseriparsa, M, Bidgoli, AM Varae, T**(2013), A Hybrid Feature Selection method to improve performance of a group of classification algorithms, *International Journal of Computer Applications*, vol. 69, no. 17, pp. 0975-8887.
- [51] **Huda, S, Yearwood, J Stranieri, A**(2011), Hybrid wrapper-filter approaches for input feature selection using maximum relevance-minimum redundancy and artificial neural network input gain measurement approximation (ANNIGMA), *Proceedings of the Thirty-Fourth Australasian Computer Science Conference, Australia*, vol. 113, pp. 43-52.
- [52] **Gunal, S**(2012), Hybrid feature selection for text classification, *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 20, no.2, pp.1296-1311.
- [53] **Yang CS, Chuang LY, Ke CH, Yang CH**,(2008),A hybrid feature selection method for microarray classification, *IAENG International Journal of Computer Science*, vol. 35, no. 3, pp. 1-3.
- [54] **M. Ester, H.P. Kriegel, J. Sander, and X. Xu**,(1996) A density-based algorithm for discovering clusters in large spatial databases with noise, *in Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, , pp. 226-231.
- [55] **J. A. Hartigan**, (1975),Clustering Algorithms, *99th ed. New York, NY, USA: Wiley*,.
- [56] **J. L. Bentley**,(1975) A survey of techniques for fixed radius near neighbor searching, *Stanford Linear Accelerator Center, Menlo Park, CA, USA*, Tech. Rep. CS-TR-75-513.
- [57] **C. Isaksson, M. H. Dunham, and M. Hahsler**,(2012),Sostream: Self organizing density-based clustering over data stream, *Proc. Mach. Learn. Data Mining Pattern Recog.*, vol. 7376, pp. 264-278.

LIST OF PUBLICATIONS

- [1] **Donia Augustine**,(2017) A Survey on Density based Micro-clustering Algorithms for Data Stream Clustering, *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol.7, Iss.1, pp. 186-190
- [2] **Donia Augustine**,(2017), Optimizing Storage Space for Higher-Dimensional Data Using Feature Subset Selection Approach, *International Journal of Engineering Research Management Technology*, Volume 6, Issue 5.