Universitat Politècnica de Catalunya

Master Thesis Project

Histogram based Hierarchical Data Representation for Microarray Classification

Author: Sandeep Kottath
Advisor: Philippe Salembier Clairon

Document presented to obtain the Master’s degree for the European Master of Research on Information and Communication Technologies

Barcelona, September 3, 2012
Dedicated to my family, friends, teachers and loved ones
Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Prof. Philippe Salembier for the continuous support and guidance throughout my masters studies and master thesis. I think the best part of my master thesis is the lessons I learned from him to become a good researcher.

Besides my advisor, I would like to thank the rest of my thesis evaluation committee members Dr. Albert Oliveras and Dr. Antoni Gasull for their encouragement and insightful comments.

I would like to sincerely thank the coordinators of the Erasmus Mundus scholarship which supported my studies and living for two years of this Masters.

I would also like to thank all the Professors who offered me courses during my masters studies and helped me to increase the knowledge in the signal processing and related areas.

I also thank Mattia Bosio, Phd student at the Signal Theory and Communications (TSC) department, for his support and guidance throughout my master thesis.

Finally, I would like to thank my family and friends for all their support throughout my studies at Universitat Politecnica de Catalonia.
Histom gram based Hierarchical Data Representation for Microarray Classification

Sandeep Kottath

Masters Student,
European Master of Research on Information and Communication Technologies,
Signal Processing and Communication Department
Universitat Politecnica de Catalonia
Barcelona, Spain

ABSTRACT

A general framework for microarray classification relying on histogram based hierarchical clustering is proposed in this work. It produces precise and reliable classifiers based on a two-step approach. In the first step, the feature set is enhanced by histogram based features corresponding to each cluster produced via hierarchical clustering, where a parameter (maximum number of dominant genes) can be tuned based on the dataset characteristics. In the second step, a reliable classifier is built from a wrapper feature selection process called Improved Sequential Floating Forward Selection (IFFS) to properly choose a small feature set for the classification task. Considering the sample scarcity in the microarray datasets, a reliability parameter has been considered to improve the feature selection process along with classification error rate. Different combinations of error rate and reliability has been used as the scoring rule. Linear Discriminant Analysis (LDA) and K-Nearest Neighbour (KNN) classifiers have been used for this work and the performances has been compared. The potential of the proposed framework has been evaluated with three publicly available datasets: colon, lymphoma and leukaemia. The experimental results have confirmed the usefulness of the histogram based hierarchical clustering and the new representative feature generation algorithm. A gene level analysis has revealed that the best features selected by the feature selection algorithm has only very few basic constituent genes involved. The comparative results showed that the proposed framework can compete with state of the art alternatives.

keywords: Microarray classification, histogram, metagenes, dominant genes, hierarchical clustering, EMD, feature selection, LDA, KNN, wrapper.
# Table of Contents

1 Introduction ......................................................... 1
   1.1 Background .................................................... 1
   1.2 State of the Art and Objectives ................................ 2
   1.3 Organization of the Thesis ..................................... 5

2 Feature Set Enhancement ............................................ 6
   2.1 Feature Representation ......................................... 7
   2.2 Distance between Features ..................................... 8
   2.3 Hierarchical Clustering Algorithm ............................. 10
       2.3.1 Treelets Clustering ....................................... 11
       2.3.2 Euclidean Clustering ...................................... 14
   2.4 Toy Example ................................................... 15

3 Feature Selection Process ......................................... 18
   3.1 Feature Selection Algorithm ................................... 19
   3.2 Classification Algorithm ...................................... 21
       3.2.1 Linear Discriminant Analysis ............................. 22
       3.2.2 K Nearest Neighbour ..................................... 24
   3.3 Feature Ranking Criterion ..................................... 26
       3.3.1 Reliability Criteria ........................................ 27
3.3.2 Scoring Rule .................................................. 30

4 Experimental Protocol ............................................ 33
  4.1 Datasets .......................................................... 33
  4.2 Experimental Protocol ......................................... 34

5 Results and Analysis ................................................ 36
  5.1 Performance analysis of the proposed Hierarchical Clustering . .. 38
  5.2 Performance analysis with LDA and KNN ......................... 45
  5.3 Generalization of the Results and Analysis ...................... 49
  5.4 Comparison of the proposed clustering with State of the art ....... 53

6 Conclusions and Future Work ...................................... 55

Bibliography .......................................................... 57
List of Figures

2.1 Hierarchical Clustering Algorithm ........................................ 12
2.2 Generation Rule for the Representative Features .................. 13
2.3 Example to illustrate the merging algorithm to create metagenes ... 14
2.4 Hierarchical Clustering tree for the toy Example .................. 17

3.1 IFFS Algorithm ................................................................. 20
3.2 Example of classification using LDA for two dimensional samples ... 24
3.3 Example of KNN classification with K=3 and K=5 .................. 26
3.4 Example of how the reliability parameter discriminates between two classifiers with zero error rate ................................................. 29
3.5 Score surface in the error-reliability space for lexicographic sorting and exponential penalization ................................. 31

5.1 Error rate for LDA classifier with euclidean and correlation distance as a function of number of selected features .............................. 41
5.2 Error Rate variation as a function of number of features with different values of K in KNN .................................................. 42
5.3 Error Rate variation as a function of number of dominant genes for a fixed number of features (Colon Dataset) ...................... 44
5.4 The scatter plot of the KNN classifier with best two features selected ... 46
5.5 The scatter plot of the LDA classifier with best two features selected ... 47
5.6 Error Rate variation as a function of number of dominant genes for a fixed number of features (Lymphoma Dataset) .................. 51
5.7 Error Rate variation as a function of number of features for M=1 and M=4 dominant genes of Leukemia Dataset .......................... 53
List of Tables

4.1 Database Characteristics ........................................... 34
5.1 Performance Comparison of all the hierarchical clustering approaches using Euclidean Distance ........................................... 39
5.2 Performance Comparison of all the hierarchical clustering approaches using Correlation Distance ........................................... 40
5.3 Performance Comparison of all the hierarchical clustering with different number of dominant genes ........................................... 43
5.4 Performance analysis of KNN classifier ........................................... 47
5.5 Performance analysis of KNN classifier for Lymphoma Dataset ........................................... 48
5.6 Performance variation of Lymphoma Dataset with M values and scoring rules ........................................... 50
5.7 Comparison of different clustering approaches with lymphoma dataset ........................................... 51
5.8 Performance evaluation for Leukemia dataset ........................................... 52
5.9 Performance evaluation for Leukemia dataset ........................................... 54
Chapter 1

Introduction

1.1 Background

Deoxyribonucleic acid (DNA) is a nucleic acid containing the genetic instructions used in the development and functioning of all known living organisms (with the exception of RNA viruses). The DNA segments carrying this genetic information are called genes. Within cells DNA is organized into long structures called chromosomes. With only few exceptions, every cell of our body has full chromosome set and identical genes. Only a fraction of these genes are turned on or "expressed", and it is this subset that confers unique properties to each type of cells. "Gene expression" is the term used to describe the transcription of the information contained within the DNA into messenger RNA (mRNA) molecules that are then translated into the proteins that perform most of the critical functions of cells. The proper and harmonious expression of a large number of genes is a critical component of normal growth and development and the maintenance of proper health. Disruptions or changes in gene expression can cause for many disorders and diseases. [1]

Two complementary advances, one in knowledge and one in technology, have been greatly facilitating the study of gene expression and the discovery of the roles played by specific genes in the development of disease. As a result of the Human Genome Project [2], there has been an explosion in the amount of information available about the DNA sequence of the human genome. Consequently, researchers have identified a large number of novel genes within these previously unknown sequences. The challenge currently facing scientists is to find a way to organize and catalogue this vast amount of information into a usable form. Only
after all the functions of the new genes are discovered, full impact of the Human Genome Project will be realized.

The second advance may facilitate the identification and classification of this DNA sequence information and the assignment of functions to these new genes: the emergence of DNA microarray technology. A microarray works by exploiting the ability of a given mRNA molecule to bind specifically to, or hybridize to, the DNA template from which it originated. By using an array containing many DNA samples, scientists can determine, in a single experiment, the expression levels of thousands of genes within a cell by measuring the amount of mRNA bound to each site on the array. With the aid of a computer, the amount of mRNA bound to the spots on the microarray is precisely measured, generating a profile of gene expression in the cell or for many cells. The whole process of a microarray chip generation has been illustrated with animation in [3]. There are two general methods for making gene expression microarrays [4]: one is to hybridize a single test set of labeled targets to the probe, and measure the background subtracted intensity at each probe site; the other is to hybridize both a test and a reference set of differentially labeled targets to a single detector array, and measure the ratio of the background-subtracted intensities at each probe site. The intensity based microarray datasets are used for this work.

1.2 State of the Art and Objectives

Since the mid-1990s, the field of Genomic Signal Processing (GSP) has exploded due to the development of DNA microarray technology. The vast amount of raw gene expression data leads to statistical and analytical challenges including the classification of the dataset into correct classes. The goal of classification is to identify the differentially expressed genes that may be used to predict class membership for new samples. An example of a classification task is to distinguish samples having colon cancer from the samples that doesn’t have colon cancer. In this case, the samples which has been known to have colon cancer will be assigned
to one class and the other samples will be assigned to another class. The central difficulties in
microarray classification are the availability of a very small number of samples in comparison
with the number of genes in the sample, and the experimental variation in measured gene
expression levels. The classification of gene expression data samples involves feature selection
and classifier design. Feature selection identifies the subset of differentially-expressed genes
that are potentially relevant for distinguishing the classes of samples.

Microarray Classification [5] is a very challenging task because of many reasons. Huge
dimensionality of feature set (thousands of gene expressions) is the primary reason. While
data matrices in most traditional applications, such as clinical studies, are "wide"
matries, with the number of cases exceeding the number of variables, in microarray studies,
data matrices are "tall", with the number of variables far exceeding the number of cases. This
situation is a typical example of sample scarcity, and it is commonly addressed as curse
of dimensionality[6]. Another reason is that, there is lack of known data structure [7] : no
a-priory relationship exists from the geometrical proximity of two expression profiles in an
array. This characteristic limits the applicability of signal processing techniques such as
wavelet filtering or other filtering techniques which assumes an underlying data structure.
On the other hand, it is well known that genes are highly interrelated, depending on the
involved regulatory biological process. Therefore, the data themselves do respond to a hidden
regulating structure that, if found, can be useful for many applications. Also, microarray
gene expression data are commonly perceived as being extremely noisy because of many
imperfections inherent in the current technology. Even if noise has been reduced by the
application of sophisticated signal processing algorithms and normalization techniques, it still
corrupts the actual values. Being able to discern the actual value from the measurement
noise is considered by many researchers a compelling problem [8]. A plethora of algorithms
has been presented in the literature to address the classification problems and produce reliable

In the proposed framework, three main classification issues (high feature number,
noise and lack of structure) have been addressed by a two step approach. The first phase tries to infer a structure of the microarray data using a histogram based hierarchical algorithm. A new feature set is created through this hierarchical clustering and has been used to enrich the original feature set. The objective of this first step is to give a structure to the microarray data, and hence to produce new features summarizing common traits of gene clusters. By combining or ”clustering” genes which has similar expression profiles, we can identify group of genes that share some common functionality [12]. This is useful to make meaningful biological inferences about the set of genes. It also helps to infer the functionality of some genes which has been found similar to another known gene in terms of expression profile. The notion of a ”cluster” varies between algorithms and is one of the many decisions to take when choosing the appropriate algorithm for a particular problem.

In a related work [7], each cluster in the hierarchical clustering has been represented by just one summarized feature called metagene, which is created by a linear combination of original genes. In this work, a gene cluster is represented by more than one feature. Here dominant features of a gene cluster are represented by few number of representative genes or metagenes. Also, in this work, the importance of any metagene is represented by the number of constituent genes. Hence, any cluster will be represented by a histogram (representative features and their occurrences). Hierarchical clustering finds the pair of clusters that are most similar, joins them together, and then identifies the next most similar pair of clusters. This process continues until all of the genes are joined into one giant cluster. From all the levels of hierarchy, the most relevant clusters will be found during the feature selection for the classifier.

The second step in the proposed framework is the development of an effective feature selection method to select the best set of features from the enriched feature set, with which the final classifier is trained. The classifier must be able to catch the key features that differentiate between sample classes. Designing such sparse classification approaches has significance in the biological point of view, since mechanisms that lead to specific diseases are
thought to involve relatively small numbers of genes. In addition, the proposed framework might help to find a second choice classifier in case if the chosen features are unavailable for further clinical testing, for example due to economic reasons. A suitable wrapper feature selection algorithm called improved sequential feature selection (IFFS) [13] has been chosen in this work. As a wrapper, it can capture multi-variate relations, and being deterministic, the results hold if the initial conditions are unaltered. In the feature selection task, a scoring rule based on the combination of reliability parameter and the error rate has been used to evaluate the predictive performance of a single classifier[7]. It helps to deal with situations where the error rate may not be precise enough due to sample scarcity of microarray data. The error rate and reliability parameter are combined into a scalar value (score) representative of the classification performance.

1.3 Organization of the Thesis

This report is organised as follows. In Chapter 2, the hierarchical clustering algorithm developed in this work has been presented. The feature selection algorithm used along with the classifiers used are described in Chapter 3. The potential of the proposed framework have been evaluated on three publicly available databases, whose characteristics and references, along with the experimental protocol is described in Chapter 4. Chapter 5 presents the results of the experiments and its analysis. Results are also compared with the state of the art approaches in the microarray based cancer classification area. The main conclusions of this project and the lines of future work are discussed in Chapter 6.
Chapter 2
Feature Set Enhancement

Feature Set enhancement is the first step in the proposed framework. The aim of this phase is to infer a hierarchical structure from the original data using a hierarchical clustering method. The newly created feature set expands the feature space and can improve the classification accuracy. In order to infer a hierarchical structure from the data, a hierarchical clustering algorithm similar to the work in [14][7] is used along with a novel idea of keeping representative features in the form of histogram for each cluster at different levels of the hierarchy. In addition to finding a structure on the data, it also summarizes different hierarchical clusters with few representative features.

The feature set enhancement step focuses on extracting new variables from the gene expression values and to get information on relationships among different gene clusters. This strategy has already been adopted by algorithms like Tree Harvesting[15], Pelora[16] and [7], where the benefit of hierarchical clustering to extract interesting new variables to enhance the original feature set is highlighted. The approach used in this work to summarize the gene clusters, preserves some significant details along with the summarized feature (metagene). Summarizing gene clusters with few representative features has many advantages. First, it gives an easy and compressed representation using linear combination of original feature set. It also gives a flexibility of choosing the amount of information needed to represent each gene cluster, since the number representative features is a variable that can be chosen iteratively. Another advantage is the residual noise reduction due to averaging of most similar features, which is more profound at the lower levels of the hierarchical tree.

For creating a hierarchical structure of the genes, an aggregation rule (similarity
metric of the clusters at different levels) and a generation rule (for creating the representative genes for each cluster) are required. In this work, the chosen clustering process is a bottom-up, pairwise hierarchical clustering based on Lee’s work in [14], where an adaptive method for multi-scale representation of data called *Treelets* is presented. The following sections will describe the important concepts behind the proposed framework and the algorithm.

### 2.1 Feature Representation

A microarray gene expression is represented by \( \{g_1, g_2, \ldots, g_p\} \), where \( g_i \) is an n-dimensional vector containing gene expressions corresponding to all the n samples (or observations) for the \( i^{th} \) gene and \( p \) is the number of genes in the microarray data. Throughout this report, a vector will be either represented by bold letter or by an underlined letter.

In the proposed hierarchical clustering approach, each gene cluster is represented by representative features. Let the maximum number of such features for each cluster be \( M \), which could take any value from 1 (same as [7]) to \( p \). The algorithm will find the best \( M' \) features (\( M' \leq M \)) that can summarize each cluster in the hierarchical tree. There are two cases to be considered here. In the initial levels of the hierarchical clustering, when the number of genes in the cluster is less than \( M \), all the genes are used to represent the cluster. But when the number of genes exceeds \( M \), the algorithm will merge the most similar genes in the cluster to create a "metagene" until the number of represented features equals \( M \). The importance of any metagene depends on the number of constituent genes. Hence, any cluster will be represented by a histogram (representative features and their occurrences) or signature.

A histogram or signature \( S \) is defined as,

\[
S = (f_1, w_1), (f_2, w_2), \ldots, (f_M, w_{M'}); M' \leq M
\] (2.1)
where $f_i \in \{g_1, g_2, ..., g_p\}$ and $w_i$ is the number of occurrences or significance of the corresponding gene expression vector.

A feature in this work refers to any set of gene/metagene expressions in corresponding to any cluster in the hierarchical clustering process. The occurrences play a very important role in the metagene generation, but has been discarded for the classification stage. It will also be interesting to consider options to include this occurrences in some ways to check if there is any performance enhancement, which is not in the scope of this master thesis. This could be considered in the future work plans in the same lines of master thesis.

### 2.2 Distance between Features

Now that we have a histogram based representation of the clusters of the hierarchical clustering, it is important to define the similarity measure between these clusters to determine which clusters has to be merged at each level of the algorithm. Earth Mover Distance (EMD)[17] has been considered in this work since it naturally extends the notion of a distance between features to that of a distance between distributions of features.

Defining a distance between two distributions requires a notion of distance between features in the underlying domain defining the distributions. This distance is called the ground distance. Two different ground distances, Correlation based distance and Euclidean Distance are considered in this work. EMD is a natural and intuitive metric between histograms if we think of them as piles of sand sitting on the ground (underlying domain). Each grain of sand is an observed feature. To quantify the difference between two distributions, we measure how far the grains of sand have to be moved so that the two distributions coincide exactly. EMD is the minimal total ground distance travelled weighted by the amount of sand moved (called flow). If the ground distance is a metric, EMD is a metric as well. There are several advantages of using EMD over other distribution dissimilarity measures. For example, it does not suffer from arbitrary quantization problems due to rigid binning strategies.
Let \( P \) and \( Q \) be two histograms (or signatures) of the form,

\[
P = (p_1, w_{p1}), (p_2, w_{p2}), \ldots, (p_m, w_{pm}) \quad (2.2)
\]

\[
Q = (q_1, w_{q1}), (q_2, w_{q2}), \ldots, (q_n, w_{qn}) \quad (2.3)
\]

and let \( D = [d_{ij}] \) be the ground distance matrix, with each element \( d_{ij} \) represents
the ground distance between the features \( p_i \) and \( q_j \). We want to find a flow, \( F = [f_{ij}] \), where
\( f_{ij} \) is the flow between the feature points \( p_i \) and \( q_j \) (remember that each feature is a point in
\( n \)-dimensional space), that minimizes the overall cost defined in the equation (2.4)

\[
W(P, Q, F) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} \cdot d_{ij} \quad (2.4)
\]

The flow \( f_{ij} \) must satisfy the following linear constraints:

\[
f_{ij} \geq 0, \quad \forall i \in 1, \ldots, m, j \in 1, \ldots, n \quad (2.5)
\]

\[
\sum_{j=1}^{n} f_{ij} \leq \omega_{pi} \quad \forall i \in 1, \ldots, m \quad (2.6)
\]

\[
\sum_{i=1}^{m} f_{ij} \leq \omega_{qj} \quad \forall j \in 1, \ldots, n \quad (2.7)
\]

\[
\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = \min(\sum_{i=1}^{m} \omega_{pi}, \sum_{j=1}^{n} \omega_{qj}) \quad (2.8)
\]

Constraint 2.5 limits the flow to move from \( P \) to \( Q \) and not the other way around.
Constraints 2.6 and 2.7 make sure that no more than available probability is displaced from
its original feature. Constraint (2.8) sets a lower bound on the flow. Let \([f^*_{ij}]\) be the optimal flow that minimizer Equation 2.4. Then, the Earth Mover Distance is defined to be,

\[
EMD(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f^*_{ij} \cdot d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f^*_{ij}}
\] (2.9)

### 2.3 Hierarchical Clustering Algorithm

The feature set enhancement is based on a bottom-up, pairwise hierarchical clustering algorithm whose general pseudocode is outlined in 2.1. This algorithm is variation of the the Treelets algorithm proposed in [14]. Treelets is an iterative process in which, at each level, the two most similar features are replaced by two newly created features, a course grained approximation feature and a residual detail feature. Such methods outputs a multi-scale representation of the original data allowing a perfect reconstruction of the original signal. In our case, the purpose of the algorithm is completely different since it is not required to have any perfect reconstruction. In this work, the purpose of clustering is to find the new set of features that will facilitate the class separability. So, the aim of the hierarchical clustering is to find the representative features (in the form of histogram as explained in Section 2.1) for gene clusters and to generate a hierarchical tree structure. To this end, at each level, M representative features (M-dimensional histogram) for the union of two most similar clusters are found, using the pseudocode outlined in Figure 2.2. Afterwards, the newly created histogram is used as a feature to be compared in the next iterations. As outlined in Figure 2.1, two main elements defining the final output are the similarity metric \(d(f_a, f_b)\) and the generation rule for representative features \(g(f_a, f_b)\). In this work, negative of the earth mover distance (EMD) has been used as the similarity metric. Ground distance for the EMD can be either Euclidean or Correlation distance, leading to two variants of the clustering algorithm as explained in Section 2.3.1 and Section 2.3.2.

As outlined in Figure 2.2, representative feature generation for a cluster in the
hierarchical tree is an iterative process. At first, the union of the two histograms of the child clusters is found. Then, if the number of gene expressions in the union exceeds M (maximum number of dominant genes allowed), the most similar gene expressions or metagenes are combined using weighted averaging using their occurrences. This weighted averaging will give more importance to the any metagene that has more constituent genes. The occurrence for the new metagene is calculated by adding the occurrences of the child genes/metagenes. When two or more genes are constituent in a metagene, the occurrence or significance will be increased accordingly. Occurrences play a very important role in the metagene generation by assigning higher weight to the child metagene that has more constituent genes.

Since the features in the new clustering takes the form of histograms, the hierarchical clustering algorithm can also be interpreted as an algorithm to find histogram of the original microarray data at different scales or resolution. In the Figure 2.3, the merging algorithm is illustrated through a toy example. Assuming that there are five n-dimensional gene expressions in the given cluster represented by \( f_1, f_2, \ldots, f_5 \) after taking the union of two most similar child clusters. For the ease of understanding, all the features are marked in the x-axis according to their relative n-dimensional euclidean distance. The Figure 2.3, from top to bottom, shows the merging process until just one metagene is remaining. But ideally the algorithm stops whenever the number of representative features reaches desired number(M).

2.3.1 Treelets Clustering

Treelet Clustering owes its name to the Treelets algorithm from [14]. In this type of clustering, the ground distance between features used in the earth mover distance (EMD) computation is the negative of the Pearson correlation. Pearson correlation denotes the normalized correlation between two features in the histogram, and is defined for two generic feature vectors \( f_a \) and \( f_b \):

\[
d_g(f_a, f_b) = \frac{\langle f_a, f_b \rangle}{||f_a||_2 \cdot ||f_b||_2}
\]  

(2.10)
Hierarchical clustering algorithm

Original feature set : $G_0 = \{g_1, \ldots, g_p\}$

New Feature Set = $F_{new} = \phi$

Metagene Set : $M = \phi$

Active feature Set : $F_0 = \{(g_1, 1), \ldots, (g_p, 1)\} = \{f_1, \ldots, f_p\}$

where, $f_k$ in general takes the form , $\{(f_{k1}, n_1), \ldots, (f_{kM'}, n_{M'})\}$ ; $M' \leq M$

where $f_{ij} \in \{G_0 \cup M\}$ and $M =$ Maximum number of dominant genes.

Initiate $F = F_0$

For $i = 1 : p-1$

1. Calculate the pairwise similarity metric $d(f_A, f_B)$ for all features in $F$

2. Find $A, B : d(f_A, f_B) = \max(d(., .))$.

3. New Feature Set $f_{new} = g(f_A \cup f_B)$ where $g$ is the generation rule explained in Figure 2.2.

4. Add new feature to the active feature set, $F := F \cup \{f_{new}\}$

5. Remove the two features from the active feature set. $F := F \setminus \{f_A, f_B\}$

6. Join the new feature to the New Feature Set. $F_{new} := F_{new} \cup \{f_{new}\}$

end

Fig. 2.1: Hierarchical Clustering Algorithm
Assuming that the hierarchical clustering algorithm selects the two clusters represented by $f_A$ and $f_B$ where,

$$f_A = f_i = \{ (f_{i1}, n_{i1}), \ldots, (f_{iM'}, n_{iM'}) \}$$

$$f_B = f_j = \{ (f_{j1}, n_{j1}), \ldots, (f_{jM''}, n_{jM''}) \}$$

The algorithm will find the union of the two histograms,

$$f_{\text{new}} = \{ (f_{i1}, n_{i1}), \ldots, (f_{iM'}, n_{iM'}), (f_{j1}, n_{j1}), \ldots, (f_{jM''}, n_{jM''}) \}$$

where, $1 \leq M', M'' \leq M$

Now the algorithm enters an iterative process to merge the most similar gene/metagene expressions until the dimension of histogram becomes $M$.

Initiate $f = f_{\text{new}}$

For $m = 1 : M' + M'' - M$

1. Calculate the pairwise similarity metric of all the gene expressions in the new histogram.
2. Find $a, b : d(f_a, f_b) = \max(d(., .))$.
3. Metagene Generation and its corresponding number of occurrences :

$$f_{\text{new}} = \begin{pmatrix} \frac{n_a \cdot f_a + n_b \cdot f_b}{n_a + n_b} \end{pmatrix} ; n_{\text{new}} = n_a + n_b$$

4. Remove the two gene expressions and its occurrences from the active feature set.

$$f := f \setminus \{ (f_a, n_a), (f_b, n_b) \}$$

5. Join the new metagene and its occurrence to the histogram.

$$f := f \cup \{ (f_{\text{new}}, n_{\text{new}}) \}$$

6. Update the Metagene set $M$ with the new metagene created $(f_{\text{new}})$.

end

Fig. 2.2: Generation Rule for the Representative Features
2.3.2 Euclidean Clustering

In this type of hierarchical clustering, the ground distance used in the earth mover distance (EMD) is the Euclidean distance as defined in Equation 2.11. Euclidean distance captures the point wise closeness rather than the profile shape similarity.

\[ d_g(f_a, f_b) = \|f_a - f_b\|_2 \]  \hspace{1cm} (2.11)
2.4 Toy Example

Let us consider a toy example to analyse the hierarchical clustering algorithm with Euclidean Clustering. Assume a microarray data having gene expressions of 8 genes corresponding to 4 observations as shown in Equation 2.12.

\[
\{g_1, g_2, \ldots, g_9\} = \begin{pmatrix}
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\end{pmatrix}
\] (2.12)

Let the number of dominant genes, \(M = 3\). Note that the gene expressions \(g_1 = g_7\) and \(g_2 = g_8\). Hence, we expect the hierarchical clustering algorithm to group these genes in the first two levels and then the most similar genes in the subsequent levels. In the beginning of the hierarchical clustering algorithm, there will be 9 clusters, each corresponding to each of the gene expressions and having number of occurrence 1 each as it can been seen in Figure 2.4. In the case of EMD, since it uses the normalized occurrences or probability, for example, \(\{(g_1, 1)\}\) and \(\{(g_1, 2)\}\) will have emd value 0.

The hierarchical clustering tree for the toy example is shown in the Figure 2.4. As the algorithm proceeds up in the hierarchy, at some levels, when it merge two most similar histograms, the number of genes in the new node will exceed \(M\) (here \(M=3\)). When the number of genes exceeds \(M\), the most similar genes in the cluster are merged iteratively till the number of genes reaches \(M\). As it can be seen in the Figure, there are new gene indexes (compared to only 9 initial genes). Those indices correspond to the metagene created from the most similar genes of the cluster in the merging algorithm, when the number of features exceeds \(M\).

To illustrate the metagene creation, let us analyse the merging of two nodes \(\{(g_2, 2), (g_4, 1), (g_9, 1)\}\) and \(\{(g_3, 1), (g_5, 1)\}\) in the tree shown in Figure 2.4.
1. The union of the two child node histograms are found.

\[ f_i = \{(g_2, 2), (g_4, 1), (g_9, 1)\} \]

\[ f_j = \{(g_3, 1), (g_5, 1)\} \quad (2.13) \]

\[ f_{\text{new}} = \{(g_2, 2), (g_4, 1), (g_9, 1), (g_3, 1), (g_5, 1)\} \]

2. Now the algorithm enters an iterative process till the number of gene expressions in the new histogram equals 3.

3. It can be found that the most similar gene expressions in the new histogram are \(g_3\) and \(g_5\). Hence we will merge these two genes to create a metagene.

\[ g_{10} = \frac{1 \cdot g_3 + 1 \cdot g_5}{1+1} = \begin{pmatrix} 0 \\ 0.5 \\ 1 \\ 1 \end{pmatrix} \quad (2.14) \]

The new metagene will be assigned a number of occurrence which is equal to the sum of the number of occurrences of the child genes (1+1=2). Substituting the actual values in the above expression, the new feature and its occurrence will be \((g_{10}, 2)\)

4. Now the new histogram will be of the form,

\[ f_{\text{new}} = \{(g_2, 2), (g_4, 1), (g_9, 1), (g_{10}, 2)\} \quad (2.15) \]

5. Since it still has more than 3 gene expressions, the algorithm will find the next most similar gene expressions (even considering the metagene as a gene expression). It can be found that \(g_9\) and \(g_2\) are the next most similar expressions. Hence we will merge
\((g_9, 1)\) and \((g_2, 2)\)

\[
g_{11} = \frac{1 \cdot g_9 + 2 \cdot g_2}{1 + 2} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0.33 \end{pmatrix}
\] (2.16)

The new metagene \(g_{11}\) will be assigned a number of occurrence which is equal to the sum of the number of occurrences of the child genes \((1+2=3)\). Substituting the actual values in the above expression, the new feature and its occurrence will be \((g_{11}, 3)\)

6. Now the new histogram will be of the form,

\[
f_{\text{new}} = \{(g_4, 1), (g_{10}, 2), (g_{11}, 3)\}
\] (2.17)

Since it has only 3 gene expressions, we stop the iteration here and assign this as the histogram (or representative feature set) of this particular cluster of genes.

**Fig. 2.4: Hierarchical Clustering tree for the toy Example**
CHAPTER 3

Feature Selection Process

The hierarchical clustering algorithm has enhanced the feature set by representative features of new clusters of genes. Now the objective is to select the best features from the pool of enhanced feature set. The feature selection algorithm considers each gene cluster of the hierarchical clustering algorithm as features and will select the best features that can improve the classification performance.

As it has already been explained in Section 2.1, a feature in this work refers to any set of gene/metagene expressions in corresponding to any cluster in the histogram based hierarchical clustering process. The occurrences play a very important role in the metagene generation, but has been discarded for the classification stage. In the future work of this project, it would be interesting to investigate on the usage of number of occurrences to scale each dimension while calculating euclidean distance in the case of KNN(K Nearest Neighbour) based classifier. It needs further detailed analysis and is out of scope of this master thesis.

The feature selection algorithms in the literature can be divided into three main classes: filters, wrappers and embedded methods. In Filters, the feature selection task is disconnected from the learning phase. Typically, a score is assigned to each feature based on univariate criteria and the features with best scores are selected. Examples of filter based feature selection are the Student t-test [11]. Filter implementation is faster but limits the discovery of multivariate interactions. Embedded methods, instead, can capture multivariate interactions, but the implementation is specific for specific classifiers. An example of embedded is the recursive Ridge regression [18]. Wrappers, on the other hand, include a classifier evaluation wrapped in the feature selection process, for any general classifier. A
search through all the feature subsets is performed by applying the learning algorithm (or classifier) to evaluate each subset [19]. Wrapper feature selection has the ability to capture multivariate interactions at the expense of higher computational cost.

3.1 Feature Selection Algorithm

The feature selection algorithm proposed in [7][20] has been used for this thesis with a new classifier and a reliability parameter for the new classifier based feature selection. Most part of the feature selection algorithm relies on the work [7] and [20]. The similar feature selection algorithm will also enable performance comparison of the proposed hierarchical clustering algorithm with the one proposed in [7] easier.

A wrapper feature selection algorithm called Improved Sequential Floating Forward Selection (IFFS) [13] has been used here since it has ability to capture multivariate interactions and can be used with virtually any classifier. It is an advanced version of the Sequential Floating Forward Selection (SFFS) [21] with a replacing step to the original algorithm. The flowchart of the IFFS algorithm is illustrated in the Figure 3.1. This algorithm uses the representative genes and metagenes of any selected cluster as a single feature. The algorithm starts with an empty feature set and the search ends when a threshold value is reached. The threshold is the desired feature set size or the maximum number of iterations to avoid perpetuating an infinite loop. Maximum limit on the number of iterations also helps in the case if the algorithm could not find any more features that improves the performance of the classifier. In the initialization block, the cardinality of the chosen feature set is made to zero and a new search is begun to find the best features. After the initialization, the process enters a loop of tasks.

1. Add Phase: Here, the best feature to add to the current set is chosen. The algorithm tests all the possible features that have not yet been selected one by one. The test implies the expansion of the current feature set with a new feature, then a classifier is trained
Fig. 3.1: IFFS Algorithm
and the corresponding classification score $J(r)$ is calculated. After a comprehensive search, the feature obtaining the best $J(r)$ score is included to the current feature set.

2. **Backtracking Phase**: In this phase, the possibility of reducing the dimension of current feature set by one is evaluated. To this end, one feature at a time is removed from the current set and the classification performance $J(r)$ is evaluated. The next step is to identify the weakest feature in the current set, which is the feature whose removal implies the minimum performance loss, or the maximum performance gain. Once this feature is identified, it is decided whether to eliminate it or not. If the elimination implies no improvement in terms of $J(r)$ score, the feature is maintained in the current set and the algorithm goes to the Replacing phase. Otherwise the feature is removed and another Back-tracking phase is performed.

3. **Replacing Phase**: In this step, the possibility to replace one feature from the current set is considered. One feature at a time is removed from the current set and then an Add phase is performed on the reduced set to find which is the best substitute for the eliminated feature. All the substitutions are then ranked and if the best one proves to be useful (i.e. the obtained $J(r)$ value with the substitution is better than without), the current set is updated and a Backtracking phase is performed. Otherwise, the subset keeps unchanged and the algorithm goes to a new Add phase.

### 3.2 Classification Algorithm

This is the core block of the feature selection algorithm. The accuracy of training the classifier and selecting the best parameters can yield very good performance results for the microarray classification task. K-Nearest Neighbour (KNN) and Linear Discriminant Analysis (LDA) are the two classifiers used for this work. The performance of the feature set enhancement phase is evaluated with these two classifier outcomes.
In the proposed framework, the feature selection algorithm selects some set of gene clusters (and hence the corresponding representative features and their occurrences) for the evaluation of its classification efficiency. Let one observation (or sample) of the microarray data with the set of all representative features (genes and metagenes from all the selected clusters) be represented as $\vec{f}$. Assume that we have $n$ samples of the microarray experiment. Each of these samples or observations will have a class assigned (for eg. class 1 : with cancer, class 2 : without cancer). Hence, each vector $\vec{f}$ will have a class assigned. We will divide the these set of vectors $\vec{f}$ as training set and test set. We train the classifier with the training set with known class information. And the test set is used for performance evaluation. The aim of the classification algorithm is to find the best way to classify a test sample $\vec{f}_t$ to one of these classes, learning from the training samples (with known classes).

### 3.2.1 Linear Discriminant Analysis

The Linear Discriminant Analysis (LDA) are methods used in pattern recognition and machine learning to find a linear combination of features which characterizes or separates two or more classes of objects or events. In the case of microarray classification scheme proposed here, the purpose is to find the best linear combination of the representative features selected by the IFFS algorithm that can separate the two classes of the microarray samples. This classifier has been selected in this work due to its advantages like simplicity, interpretability and precision [5] [22].

Consider a set of features $\vec{f}$ for each sample of gene expression with known class $y$. This set of samples is called the training set. Here we are considering two class problem, when $y$ takes values 1 and 0 (representing class 1 and class 2). The classification problem is then to find a good predictor for the class $y$ of any sample of the same distribution given a new observation $\vec{f}_t$.

The approach used by LDA can be illustrated through a simple two-class example.
here. The two classes of observations or samples are considered to be part of two distributions. Suppose that, the two classes of observations have means $\vec{\mu}_{y=0}$, $\vec{\mu}_{y=1}$ and covariances $\Sigma_{y=0}, \Sigma_{y=1}$. Then, any linear combination of the features $\vec{\omega} \cdot \vec{f}$ will have means $\vec{\omega} \cdot \vec{\mu}_{y=i}$ and variance $\vec{\omega}^T \cdot \Sigma_{y=i} \cdot \vec{\omega}$ for $i = 0, 1$. The separation between these two distributions is defined in Equation 3.1 as the ratio of inter-class variance to the intra-class variance.

$$S = \frac{\sigma^2_{\text{inter}}}{\sigma^2_{\text{intra}}} = \frac{(\vec{\omega} \cdot (\vec{\mu}_{y=1} - \vec{\mu}_{y=0}))^2}{\vec{\omega}^T (\Sigma_{y=0} + \Sigma_{y=1}) \vec{\omega}} \quad (3.1)$$

It can be shown that the maximum separation occurs when,

$$\vec{\omega} = (\Sigma_{y=0} + \Sigma_{y=1})^{-1}(\vec{\mu}_{y=1} - \vec{\mu}_{y=0}) \quad (3.2)$$

The hyperplane that best separates the two classes is the one that is normal to the vector $\vec{\omega}$. In the case of m-dimensional samples, the hyperplane is represented by an equation with $(m - 1)$ dimensions. An example to illustrate this concept is given below.

Consider that the feature selection algorithm selects only one cluster represented by two gene/metagene expressions. Hence, each of the samples $\vec{f}$ can be represented in two dimensional plane. Let the two dimensions be represented by $x_1$ and $x_2$ (or $f_1$ and $f_2$). Here the dimensionality of the classification problem $m = 2$. Hence, the LDA classifier will find the best hyperplane with dimension $(m - 1)$ (here it is equal to 1, meaning it is a line, perpendicular to $\vec{\omega}$) that if all samples are projected, will make the classification easier. Figure 3.2 shows the two cases of projecting the two dimensional samples to different lines. In Fig.(a), the two classes are not well separated when projected on to the line. But in Fig.(b), the line succeeded in separating the two classes and in the meantime reducing the dimensionality of the problem from 2 features ($x_1, x_2$) to only a scalar value (one dimension) that separates the line. This example shows the importance of selecting the best line for projecting the samples. As it has been already mentioned, the $\vec{\omega}$ will correspond to the normal to the projection lines.
in the figure.

Fig. 3.2: Example of classification using LDA for two dimensional samples

3.2.2 K Nearest Neighbour

The k-nearest neighbour (KNN) algorithm is amongst the simplest of all machine learning algorithms but still one of the most preferred classifier for the microarray classification task, since it is able to model any non-linear interactions among the features. In a KNN based classifier, a sample is classified by a majority vote of its neighbours, with the sample being assigned to the class most common amongst its k nearest neighbours (k is a positive integer, typically small). Nearness is defined based on some distance metric, euclidean and correlation distances are used in this work. It has been suggested in [23] to use higher values of k to make more reliable classifier. It is also recommended to use a k value less than the square root of the number of samples, defining an upper bound due to the sample scarcity of microarray experiments. In this work, we will consider a range of values of k in the classification algorithm to analyse the performance. The KNN algorithm has some strong consistency results. As the amount of data approaches infinity, the algorithm is guaranteed to yield an error rate no worse than twice the Bayes error rate [24](the minimum achievable error rate given the
distribution of the data). KNN is guaranteed to approach the Bayes error rate, for some value of k (where k increases as a function of the number of data points).

Along with all the advantages mentioned above, the use of distance metric in KNN classifier has also motivated for its usage in this work. Since any cluster in the hierarchical clustering has a histogram based representation, the number of occurrences of each multidimensional histogram could possibly be used as a scaling factor for the distance metric. This requires a thorough investigation along these lines. Due to time restrictions of this master thesis, it will not be addressed in this work, but will be addressed in the future work of this project.

As already discussed before, any training samples with the selected features can be represented as $\vec{f}$. Each of this training examples are vectors in a multidimensional feature space with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the classification phase, k is a user-defined constant, and an unlabelled vector $\vec{f}_t$ is classified by assigning the label which is most frequent among the k training samples nearest to that query point. Correlation distance and Euclidean distance are used as distance metric to define the "nearness" in this work. Figure 3.3 shows an example of KNN classification with only two selected features, hence two-dimensional classification problem.

KNN is an example of non-linear classifier where the classification boundary can take any non-linear shape. Here we have to clearly distinguish a classification boundary with a decision boundary. Classification boundary is any boundary that separated two classes of samples. In the case of KNN with euclidean distance, decision boundary defines the boundary of the area where the classifier applies the majority rule to predict the class of the test sample. The boundary shown in Figure 3.3 is actually a decision boundary. The accuracy of the KNN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. The feature selection algorithm used in this work will try to minimize any such performance degradation.
Fig. 3.3: Example of KNN classification. The new observation or sample (green star) should be classified either to the first class of blue circles or to the second class of red squares. For $k = 3$ it is assigned to the first class because there are 2 circles and only 1 square inside the first decision boundary. If $k = 5$ it is assigned to the second class (3 squares vs. 2 circles inside the outer boundary).

### 3.3 Feature Ranking Criterion

$J(.)$ plays a major role throughout the feature selection process. It is a measure of classifier performance. Since IFFS is a wrapper algorithm, the classifier (LDA or KNN) is applied multiple times and in every case, a $J(.)$ score is extracted. The most popular $J(.)$ score is based on classification error. A reliable estimation of the error rate is obtained when there is sample abundance. But in the case of microarray experiments where the sample scarcity is a problem, different error rate estimation techniques are used, one of which is Cross Validation[10]. In cross validation error estimation, the following steps are performed iteratively:

1. Divide the dataset in two parts, a training set which has majority of the available samples, and an internal validation set (or test set).

2. Train the classifier on the training set.
3. The trained classifier is applied on the internal validation set.

4. Extract the performance result.

After iterating the process N times, the global results are obtained as the average of individual iteration outcomes. An example is 5-fold cross validation. It consists of 5 iterations, in which for each iteration, the internal validation set is composed of 20% of the available samples and the training set consists of remaining 80% samples. Cross-validation is an unbiased estimator of error rate, but in the case of sample scarcity, it can show high estimation variance [5]. In order to obtain more robust error rate, many repeated runs of cross-validations are performed and dataset partition tries to maintain the same class distribution in the training and internal validation set.

Since microarray data has very few samples and huge dimensionality, a $J(.)$ criterion based only on the error rate may not be sufficient in ranking purpose. Since the feature set dimension is comparatively very high, it is common to have group of features with the same error rate, among which only one must be chosen at any phase of the feature selection algorithm. Also, slight error differences can arise due to unfortunate data partition for the cross-validation. Hence a new criterion is also added to the $J(.)$ score, a reliability parameter.

### 3.3.1 Reliability Criteria

The reliability criteria used in this work are different for KNN and LDA classifier. For IFFS based on LDA classifier, the reliability criteria defined in [7] is used. For IFFS based on KNN classifier, a new reliability parameter is defined.

**Reliability Parameter for LDA**

This reliability parameter considers that a feature obtaining well separated class is better than a feature in which the class separation is very less [7]. It can also be related to the
idea of margin in the linear Support Vector Machine (SVM). The reliability parameter, $r$, measures a weighted sum of sample distances from the classification boundary as a goodness estimation. It is calculated on the internal validation set samples and averaged through the cross-validation iterations. For each cross-validation iteration, reliability parameter is defined in 3.3 for a binary classifier. In Equation 3.3, $n_{\text{test}}$ is the number of samples in the test set, $c_l$ is the class label (1 or 2) of the $l^{th}$ sample of the training set, and $p(c_l)$ is the probability of the class $c_l$ in the test set. If we assume a uniform class distribution, $p(c_l)$ will take the value of 0.5. The value $d_l$ is the euclidean distance of the $l^{th}$ sample from the classification boundary, with a positive sign if the $l^{th}$ sample has been classified correctly and a negative sign otherwise. $\hat{\sigma}_d$ is defined in 3.4 and it is an estimate of the intra-class variance of the sample distances from the classification boundary. $n_1$ and $n_2$ are the number of samples in class 1 and class 2 respectively. $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the estimated variances of the sample distances from the classification boundary for class 1 and class 2 respectively. It is obtained as in the independent two-sample t-test with classes of different size and variance.

$$r = \frac{1}{n_{\text{test}}} \cdot \hat{\sigma}_d \cdot \sum_{l=1}^{n_{\text{test}}} \frac{d_l}{p(c_l)}$$ (3.3)

$$\hat{\sigma}_d = \sqrt{\frac{\hat{\sigma}_1}{n_1} + \frac{\hat{\sigma}_2}{n_2}}$$ (3.4)

Now let us focus on the analysis of the Equation 3.3. Division by $\hat{\sigma}_d$ guarantees that $r$ is invariant of the scaling factor, thus $r(f_{n_1}) = r(kf_{n_1}) \quad \forall k \in \mathbb{R}^+$. Division by $p(c_l)$ assigns the same relative weights to each class and it is useful when the class distribution is highly skewed. The reliability value, $r \in (-\infty, \infty)$ is positively influenced by a large mean class separation in the perpendicular direction to the classifier boundary, and by small intra class data variance. On the other hand, it is negatively influenced by the number of errors. An example of how the reliability can discriminate between features with equal error rate is illustrated in Figure 3.4.
Fig. 3.4: Example of how the reliability parameter discriminates between two classifiers with zero error rate

**Reliability Parameter for KNN**

For IFFS based on KNN classifier, the reliability criteria is defined as in equation 3.6. There are many variables in this equation which are same as in Equation 3.3. Only the new variables that are not explained in 3.3 are explained in this section.

For $l^{th}$ sample in the test set, the algorithm assigns a class that has highest occurrence among its K neighbours. Assume that, among K neighbours, $n_1(k)$ is the number of class 1 samples and $n_2(k)$ is the number of class 2 samples. Then, the probability of the class of $l^{th}$ sample (which will be class 1 if $n_1(k) > n_2(k)$ or class 2 if $n_2(k) > n_1(k)$) is given by 3.5.

\[
p_k(c_l) = \frac{n_1(k)}{n_1(k) + n_2(k)} \quad \text{if } n_1(k) > n_2(k)
\]

\[
p_k(c_l) = \frac{n_2(k)}{n_1(k) + n_2(k)} \quad \text{if } n_1(k) < n_2(k)
\]  

(3.5)

In Equation 3.6, $\hat{\sigma}$ refers to an estimation of intra-class variance and the method of its calculation is shown in 3.7. In detail, $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the estimated variances of the samples in class 1 and class 2 respectively.
3.3.2 Scoring Rule

The error rate $e$ and the reliability $r$ together determines the final classification score $J(\cdot)$. A classifier based on certain features is ranked to be better than another if its $J(e, r)$ score is higher. Its definition plays a crucial role in the feature selection. An effective scoring rule combining error rate and reliability can highly improve the performance of the classifier. In this work, two different scoring rules are considered.

The first scoring scheme called **lexicographic sorting** is introduced in [7] and is a two step ranking process. In this approach, features are firstly sorted by increasing error rate, and then, reliability is considered to break ties among features with equal error rate. This criterion produces a lexicographic sorting of the features in which the reliability parameter has a secondary role.

Since lexicographic approach has been known to perform less as the feature set cardinality grows, a second scoring scheme to make better use of the reliability information is also considered. In this approach introduced in [20], error rate and reliability value are combined to produce a score. With this method, it is possible to select among two features, a feature with a higher reliability and a slightly higher error rate as the best feature. This flexibility can be useful for small sample datasets like microarrays. It compares the features in terms of reliability value, properly penalized depending on the estimated error rate. The aim of the penalization is to introduce a fixed penalization factor to the reliability value for a constant error difference. Such a behaviour is obtained through **exponential penalization** to the reliability value as shown in Equation 3.8, where $e$ is the error rate value, $r$ is the
reliability value and $\alpha$ is the penalization parameter.

$$J(e,r) = r \cdot \exp \left( -\text{sign}(r) \cdot \frac{100}{\alpha} \cdot e \right)$$

In the Equation 3.8, $-\text{sign}(r)$ factor in the exponent has been included to highly penalize features with negative reliability, while $\alpha$ parameter defines the steepness of the penalization: between two features with equal reliability value, an $\alpha\%$ difference in error rate induces a $e^{-1}$ penalization factor in the final score. Hence, when $\alpha$ is small, the dominant parameter is the error rate (at the limit, when $\alpha \rightarrow \infty$, the reliability parameter has no influence at all. When $\alpha$ is large, the dominant parameter becomes the reliability (when $\alpha \rightarrow \infty$, the error rate has no influence in the scoring rule).

Fig. 3.5: Score surface in the error-reliability space for (a) lexicographic sorting and (b) exponential penalization with $\alpha = 10$

To visualize how the score value changes with scoring rule, Figure 3.5 has been introduced. Figure 3.5(b) shows how in the exponential combination case, the score has an exponential decrease along the error dimension and linear trend along the reliability dimension. Figure 3.5(a) shows the score variation for 10 error rate values (assuming that there are only 10 samples) in order to visualize its behaviour properly. It is a stairway surface showing how the main dimension is the error value. Only if two features share the same error value the reliability is taken into account (it shows a linear trend in the reliability direction). Otherwise the score of a feature with smaller error rate is higher, regardless of the reliability value. From
Figure 3.5, it can be observed how both the scoring rules combining reliability and error rate radically change the score surface. From a stairway-like surface (with discontinuities between allowed error rates), the score surface is transformed into a continuous surface in which the reliability gains more decision power. This change is more noticeable when the test set cardinality grows. In such a scenario, the lexicographic scoring would be like a stairway with many small steps, making the reliability parameter almost useless.

As can be observed by the definition of exponential combination, it depends on a parameter (i.e. $\alpha$) that must be previously chosen. Most of the cases in this work, an alpha value of 10 is used as it has been yielding good results whenever exponential penalization based scoring criteria has been used. It is also the best $\alpha$ value when comparing the performance studies in [20].
In this chapter, the datasets used to evaluate the predictive potential of the classifiers built with proposed framework are presented. The datasets used in this work are the same as the ones used in [7] to make the performance comparisons easier. The datasets used and their main characteristics are presented in Section 4.1. The experimental protocol used to test the datasets, the data preprocessing, parameters selected for the experiments, and the methods for the performance analysis are described in Section 4.2.

4.1 Datasets

The predictive properties of the proposed framework has been evaluated on three publicly available datasets. Their main characteristics are summarized in Table 4.1.

- **Colon Dataset**: It consists of 2000 gene expressions each for 62 patients, out of which 40 with colon cancer [25]. This is a commonly used dataset in the literature to evaluate classification algorithms and can be downloaded at [http://genomics-pubs.princeton.edu/oncology/](http://genomics-pubs.princeton.edu/oncology/)

- **Leukemia Dataset**: This is another commonly used dataset in the microarray research [26]. It has 7129 gene expressions of 72 cancer patients, 47 of them having Acute Lymphoblastic Leukemia (ALL) and 25 of them having Acute Myeloblastic Leukemia (AML). In this dataset, the training set and the validation set are already defined. Training Set consists of 38 samples (27 ALL and 11 AML) and testing set consists of 34 samples (20
ALL and 14 AML). It can be downloaded from: \url{http://www.broadinstitute.org/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43}

- **Lymphoma Dataset**: It is a collection of expression measurements from 96 normal and malignant lymphocyte samples [27]. It has 4026 gene expressions of 42 diffused large B-cell Lymphoma (DLBCL) patients and 54 other patients. The dataset is available at \url{http://llmpp.nih.gov/lymphoma/data.shtml}.

<table>
<thead>
<tr>
<th></th>
<th>Colon</th>
<th>Leukemia</th>
<th>Lymphoma</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Genes</td>
<td>2000</td>
<td>7129</td>
<td>4026</td>
</tr>
<tr>
<td>No. of Samples</td>
<td>62</td>
<td>72</td>
<td>96</td>
</tr>
<tr>
<td>Class 1 Members</td>
<td>40(Cancer)</td>
<td>47(ALL)</td>
<td>42(DLBCL)</td>
</tr>
<tr>
<td>Class 2 Members</td>
<td>22(No Cancer)</td>
<td>25(AML)</td>
<td>54(No DLBCL)</td>
</tr>
</tbody>
</table>

Table 4.1: Database Characteristics

### 4.2 Experimental Protocol

The adopted experimental protocol to test the proposed framework is described here. All the datasets pass through a preprocessing phase before the hierarchical clustering process. The preprocessing consists of applying a base two logarithmic transformation in order to reduce the dynamic range, and applying of a minimum threshold of $log_2 10$ in order to remove the unreliable smaller probe values. Finally, each probe set (or each gene expression vector) is forced to have zero mean. Since Leukemia dataset had many negative values in the original data making the logarithmic transformation difficult, it underwent a filtering process before the preprocessing phase. The filtering process is same as in [28][7], that led to a feature number reduction from 7129 to 3859 genes. It relies on a threshold operation, where the minimum value is set to 20 and the maximum value is set to 16000 for all the original data, followed by the exclusion of all genes with a dynamic range smaller than 500 or fold change (ratio of maximum value to minimum value) less than 5.
Afterwards, each of the training dataset is passed to the hierarchical clustering algorithm to generate new feature set based on Treelet clustering or Euclidean clustering as explained in Section 2.3. Since the maximum number of dominant genes (M) for any cluster has to be selected, a range of M values (M=1,2,4,6,8,10) are considered for the analysis. The performance of each dataset for different M values will be analysed.

After the feature set enhancement is done with different cases, each of them will be tested with the feature selection algorithm. The feature selection algorithm has been implemented with two different scoring rules, lexicographic scoring and exponential penalization (with the best case α value proposed in [20]) as discussed before. The performance of different classifiers (LDA and KNN) will also be analysed during the evaluation. For the feature selection algorithm with KNN classifier, the performance results with different K values \((K = 3, 5, 7)\) are compared and the best representative K value (or values) is chosen for each dataset. Only those best case K value results will be discussed. For each experimental set-up, depending on the clustering algorithm or the adopted scoring rule, the best results for each classifier (KNN and LDA) are taken as the representative of the method potential.

Inside each feature selection phase, a five times ten-fold cross validation has been adopted in this work. Furthermore, a different dataset partition has been applied at each iteration of the cross validation, in order to reduce the cross validation variance and bias, except for the Leukemia dataset. In the case of Leukemia dataset, since it has already predefined training and validation set, only those partitions will be used. But, since the number of samples in the training set is comparatively less 4.1, it might degrade the performance of the classifiers.

The performance result for any experimental set-up will be shown with the best case error rate and reliability values. Error rate is shown in the fractional scale, which can be converted to percentage by multiplying by 100. For each of the datasets, the representative scheme for the proposed approach will be compared with the performance of the hierarchical clustering algorithm in the previous work [7].
This chapter discusses the main performance results of the proposed framework following the protocol proposed in Chapter 4. The main goal of the experiments is to analyse whether the enrichment of the microarray data with the introduction of histogram based features improves the performance of the microarray classification task. Since the approach used in this work has a lot of flexible parameters, it can be used for a broad range of analysis. The proposed approach will be considered useful if it gives better classification performance, or it allows the same performance with fewer features, than using original microarray data. Fewer number of features is considered as a performance criteria since it makes the computation easier. Ideally the algorithm is expected to select all the genes involved in a particular cancer as the best features, though it might not happen due to small sample scenario and variation in microarray experiments. If the algorithm could find all the genes involved with the particular cancer, this information could also be used for treatment of genes involved in cancer, as in gene therapy. But, due to heavy computation needed as the number of features needed, it is recommended to group the genes in some clusters so that the cluster can be considered as a feature, similar to the idea proposed in this work. In this way, better the biological similarity of genes in a given cluster in the hierarchical clustering, easier will be for the algorithm to select those features to get best performance. Various performances of this work will be compared with the previous work in [7]. There has been mainly two new concepts used in this work for the microarray classification. One, using the histogram representation and hence using earth mover distance (EMD) as a distance measure. Second, the merging approach used to create representative features.
The analysis method followed here is as follows. Firstly, performance of the proposed hierarchical clustering has been compared with the performance of the approach used in [7]. Once a reasonable performance has been assured, the analysis to select the best candidate for number of dominant genes (M) has been done. Ideally, the candidate for M should be only dataset dependent, or dependent on the type of cancer. For any dataset, the best M value that can perfectly represent most of the variation in a given cluster in the hierarchical clustering algorithm will be chosen. If the best M value has been chosen, the IFFS algorithm should be able to get the best performance using very low number of features. This performance should be ideally independent of the classifier used, and hence if a classifier is not performing well with one dataset, it points to the weakness of the classifier or the classifier design for that particular dataset.

Colon Dataset [25] has been used here as the primary analysis dataset. The main results and analysis from the colon dataset is later compared with the Lymphoma Dataset [27]. Leukemia Dataset [26] has been used just to compare some general performance results. Mainly the critical points of analysis made from colon dataset will be tested with other datasets to check any possible bias of the results due to dataset. Also, lexicographic sorting has been used in the IFFS by default, unless specified as exponential penalization.

Throughout this results and analysis chapter, 'MaC' refers to clustering approach used in [7] and 'SanC' refers to the clustering approach developed in this work, which can be using euclidean (euc) or correlation (corr) distance as a ground distance for EMD, and different number of dominant genes (M). As it has already been mentioned before, by default lexicographic sorting ('ls') is used. When exponential penalization ('ep') is considered, it will be either explicitly mentioned or the results will have 'ep' label with it.
5.1 Performance analysis of the proposed Hierarchical Clustering

For the first analysis, the number of dominant genes in a cluster has been set to 1 (i.e., M=1). This makes the histogram based feature with just one gene/metagene expression and its occurrence. Hence, earth mover distance will converge to the corresponding ground distance (euclidean or correlation distance). This particular case will make the performance comparisons with [7] easier since, only one difference exist in the two approaches in this case. The only difference is that in [7], a metagene is created using a principal component analysis (PCA) of the two most similar gene/metagene expressions in a given cluster in the hierarchical clustering process. In the current work, a weighted averaging based metagene generation is used as explained in Section 2.3. Hence the analysis with M=1 is mainly a comparison of the metagene generation process. In order to compare the performances, the IFFS algorithm is applied to both enhanced feature sets, and the performance results are shown in the Table 5.1 for euclidean distance based clustering and Table 5.2 for correlation distance based clustering. The two tables are also compared with each other to compare the performance of this two distance measures. Both tables shows the error rate and reliability variation as a function of number of features (n) selected for each of the approaches with LDA or KNN classifiers used in IFFS. The first value in any cell is the fractional error rate (multiplying by 100 yields the percentage) and the second value gives the reliability values as defined in the Chapter 3. The last column of any of the performance table shows the best performance of that particular approach (zero or least error rate with lowest number of features possible), and the number of features is indicated in braces {.}.

As it can be seen in Table 5.1, the new approach introduced in this work (indicated as SanC) is able to perform really well compared to the approach introduced in [7] (indicated as MaC). The new approach is able to give zero error rate at much less number of features. This result shows that the weighted averaging method of metagene generation might be better
Table 5.1: Performance Comparison of all the hierarchical clustering approaches using Euclidean Distance

<table>
<thead>
<tr>
<th></th>
<th>n=1</th>
<th>n=2</th>
<th>n=3</th>
<th>n=4</th>
<th>n≤ 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SanC+ LDA</td>
<td>0.114286,</td>
<td>0.071429,</td>
<td>0.028571,</td>
<td>0.028571,</td>
<td>0.0000,</td>
</tr>
<tr>
<td></td>
<td>2.2399</td>
<td>4.926720</td>
<td>5.332640</td>
<td>6.451355</td>
<td>7.426177</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>{7}</td>
</tr>
<tr>
<td>MaC+ LDA</td>
<td>0.14286,</td>
<td>0.071429,</td>
<td>0.058571,</td>
<td>0.042851,</td>
<td>0.000000,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>{8}</td>
</tr>
<tr>
<td>MaC+ KNN(K=3)</td>
<td>0.142857,</td>
<td>0.100000,</td>
<td>0.071429,</td>
<td>0.071429,</td>
<td>0.014286,</td>
</tr>
<tr>
<td></td>
<td>1.512016</td>
<td>2.092873</td>
<td>3.210530</td>
<td>3.472993</td>
<td>6.130867</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>{9}</td>
</tr>
<tr>
<td>SanC+ KNN(K=3)</td>
<td>0.142857,</td>
<td>0.057143,</td>
<td>0.028571,</td>
<td>0.028571,</td>
<td>0.028571,</td>
</tr>
<tr>
<td></td>
<td>2.832228</td>
<td>4.221785</td>
<td>3.462391</td>
<td>3.622241</td>
<td>3.842640</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>{7}</td>
</tr>
</tbody>
</table>

than PCA based method. One possible reason for this is the following. Since, the purpose of hierarchical clustering is to group the most similar genes, then using the best representation of metagene could be the mean or average of the two gene expressions. In principal component analysis (PCA), an orthogonal transformation is used to convert possibly correlated gene expressions into a set of values of linearly uncorrelated variables called principal components. The first principal component is used as a metagene to preserve maximum variance of the initial gene expressions.

Comparison of Performances with Euclidean and Correlation Distances

The correlation distance tries to capture the profile shape similarity among gene expressions while euclidean distance finds the point-wise distance. To understand the significance of correlation distance on the microarray gene expressions, a separate analysis has been carried out. The microarray samples can be in any order depending on the manufacturer of the microarray chip. Hence it was necessary to make sure that the correlation distance between
Table 5.2: Performance Comparison of all the hierarchical clustering approaches using Correlation Distance

two different genes does not depend on the arrangement of samples, to make sure that the hierarchical clustering algorithm yields same result irrespective of the arrangement of samples. For the case of euclidean distance, since it depends only on the point wise similarity, it is obvious that it wont depend on the sample arrangement. But for the correlation distance, the answer doesn’t seem obvious. In this analysis, different samples were interchanged and the correlation distance was found. It has been confirmed that the change of arrangement of samples does not affect the correlation distance along the other dimension.

Figure 5.1 shows the plot of error rate as a function of number of features selected, for the correlation and euclidean distance based hierarchical clustering and using LDA as a classifier for IFFS. Even though both distance based methods are able to give a good performance, the euclidean distance has a slight better performance in terms of convergence of error rate for the first feature selected.

Another important observation from these results is that, LDA classifier has better consistency in providing zero error rate classification. KNN classifier performance is highly varying. For a particular case(SanC), KNN classifier is able to provide really good performance with very less number of features. This particular case will be analysed in detail to verify
whether it is a dataset specific result. For correlation based KNN, Table 5.2 shows a reliability of 0. It is due to the fact that, when only one feature is selected, there is no profile shape for the samples, and hence all the pairwise distances are equal, causing a random selection of the feature and the algorithm assigns a reliability of 0. The IFFS algorithm will make sure that this feature, if not the best feature, will be replaced by the best feature in the replacing phase. The variation of performance among KNN classifier with different K value as shown in Figure 5.2 is also interesting to analyse. It can be seen that, as K increases, the error rate convergence is less. It might be due to the fact that the number of samples in the chosen dataset is low. Hence future studies with higher number of samples is required to study the influence of number of samples with KNN.

**Gene Level Analysis**

The features selected (in terms of their gene constituency) by the algorithm to produce the result in the Table 5.1 is analysed here. Due to biological reasons, the best features selected should ideally have all the genes that are involved in the disease. Hence ideally, we expect all the best approaches proposed having exactly same gene constituency. But this may not be
the case due to many variations among the microarray experiments and also due to lack of perfect characterization of noise in the microarray data. One important observation made is that, the two best features obtaining the results of SanC+LDA and MaC+LDA has a common gene involved. And one more observation is that total number of constituent genes in the metagenes selected are only very few (four or five). This seems very logical as most of the genetic diseases are resulting due to irregularities with few genes [29]. As seen in the Table 5.1, these two cases which selected the features that has this particular gene expression gave the best performance. Further analysis on the gene constituency will be done with the best performing result in the following sections.

**Performance analysis with the number of Dominant Genes**

In this part of the analysis, performance of the algorithm is compared with different M values. Increasing M larger than 1 will cause the features to take the form of multidimensional histograms (genes/metagenes and their corresponding significance). The main proposal of this work will be proven if increasing M has any performance gain. The merging algorithm will be more significant when M value is greater than 1. Increasing M means, increasing the
number of representative features for any given cluster. The main advantage of increasing $M$ is that each of the clusters can have up to $M$ gene/metagene expressions, which prevents two gene expressions which are too distant to merge even closer to the root node as it might happen in the approach used in [7]. Also, it gives similar gene expressions for the classifier by selecting just one feature, hence increasing the classification performance. This analysis is done using only LDA classifier and euclidean distance as a ground distance for the EMD, as they both gave consistent performance for most of the experiments. The performances for each $M$ value as a function of number of features selected are shown in the Table 5.3.

<table>
<thead>
<tr>
<th>M</th>
<th>n=1</th>
<th>n=2</th>
<th>n=3</th>
<th>n=4</th>
<th>n≤10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1285, 2.2399</td>
<td>0.071429, 4.926720</td>
<td>0.057143, 5.332640</td>
<td>0.042857, 6.451355</td>
<td>0.0000, 7.426177 {7}</td>
</tr>
<tr>
<td>M=2</td>
<td>0.085714, 3.205664</td>
<td>0.057143, 4.136158</td>
<td>0.042857, 4.701592</td>
<td>0.042857, 5.602536</td>
<td>0.000000, 8.794973 {6}</td>
</tr>
<tr>
<td>M=4</td>
<td>0.100000, 4.427365</td>
<td>0.042857, 6.062290</td>
<td>0.028571, 6.374049</td>
<td>0.014286, 8.955011</td>
<td>0.000000, 9.311024 {5}</td>
</tr>
<tr>
<td>M=6</td>
<td>0.071429, 4.294168</td>
<td>0.057143, 5.440911</td>
<td>0.042857, 5.843734</td>
<td>0.014286, 9.549024</td>
<td>0.000000, 13.928483 {5}</td>
</tr>
<tr>
<td>M=8</td>
<td>0.057143, 4.769390</td>
<td>0.042857, 4.479889</td>
<td>0.014286, 5.915590</td>
<td>0.000000, 11.769889</td>
<td>0.000000, 11.769889 {4}</td>
</tr>
<tr>
<td>M=10</td>
<td>0.057143, 4.848773</td>
<td>0.042857, 8.226305</td>
<td>0.014286, 12.377313</td>
<td>0.000000, 21.132098</td>
<td>0.000000, 21.132098 {4}</td>
</tr>
<tr>
<td>M=20</td>
<td>0.085714, 4.723699</td>
<td>0.071429, 11.605598</td>
<td>0.014286, 28.946030</td>
<td>0.000000, 43.227637</td>
<td>0.000000, 43.227637 {4}</td>
</tr>
</tbody>
</table>

Table 5.3: Performance Comparison of all the hierarchical clustering with different number of dominant genes

As it can be seen from the Table 5.3, only very less number of features are needed for the algorithm to give a good performance with higher $M$. Hence performance is increasing with $M$. The performance is increasing until all the important features of any cluster has
been represented by the dominant genes. After a particular M which is dataset specific, the performance saturates. As it will be seen in the case of lymphoma dataset, too high value of M can cause performance degradation. In all the cases analysed, a smaller increase of M from 1 is always resulting in performance gain. Also, the reliability value is increased for a fixed number of features, as M is increased. Reliability is a measure of inter-class separability and hence higher reliability is better to have better and more stable performance. The variation of error rate as a function of M for a fixed number(4) of features selected is shown in Figure 5.3.

![Error Rate variation as a function of number of dominant genes for a fixed number of features (Colon Dataset)](image)

Fig. 5.3: Error Rate variation as a function of number of dominant genes for a fixed number of features (Colon Dataset)

Hierarchical clustering can be considered as highly effective if the majority of the features selected by the IFFS algorithm is from the new set of features. It has been noticed that, as M increases, the majority of features selected are from the new feature set. In a particular case of M=4, all the features selected by the IFFS are from the new feature set. As it can be seen from the Table 5.3, the best performance of this new approach is giving 0 error rate with 4 features. M=20 case was analysed to see if the performance is degrading after a particular M value. As it can be seen for M=20, the convergence rate is less, but final performance is still the same with higher reliability.
5.2 Performance analysis with LDA and KNN

This section tries to understand the difference in the performance between LDA and KNN. LDA classifier has been yielding consistent and good performance in all the analysis. But KNN performance has been not so stable, hence this section mainly focus on the KNN analysis. For analysing the performance difference, the two best features selected by the LDA and KNN in the case of $M=1$ analysis has been used to provide a classification scatter plot as shown in Figure 5.4 and 5.5. One interesting observation from the figure is that, the best features selected by KNN classifier also has a almost linear boundary between the two classes, which explains the performance advantage of LDA classifier. Even then, KNN is giving less number of error in the figure, due to the flexibility allowed by KNN to choose the best $K$. This result is the best result among the cross validation iterations with LDA and KNN. Even though we can use it for a general analysis, more analysis is needed to compare the performances. $M=8$ can be seen as a perfect $M$ value to get a very good performance for the colon dataset. Ideally, this result should not depend on the classifier used.

In order to understand the performance of KNN with different distances, four different cases of analysis has been carried with combinations of euclidean and correlation distance being used for hierarchical clustering and also as the distance measure in the KNN. The results are shown in the Table 5.4 as a function of number of features. All this analysis has been done with hierarchical clustering using $M=8$ dominant genes since it has been identified as the best performing value for colon dataset.

One interesting observation made by analysing the selected features by the best performing KNN classifier is that, it selected the feature that has contributions of only 8 basic genes to give the best result marked as (**) in the Table 5.4. This result seems very similar to the best result obtained for $M=8$ in Table 5.3. This is pointing to the fact that only more or less 8 genes are involved in the colon cancer. Hence $M=8$ as the best choice for colon dataset is perfectly justified. This result also shows that KNN can perform extremely well for some
datasets, but its performance is not stable as we have seen.

![Fig. 5.4: The scatter plot of the KNN classifier with best two features selected](image)

As it can be seen in Table 5.4, the best performance of KNN comes with correlation distance used as a distance measure for KNN. This result is same irrespective of the ground distance used in the hierarchical clustering. But, this has to be tested with other datasets to confirm if it is a general result. As we will see later in Table 5.5, the performance was not the same for lymphoma dataset. Hence KNN in general has a less consistent performance when comparing the performances with different datasets. One possible reason [30] for such a performance degradation would be that the class distribution of the datasets used in this work are very skewed, meaning number of samples are not the same for different classes. As it can be seen in Section 4.1, all the datasets used in this work has non-uniform class distribution, with unequal number of training samples in both the classes. It is a main drawback to the basic “majority voting” classification, the classes with the more frequent training samples tend to dominate the prediction of the new sample, as they tend to come up in the k nearest neighbours when the neighbours are computed due to their large number. One
Fig. 5.5: The scatter plot of the LDA classifier with best two features selected

<table>
<thead>
<tr>
<th>M=8, Colon Dataset</th>
<th>n=1</th>
<th>n=2</th>
<th>n=3</th>
<th>n=4</th>
<th>n≤ 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SanC(euc)+3NN(corr)**</td>
<td>0.114286, 2.038467</td>
<td>0.042857, 2.787556</td>
<td>0.028571, 4.025453</td>
<td>0.000000, 2.762684</td>
<td>0.000000, 2.762684 {4}</td>
</tr>
<tr>
<td>SanC(corr)+3NN(corr)</td>
<td>0.128571, 3.437814</td>
<td>3.437814, 3.157973</td>
<td>0.042857, 4.279092</td>
<td>0.0142863, 5.075342</td>
<td>0.0142863, 5.075342 {4}</td>
</tr>
<tr>
<td>SanC(corr)+3NN(euc)</td>
<td>0.100000, 2.849337</td>
<td>0.071429, 2.858991</td>
<td>0.042857, 3.050362</td>
<td>0.057143, 4.393319</td>
<td>0.028571, 3.400479 {7}</td>
</tr>
<tr>
<td>SanC(corr)+3NN(euc)-mr</td>
<td>0.100000, 3.198716</td>
<td>0.057143, 7.117960</td>
<td>0.042857, 4.991416</td>
<td>0.014286, 5.519891</td>
<td>0.014286, 8.418647 {5}</td>
</tr>
<tr>
<td>SanC(euc)+3NN(euc)-mr</td>
<td>0.100000, 3.430395</td>
<td>0.057143, 4.135433</td>
<td>0.042857, 7.796659</td>
<td>0.042857, 7.412749</td>
<td>0.028571, 4.469279 {6}</td>
</tr>
</tbody>
</table>

Table 5.4: Performance analysis of KNN classifier
way to overcome this problem is to weigh the classification taking into account the distance from the test point to each of its k nearest neighbours. This can be analysed in the future work of this project.

The results in the Table 5.4 also confirms to our previous analysis that hierarchical clustering based on euclidean distance is performing better. The best performing combination (Euclidean based hierarchical clustering and correlation based KNN(K=3)) in the previous analysis for the colon dataset has been applied to lymphoma dataset to see whether the performance is similar for KNN classifier. Two different M values are analysed to make sure that the results are not biased. The performance of the KNN classifier for this particular case with different M values is shown in Table 5.5. Even with KNN, increasing M has a good effect on the classification performance. Performance and reliability of KNN classifiers increases with higher K values [23]. But since it is not possible for a larger K value with datasets having sample size around 100, the KNN classifiers are not able to perform to give zero error rate in most of the cases. K=5 was also considered for the simulation, and the result is not shown since the performance with K=5 was also same as the case with K=3.

<table>
<thead>
<tr>
<th>M=1</th>
<th>n=1</th>
<th>n=2</th>
<th>n=3</th>
<th>n=4</th>
<th>n≤10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.440000,</td>
<td>0.00000</td>
<td>0.020000,</td>
<td>0.020000,</td>
<td>0.010000,</td>
<td>0.010000,</td>
</tr>
<tr>
<td>6.213286</td>
<td></td>
<td>6.170896</td>
<td>7.793573</td>
<td></td>
<td>9.584797</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>{6}</td>
</tr>
<tr>
<td>M=4</td>
<td>0.140000,</td>
<td>0.020000,</td>
<td>0.010000,</td>
<td>0.010000,</td>
<td>0.010000,</td>
</tr>
<tr>
<td>6.380983</td>
<td>7.890699</td>
<td>8.167689</td>
<td>11.882160</td>
<td>14.512450</td>
<td>{5}</td>
</tr>
<tr>
<td>M=10</td>
<td>0.110000,</td>
<td>0.040000,</td>
<td>0.020000,</td>
<td>0.020000,</td>
<td>0.020000,</td>
</tr>
<tr>
<td>6.750165</td>
<td>6.620572</td>
<td>10.358335</td>
<td>14.972809</td>
<td>14.836302</td>
<td>{5}</td>
</tr>
</tbody>
</table>

Table 5.5: Performance analysis of KNN classifier for Lymphoma Dataset

Table 5.5 proves that the performance result of particular case of the KNN classifier was a dataset specific result. But still, there is a general trend for KNN classifier to give very less error rate with low number of features. This can be seen in all the dataset. But the reason of skewed class distribution and sample scarcity might be causing it to not reach the
final performance task of zero error rate. KNN classifier needs to be analysed with a larger
dataset and using many combination of distance measures to utilize its maximum capacity.
But in general, KNN classifier performance is highly varying and cannot be used as a standard
for the datasets used in this work. As discussed before, using the number of occurrences in
the histogram based feature, to scale the distance also need to be analysed in the future work.

Trying to understand performance degradation for the case of M=10 in Table 5.5,
the gene level analysis has been carried out. It has been found that the feature selected by
the algorithm has gene contributions from around 1000 basic genes. As it has been already
discussed and verified, most performing features are the ones that have very less genes involved.
But the algorithm selected this feature since it gave good performance in terms of error rate
with just two features. But since it did not match the underlying gene expression’s differential
power, it is unable to give zero error rate even with higher number of features.

5.3 Generalization of the Results and Analysis

In this section, the main analysis steps of the colon datasets are repeated with lymphoma
dataset to make sure that there has been no bias in the analysis due to some specificity of
the colon dataset. Finally, a section also discusses the performance of the algorithm with the
Leukaemia dataset.

Lymphoma Dataset

Table 5.6 shows the performance of the LDA classifier for different M values as a function of
number of features selected. This analysis has also been performed in order to compare the
performance of the two scoring rules used in this work, lexicographic scoring and exponential
penalization method.

As it can also be seen in Table 5.6, increasing M from 1 to 3 made the algorithm to
Table 5.6: Performance variation of Lymphoma Dataset with M values and scoring rules

give the zero error rate with just 3 features. The general trend noticed for all the performance analysis is that, as M increases, the reliability is increasing always. But number of features needed for zero error rate is reaching a minimum value and increases again as shown in Figure 5.6. It is expected as most of the genes/metagenes in a given cluster are very similar and atleast a few set of similar genes are needed for getting very good classification. It can also be observed that, exponential penalization (with $\alpha = 10$) improves the performance of the classification for M=10 compared to lexicographic sorting based approach. At the same time, lexicographic approach gives better result for the case of M=4. This is expected from the definition of the two scoring rules. Lexicographic method works very well when there are many features that has error rate variation. The main use of exponential penalization is when there are many features with same error rate but with very different reliability results.

Now the the performance of hierarchical clustering used in [7] and the new approach with M=1 for the lymphoma dataset (both euc and corr is used to see if the perfor-
mance difference if any) is done. This analysis is also done to check whether the analysis on colon dataset can be generalized. Table 5.7 shows the performances of different cases.

<table>
<thead>
<tr>
<th></th>
<th>n=1</th>
<th>n=2</th>
<th>n=3</th>
<th>n=4</th>
<th>n≤ 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SanC(corr)+</td>
<td>0.050000, 7.085590</td>
<td>0.020000, 8.263917</td>
<td>0.010000, 11.414924</td>
<td>0.000000, 10.777121</td>
<td>0.000000, 10.777121</td>
</tr>
<tr>
<td>LDA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>{4}</td>
</tr>
<tr>
<td>SanC(euc)+</td>
<td>0.050000, 7.403848</td>
<td>0.030000, 8.593285</td>
<td>0.020000, 9.251564</td>
<td>0.000000, 10.462487</td>
<td>0.000000, 10.462487</td>
</tr>
<tr>
<td>LDA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>{4}</td>
</tr>
<tr>
<td>MaC(corr)+</td>
<td>0.040000, 7.825007</td>
<td>0.020000, 9.686233</td>
<td>0.010000, 10.680459</td>
<td>0.000000, 11.994877</td>
<td>0.000000, 11.994877</td>
</tr>
<tr>
<td>LDA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>{4}</td>
</tr>
<tr>
<td>MaC(euc)+</td>
<td>0.030000, 8.805329</td>
<td>0.020000, 10.183921</td>
<td>0.010000, 11.557658</td>
<td>0.000000, 12.630389</td>
<td>0.000000, 12.630389</td>
</tr>
<tr>
<td>LDA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>{4}</td>
</tr>
</tbody>
</table>

Table 5.7: Comparison of different clustering approaches with lymphoma dataset

For lymphoma dataset, MaC and SanC has very good performance. But if we compare the performance with the colon dataset, the new proposed approach has a performance that is more stable irrespective of the dataset. In order to analyse more on this, more experiments with new datasets are needed in future. The general performance improvement for this dataset is also due to the fact that the lymphoma dataset has more number of training
samples.

**Leukaemia Dataset**

The main challenge or problem with this dataset [26] is that, the training set samples are too less, and hence the performance may not be reliable. When the training dataset doesn’t have much samples, not all the variability can be easily captured by the algorithm. The results of this database shows the importance of the training data for any classification. If the number of training data is too less, learning the characteristics of the cancer from the dataset is very difficult and hence the performance of the classifier won’t be good.

The results of the LDA classifier applied on the enhanced Leukaemia dataset is shown in Table 5.8. Even though the performance is comparatively very low, the proposed approach is still able give considerable performance. Some of the best results are shown in the table. Exponential penalization based scoring rule has been used in the IFFS which proved to give better result in this dataset. The ground distance used for the earth mover distance(EMD) in this case is euclidean distance.

<table>
<thead>
<tr>
<th>M=1</th>
<th>n=1</th>
<th>n=2</th>
<th>n=3</th>
<th>n≤ 10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.264706,1.514318</td>
<td>0.135294,3.902352</td>
<td>0.061978,3.328730</td>
<td>0.029412,4.6180 {7}</td>
</tr>
<tr>
<td>M=2</td>
<td>0.264706,2.215515</td>
<td>0.144265,4.207642</td>
<td>0.085294,4.207642</td>
<td>0.01412,3.690789 {9}</td>
</tr>
<tr>
<td>M=4</td>
<td>0.122500,2.497607</td>
<td>0.075000,3.660564</td>
<td>0.075000,5.343028</td>
<td>0.000000,4.581450 {7}</td>
</tr>
</tbody>
</table>

Table 5.8: Performance evaluation for Leukemia dataset

As it can be seen in Table 5.8, most of the classifiers did not yield any zero error rate. This result has been shown mainly for the completion sake to compare the performance with other approaches. In general, any dataset that has very low number of training set
will yield a low performance classification. The same dataset when applied through cross validation, is able to produce much better result, but it cannot be compared here since the dataset itself has a predefined training set and validation set partition.

Even then, the proposed approach of hierarchical clustering is giving better result. Figure 5.7 shows the performance variation for different values of M. As it can be seen there, M=4 has an advantage over M=1. The performances of all the datasets will be compared with the state of the art methods in the next section.

![Fig. 5.7: Error Rate variation as a function of number of features for M=1 and M=4 dominant genes of Leukemia Dataset](image)

**5.4 Comparison of the proposed clustering with State of the art**

As we have already seen in previous sections, the proposed approach has a better or comparable performance depending on the scenario. The Table 5.9 shows the comparison of the best approaches described in the proposed work with the state of the art approaches. The performance is shown as the minimum achievable error rate and the number of features required for that error rate is shown in braces \{ . \} when available.
Comparing the results obtained with the state of the art, it can be seen that proposed method produces better classifiers (zero error rate with very few features) for Colon dataset and Lymphoma dataset. For the Leukaemia dataset, the performance of the classifier could be improved with further analysis and using the flexibility provided by the proposed approach. It needs further optimization studies to get the best combination of parameters to get competitive results.

<table>
<thead>
<tr>
<th></th>
<th>Colon</th>
<th>Lymphoma</th>
<th>Leukemia</th>
</tr>
</thead>
<tbody>
<tr>
<td>SanC(euc) 10CV</td>
<td>0.00% {4}</td>
<td>0.00% {3}</td>
<td>0.00% {7}</td>
</tr>
<tr>
<td>SanC(corr) 10CV</td>
<td>0.00% {4}</td>
<td>0.00% {4}</td>
<td>.</td>
</tr>
<tr>
<td>MaC(euc) 10CV</td>
<td>0.00% {5}</td>
<td>0.00% {4}</td>
<td>2.80% {15}</td>
</tr>
<tr>
<td>MaC(corr) 10CV</td>
<td>0.00% {6}</td>
<td>0.00% {5}</td>
<td>0.00% {5}</td>
</tr>
<tr>
<td>NSGAA II [28]</td>
<td>0.00% {12}</td>
<td>0.00% {12}</td>
<td>0.00% {4}</td>
</tr>
<tr>
<td>TSP [31]</td>
<td>5.40% {2}</td>
<td>.</td>
<td>10.60% {2}</td>
</tr>
<tr>
<td>ICA+LDA [31]</td>
<td>14.045%</td>
<td>.</td>
<td>5.35% {5}</td>
</tr>
</tbody>
</table>

Table 5.9: Performance evaluation for Leukemia dataset
Chapter 6

Conclusions and Future Work

In this work, a new microarray classification framework has been studied. It has been developed as an improvement to the microarray classification scheme proposed in [7]. The main contributions of this work are the concept of histogram based hierarchical clustering, the use of earth mover distance (EMD) as a distance measure for microarray classification, the representative feature (metagene) generation process and the comparison between LDA and KNN classifiers. This work also attempts to analyse the results at the gene level.

Histogram based feature enhancement has proved its usefulness. The performance of the classifiers has always been improved by using more than one representative features for a cluster. The new method of generating metagene has also been found performing very well. It has been found that the best candidate for the maximum number of dominant genes is dataset specific.

Feature selection algorithm has been performing well with lexicographic sorting, except for a few cases where exponential penalization based scoring rule performed better. Linear discriminant analysis (LDA) classifier has been giving consistent performance results across all datasets used. K-nearest neighbour (KNN) classifier performance has been found varying with datasets and was unable to produce zero error rate in most cases, though the error rate convergence has been faster for first few features. The challenges faced by the KNN classifier in the current dataset has been discussed and the solutions could be developed in the future work of the same project.

A gene level analysis has revealed that the best features selected by the feature selection algorithm has only very few basic constituent genes involved. The gene level analysis
done on some specific cases of the results in this work needs to be extended for all the results with the new datasets. A more in-depth analysis of the functions involved in the selected genes will help the tuning of the different parameters in the proposed framework in a better way.

The proposed framework has proven to be a good choice for the microarray classification. It has shown consistent performances with many datasets, yielding competitive performances compared to other state of the art techniques. The main advantage of proposed approach is that, it provides a lot of flexibility to the classification algorithm as it has many tunable parameters depending on the dataset. Further studies are needed to completely exploit all the advantages offered by the proposed framework. One main possible line of work is to analyse the results of the proposed framework for the datasets that has all the genes involved in the human genome, made publicly available by the microarray quality control study, phase II (MAQC II) [22].
Bibliography


