Feature dimension reduction for microarray data analysis

Shi, Chao

2005

Shi, C. (2005). Feature dimension reduction for microarray data analysis. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/3241>

<https://doi.org/10.32657/10356/3241>

Nanyang Technological University

Downloaded on 26 Apr 2023 13:56:48 SGT

Feature Dimension Reduction

For

Microarray Data Analysis

Shi Chao

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Master of Engineering
2005

QH
450
3555
2005

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research done by me and has not been submitted for a higher degree to any other University or Institute.

2005-08-23
.....

Date

Shi Chao
.....

Shi Chao

Acknowledgement

I would like to express my deepest gratitude to my supervisor, Dr. Chen-Lihui, for her guidance, patience and many invaluable ideas and comments throughout the whole research project.

I also take this opportunity to thank Nanyang Technological University for the award of the research scholarship and Information Systems Research Laboratory and Software Engineering Laboratory for their excellent facilities without which present work would not have been possible.

Special thanks also go to my friends He Yu, Chu Feng, Xie Wei, Jin Guosheng, Zhao Ying, Shu Haiyan and all the people in Software Engineering Laboratory, for their generous assistance provided throughout the entire project.

Contents

1	Introduction	11
1.1	Motivation	11
1.2	Objectives	14
1.3	Major contribution of the thesis	14
1.4	Organization of the thesis	15
2	Reviews	16
2.1	Biological background	16
2.1.1	Cells and DNA	16
2.1.2	Structure of DNA	17
2.1.3	Microarray technology	19
2.2	Reviews on machine learning of gene expression data	21
2.3	Problem statement	22
2.4	Reviews on feature reduction	23
2.4.1	Feature selection vs. feature extraction	24
2.4.2	Wrapper approach vs. filter approach	25
2.4.3	Individual gene ranking vs. gene subset ranking	26
2.4.4	Well known methods used in gene expression data feature reduction	28

2.4.5	Classification techniques used	31
3	Entropy ranking for feature selection	34
3.1	Entropy description of cluster	35
3.2	Using entropy to rank features	36
3.3	A fast implementation	38
3.4	Experiments	40
3.4.1	Dataset	40
3.4.2	Software and environment	41
3.4.3	Dataset preprocessing and normalization	42
3.4.4	SVM parameter settings	43
3.4.5	LOOV and accuracy statistics	45
3.4.6	Results	47
3.4.7	Discussions	49
4	Locally linear embedding—a feature extraction approach	54
4.1	Locally linear embedding	54
4.2	Invariance to scaling, rotation and translation	59
4.2.1	Scaling and rotation	59
4.2.2	Translation	60
4.3	Solution for reconstruction weights	61
4.4	Solution for lower-dimensional coordinates	62
4.5	Fractional metrics	64
4.6	LLE from pairwise distance	65
4.7	Experiment and discussion	68
4.7.1	Comparison of LLE+SVM with SVM alone	68

- 4.7.2 Comparison of LLE with other feature reduction methods 70
- 4.7.3 Comparison of Fractional metrics with Euclidean distance 70
- 4.7.4 Discussions 71
- 5 Feature grouping framework – a feature selection approach 77**
 - 5.1 Implementation of the framework 80
 - 5.1.1 Pre-processing 80
 - 5.1.2 Individual filtering 81
 - 5.1.3 Clustering by genes 81
 - 5.1.4 Clusters validation 82
 - 5.1.5 Gene cluster shrinking 83
 - 5.2 Prototyping and discussion 84
 - 5.2.1 Prototype 85
- 6 Conclusion and Future work 89**
 - 6.1 Conclusion 89
 - 6.2 Further research 92

List of Figures

- 2.1 The process of gene expression. Courtesy to [1]and website: [2]. 17
- 2.2 Illustration of how feature reduction helps data classification . 24
- 3.1 Illustration of how entropy varies with pairwise distance . . . 36
- 3.2 Illustration of features’ discrimination capability 37
- 3.3 Data preprocessing and normalization steps 42
- 3.4 Performance of entropy ranking on leukemia dataset 49
- 3.5 Performance of entropy ranking on lymphoma dataset 50
- 3.6 Performance of entropy ranking on colon dataset 51
- 4.1 Flow chart of LLE programme 58
- 4.2 illustration of triangle 66
- 4.3 The performance of LLE feature extraction on leukemia dataset 72
- 4.4 The performance of LLE feature extraction on colon dataset. . 73
- 4.5 The performance of LLE feature extraction on lymphoma dataset. 74
- 4.6 2d scattering of leukemia classes generated by LLE. 75
- 4.7 2d scattering of colon classes generated by LLE. 75
- 4.8 2d scattering of lymphoma classes generated by LLE. 76
- 5.1 Illustration of the framework 80

5.2 Data scattering of leukemia 86

5.3 Performance comparison of feature grouping on leukemia . . . 87

5.4 Performance comparison of feature grouping on leukemia2 . . 88

List of Tables

3.1	Sources of datasets	41
3.2	Gene expression datasets	41
3.3	Comparison of time consumed to generate complete ranking list with and without optimization	48
3.4	Comparison of classification accuracy with and without en- tropy ranking feature selection	52
3.5	Comparison of classification accuracy between Entropy rank- ing with other classical methods using SVM(RBF kernel) . .	52
3.6	Comparison of classification accuracy between Entropy rank- ing with other classical methods with 25 features using SVM(RBF kernel)	53
4.1	Comparison of classification accuracy with and without fea- ture selection	69
4.2	Time consumed for each run of LLE feature reduction	69
4.3	Comparison of classification accuracy between LLE and other classical methods using SVM(RBF kernel)	71

Abstract

Cancer classification is one major application in microarray data analysis. Due to the ultra high dimensional nature of microarray data, data dimensionality reduction has drawn special attention. The currently available data dimensionality reduction methods are either supervised, where data need to be labeled, or computationally complex. In this project, we target to find effective and unsupervised feature reduction tools for gene expression data classification purpose. We have tackled the problem from both feature selection and feature extraction approaches. Three feature reduction algorithms, *fast entropy ranking*, *revised locally linear embedding*, and *feature grouping* are proposed, analyzed and tested on several microarray datasets. The effectiveness and advantage over traditional methods are observed, and recommendations for future research are given.

Chapter 1

Introduction

1.1 Motivation

Cancer is one of the major health concerns in the world. It has been evident from the beginning of the molecular study of cancers that it is not one disease but many, and that any particular cancer arises as the result of the gradual accumulation of genetic changes within a single cell. In most cases, the early detection and treatment can substantially improve the survival rates of cancer patients. Previously, cancer diagnosis has been morphological and phenotype based, which maybe complex and inaccurate. Such diagnosis methods are reported to have several limitations in their diagnostic ability [3]. It has been suggested that specification of therapies according to tumor types differentiated by pathogenetic patterns may maximize the efficacy of the treatments [4] [5] [6] [7][8][9]. Also, the existing tumor classes have been found to be heterogeneous and comprise diseases that are molecularly distinct and follow different clinical courses [10]. Cancer genetics, based on analysis of cancer genotypes, provides a valuable alternative in

both theory and practice.

Michael F.O. and Andrew K.G. [11] have reviewed the progress made on cancer diagnosis with the emergency of microarray technology. “New technologies offer the potential to describe specific types of genetic changes and the patterns of altered gene product expression and function that define the actual definition of any cancer in the context of, but not strictly dependent upon, its site of origin” [11]. “We anticipate that these new technologies will lead to the identification of previously undetectable features of the molecular basis of individual tumor characteristics and profile progression and response to therapy” [11]. It is specially necessary to link the information gained from these new technologies with a wealth of clinical information (obtained from pathologists to oncologists) associated with each clinical sample [11]. *DNA microarrays* are one of these new powerful technologies [11]. Microarray technology has become a major tool for the investigation of global gene expression of all aspects of human disease and in biomedical research since its inception 8 years ago [11].

The expression level of genes is known to contain the keys to address fundamental problems relating to the prevention and treatment of diseases, biological evolution mechanisms and drug discovery. The emergence of cDNA technologies makes it possible to monitor states of thousands of genes in a single experiment [12] [13] [14] [15]. Expression data derived from cDNA chips contains the mRNA level of thousands of genes taken from the same tissue across many patients, which leads to a new way of cancer diagnosis. By studying difference among expression levels, we are able to build up the linkage between *gene expression level* with *disease phenotypes* [16].

In this project, we are specifically interested in mining cancer subtype identification problems. Practical cancer identification process involves employing data mining or machine learning tools, such as data classification and clustering algorithms, to mine gene expression data. During the past several years, many classification and clustering algorithms have been proposed and tested on gene expression datasets [10]. The results reported in the literature have confirmed the effectiveness of mining cancer information from gene expression data [10]. However, the ultra high dimensionality of gene expression data makes the mining still a non-trivial task. Effective feature reduction tools are in great need.

Feature reduction tools play a very important role in the field of data mining:

- Feature reduction helps to reduce the computational burden of many classification and clustering algorithms.
- Feature reduction helps data mining algorithms to concentrate on relevant features, and thus boost the classification or clustering accuracy.
- Feature reduction helps to find the latent factors in multivariate problems, and saves the cost of data acquisition.
- Specific to gene expression mining problems, feature reduction helps to identify the 'marker' genes, and gives hints to biological interpretation.

We are particularly interested in the ability of using unsupervised methods to select features in high dimensional datasets.

1.2 Objectives

Feature reduction has been extensively researched in the data mining literature. However, most of the currently available data dimensionality reduction methods are either supervised, where data need to be labeled, or computationally complex. In this project, we target to explore effective and unsupervised feature reduction strategy to facilitate the work of gene expression classification.

1.3 Major contribution of the thesis

The major contributions of the thesis are highlighted as follows:

- An unsupervised *feature selection* algorithm based on entropy ranking [17] is analyzed and optimized. The optimized algorithm is implemented in Matlab and tested on three publicly available datasets.
- A relatively new *feature extraction* algorithm – Locally linear embedding (LLE) [18] has been analyzed in mathematical details. A more flexible similarity measure is incorporated into the original algorithm. The revised algorithm is implemented and tested on three publicly available datasets.
- An unsupervised feature reduction framework is proposed based on the idea of feature grouping. A prototype of the framework is implemented in Matlab and tested using the *leukemia* datasets.

1.4 Organization of the thesis

The rest of the thesis is organized as follows:

- In Chapter 2, we present the biological background of using gene expression data as a tool for tumor classification, and the necessity of feature reduction tools. A review of the classical and contemporary feature reduction approaches is conducted.
- In Chapter 3, we review the concept of *entropy ranking* proposed by Dash&Liu [17] and analyze the details of using entropy to select informative features. The computational complexity of the original algorithm is emphasized, and optimization steps are suggested. The optimized algorithm is implemented and tested. Comparison with the original algorithm has been made.
- In Chapter 4, we present the application of Locally linear embedding(LLE) [18] algorithm for feature extraction for microarray data analysis. The original algorithm is analyzed in great details, and the revision made is justified via experiments.
- In Chapter 5, we present a new feature selection framework based on the idea of feature grouping and shrinking. A prototype of the proposed framework is implemented and tested.
- In Chapter 6, the project is summarized and suggestions for future work are given.

Chapter 2

Reviews

2.1 Biological background

2.1.1 Cells and DNA

From biology, we know all living beings on earth are composed of cells or cell. Although varying in sizes and shapes, cells share some common internal structure and functionalities. The basic construction units of cells include cell membrane, hereditary material, organelles, and cell wall. Prokaryotic cells store hereditary materials in an area known as nucleoid, while eukaryotic cells store them in cell nucleus. The hereditary materials twist themselves to form complex fila of *chromosomes*. The actual chemicals that cells use to replicate or reproduce exist in the form of *DNAs* or *RNAs*. Most organisms use DNAs, while viruses and viroids use RNAs.

Figure 2.1: The process of gene expression. Courtesy to [1] and website: [2].

2.1.2 Structure of DNA

Unwinding the high level structure of chromosomes, the DNA molecule can be found as a double helix formed by two continuous strings of deoxyribonucleotides. A nucleotide consists of 5-Carbon sugars(deoxyribose), nitrogenous bases, and phosphate groups. The nitrogenous bases lie in the center of the helix, with sugar phosphate units along the sides. There are four kind of nitrogenous bases, namely, cytosine(C), thymine(T), adenine(A), and guanine(G). The pair of bases link across the two strings of the helix conforming to the *complementary matching rule*, with adenine(A) coupled to thymine(T), and guanine(G) coupled to cytosine(C).

Information flow during cell replication or reproduction is governed by the law of *central dogma in inheritance*. By central dogma, the sequence of nitrogenous bases along the strings of the DNA helix actually carries the information for protein synthesis. The actual information flow takes place in

two stages, (I) During *transcription*, a single strand of RNA is made from the template of DNA helix through complementary base pairing. The matching rule is the same as in DNA helix, except uracil(U) replaces thymine(T). When the DNA helix is unwound, one strand of its bases finds their complements to construct a new strand of RNA. (II) During *translation*, amino acid units are constructed from triplets of RNA codons [19]. One triplet of the nitrogenous four bases gives $(4 \times 4 \times 4) = 64$ possible combinations of codons, which eventually codes 20 different amino acids (Many amino acids are coded with redundancy for robustness, and also some codons serve as coding signals). The whole process is illustrated in Fig 2.1. While amino acid serves as the basic building block of protein, it is concluded that the sequence of DNA actually determines the protein synthesized. Since protein constructs and regulates everything in the organisms, the DNA inside the nucleoid or nucleus actually monitors everything in the tissues and organs, therefore, observation of DNA activity in the tissue gives hints to disease diagnosis.

The process of transcribing a gene's DNA sequence into RNA is called *gene expression* [10]. A gene's expression level indicates the approximate number of copies of that gene's RNA produced in a cell and it is generally correlated with the amount of the corresponding proteins made. Some diseases, such as cancer, can be reflected in the change of expression levels of certain genes. The research on gene expression data has shown effectiveness in identifying gene functions and cancer diagnosis [10].

2.1.3 Microarray technology

Previous research on gene expression profile has been carried out on “one gene one experiment” basis, i.e., during each experiment, only one gene is identified and analyzed. The emergence of *microarray (cDNA)* technology has triggered revolutionary changes in this domain, and lead cancer research to a new era. Using the microarray chips, expression levels of thousands of genes can be measured in one single experiment, and the gene expression profile provides much more biological and medical information.

Microarray has exploited the complementary base pairing property of DNA. The microarray chip uses a glass slide with thousands of tiny spots, on which the DNA molecules are attached. In the gene expression experiments, each of these spots should identify one gene or one segment of gene. The chips can be used to compare gene expression levels in different samples, for example, the same tissue cells in healthy and tumor state. The mRNAs from the cells are first extracted and *reversely* transcribed to its complementary DNA(cDNA) strands, and then dyed with two different fluorescent labels, for example, green fluorescent dye Cy3 for cells in reference state, and red fluorescent dye Cy5 for cells in suspected state. The extracts are mixed and applied on the glass slide, and each cDNA molecule is supposed to hybridize with its complementary DNA at certain spot. The different concentration of mRNA molecules in the samples will be reflected by the color intensity in the corresponding spots. If the abundance of the mRNA in reference state is in dominance, the spot will be in green color; If the abundance of the mRNA in suspected state is in dominance, the spot will be in red color. If they are almost equal, the spot color will be mixed. Therefore the relative expression

level of the DNA in two samples can be estimated from the color intensity in each spot [20]. The glass slides are then scanned by laser, and output as microarray images. The intensity values are obtained by image analysis software. The log ratio between the two intensities of each color is taken as the *gene expression* data [21] [22][23].

$$gene-expression = \log \frac{Intensity(Cy5)}{Intensity(Cy3)} \quad (2.1)$$

There are generally two classes of problems associated with gene expression data research: the grouping of genes into similar functional groups and the grouping of samples into similar tumor status. By cancer classification, we normally refer to the latter one.

By allowing monitoring of thousands of genes simultaneously, we may solve three major types of tumor classification problems [19], (i) the identification of new unknown tumor classes using gene expression profile cluster analysis or unsupervised learning, (ii) the classification of malignancies into known classes through discriminant analysis or supervised learning, and (iii) the identification of “marker” genes that characterize the different tumor classes. Here, we are focusing at using data mining tools to solve the (ii) problem, i.e., differentiation of tumor subtypes. In this project, we target developing some unsupervised feature reduction tools which may assist or accelerate the cancer classification process.

2.2 Reviews on machine learning of gene expression data

Classification has long been an extensively researched and fruitful topic in data mining, and it is widely used in machine learning, pattern recognition, medical diagnosis, etc. In the past years, people have applied various classification techniques to gene expression data to either differentiate normal tissues from tumor tissues or one tumor tissue subtype from another. Golub(1999) [24] pioneered the work in tumor classification through his experiment of neighborhood analysis to *leukaemia* data. Alizadeh(2000) [25] has investigated the *lymphoma* dataset. Using hierarchical methods, not only three prevalent lymphoid malignancies(DLBCL, FL, and CLL) were differentiated, but also some previous unknown subtypes of DLBCL were discovered. Ross(2000) [26] has applied hierarchical methods to 61 cell lines which were taken from 8 categories of tumor tissues, and discovered that 6 of them were well separated in the hierarchy. Amir Ben-Dor(2000) [16] has compared several classifications algorithms and evaluated their performance on colon, ovarian, and leukemia datasets, and confirmed Golub's conclusion that the tissue samples can be classified with a relatively small number of genes.

Cho(2003) [27] and Lu(2003) [10] have reviewed the progress recently. Their works involve reviewing a number of traditional and microarray-specified feature selection methods and classification approaches [28]. These methods normally serve to differentiate the malignancies from normal tissues, or one malignancy from other subtypes. The testing results have generally affirmed

the idea of using gene expression data as a cancer prediction or diagnosis tool. However, a few problems need to be addressed:

- most of the currently available data mining algorithms fail in gene expression data clustering or classification problems; the clustering or classification performance degrades with the increase in dimensionality of input data vectors.
- the usually very limited number of gene expression data samples, compared with the ultra high dimensionality of the samples, makes classification algorithm easily trapped in “overfitting” problems.
- the computational complexity of many clustering and classification algorithms goes up exponentially with increase in dimensionality, which prohibits such algorithms from being actually carried out.
- the gene expression data usually contains a large number of noisy or irrelevant features, which creates difficulty for data mining.

2.3 Problem statement

Expression data derived from cDNA chips contains the mRNA level of thousands of genes taken from tissues across many patients. Gene expression data is given in form of matrix G , with rows g_i corresponding to genes, and columns corresponding to samples (tissue from patients). Element $g_{i,j}$ simply represents the expression level of gene i in tissue taken from patient j , ($i = 1, 2, \dots, M, j = 1, 2, \dots, N$), and each row of gene expression $\vec{g}_i$ is called

a gene vector.

$$\underbrace{\left\{ \begin{array}{cccc} g_{1,1} & g_{1,2} & \cdots & g_{1,N} \\ g_{2,1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & g_{i,j} & \vdots \\ g_{M,1} & \cdots & \cdots & g_{M,N} \end{array} \right\}}_{N \text{ samples}} M \text{ genes} \quad (2.2)$$

For classification purpose, the expression data samples are normally labeled and divided into training set T and test set S . The classifier is a function $Class$ with two arguments, T and s , with T denotes the training samples and s is a testing sample from the test set S . The function $Class$ returns a class prediction for sample s . The classification *accuracy* is measured by the number of correct predictions made by the classifier over the test set S using the function $Class$ trained on the training samples T .

Feature reduction is applied to the gene expression data as a preprocessing stage to select most relevant genes. The classifier is built on the training data samples with the selected genes, and the performance of the feature selection and classifier is evaluated by the classification accuracy of the classifier on the testing data samples. An effective feature reduction algorithm shall reduce the computational load of the classification task, while maintaining or improving the data classification accuracy on the testing data. The whole process is illustrated in Fig 2.2.

2.4 Reviews on feature reduction

In the feature reduction research literature, many classical and microarray-specified algorithms have been proposed and experimented [29] [30] [31] [32].

Figure 2.2: Illustration of how feature reduction helps data classification

Here, we try to classify them into several major categories, and analyze their strengths and weaknesses.

2.4.1 Feature selection vs. feature extraction

Feature reduction refers to either *feature selection* [33] [34] or *feature extraction* [35] [36] [37] [38]. Feature reduction refers to the process whereby a data space is transformed into a feature space that, in theory, has exactly the same dimension as the original data space. However, the transformation is designed in such a way that the data set may be represented by a reduced number of “effective” features and yet retain most of the intrinsic information content of the data; in other words, the data set undergoes a *dimensionality reduction*.

Feature selection differs from feature extraction in that *feature selection*

targets to find a subset of features from the original whole feature set, while *feature extraction* tries to derive new features from the original feature set by means of projection or other complex methods.

Feature selection has one natural advantage over feature extraction, i.e. better interpretation preservation. The reason is straight forward: Suppose the original features have some physical meaning, then projection or transformation of the features may lose such direct interpretation. Sometimes, when the interpretation of features is required for the final goal, feature selection is preferred. Another advantage associated with feature selection is the saving in data acquisition cost. When it is known that only a subset of features is necessary for the mining work, we can purposely measure and process the necessary ones only in the future work.

However, feature extraction has its advantages. By means of linear transformation, such as PCA [39] [40] [41] [42], feature extraction normally can reduce the number of feature to 2 or 3 dimensions while retaining the majority of the information, however, this is not normally achievable via feature selection. Also, practically, feature extraction normally has a higher discrimination power, and can help to achieve better clustering or classification accuracy.

2.4.2 Wrapper approach vs. filter approach

Depending on whether the feature reduction process is coupled with a certain induction algorithm, it can be classified as *wrapper* approach and *filter* approach [43]. In supervised machine learning, the classification algorithm is normally presented with the *training* set and *testing* set. The classifica-

tion algorithm is to build a classifier making use of the features and labeling information from the training samples, and verify the predictive power of the classifier using the samples from the testing set. The *wrapper* feature reduction iteratively runs on the training set using various feature subsets and evaluates its performance on each subset, and the feature subset with the highest evaluation score is chosen as the final set on which to run an independent testing set. Since the goal of supervised learning algorithm is to maximize the classification accuracy, classification accuracy is normally chosen as the performance metric to guide the search for optimal feature subsets.

The *filter* approach is normally unsupervised and independent of the induction algorithm. It iteratively runs on various feature subsets, however, the performance or “goodness” of the feature subset is not evaluated by the classification algorithm, but by the properties of the data itself.

2.4.3 Individual gene ranking vs. gene subset ranking

In the literature, many feature reduction techniques have been proposed and applied. Most of the techniques can be classified into two categories: the *individual gene ranking* (IGR) approach and gene subset *ranking* (GSR) approach [10].

Individual gene ranking The individual gene ranking approach first ranks all the genes by criteria of their individual class separation ability [24] [44]. The ability is usually measured by similarity between individual gene vector and the class labels, such as Pearson’s correlation or Euclidean distance.

Pearson's correlation

$$r = \frac{\sum \vec{X}\vec{Y} - \frac{\sum \vec{X}\sum \vec{Y}}{M}}{\sqrt{(\sum \vec{X}^2 - \frac{\sum \vec{X}^2}{M})(\sum \vec{Y}^2 - \frac{\sum \vec{Y}^2}{M})}} \quad (2.3)$$

Euclidean distance

$$r = \sqrt{\sum_{i=1}^M (x_i - y_i)^2} \quad (2.4)$$

where $\vec{X}$, $\vec{Y}$ are feature vector and label vector respectively, and M is the dimension of the vector.

Genes with better ranking status are chosen according to certain threshold or a predefined number of features. IGR approach may be very fast and effective [24], however, there are inherited problems associated with this approach. It only considers features' individual discrimination capability, and ignores the correlation among features. Features which are complementary to work may be missed due to their weak individual class discrimination power. Also, features selected in this way are all correlated with the class label, which adds to the *redundancy* of the features selected.

Gene subset ranking Gene subset ranking aims to find a group genes which are complementary to achieve good classification accuracy [45]. GSR approach normally starts with the whole set of features and gradually discards certain features, monitoring the performance of the remaining subsets of features [46]. Since the performance of the feature subset is normally monitored by a certain classifier, the GSR methods always involves an iterative classification process, which is computationally intensive and very slow. For gene expression data sets, exhaustive subset searching is an NP-hard problem.

2.4.4 Well known methods used in gene expression data feature reduction

In this section, several well known feature reduction methods used in statistics and machine learning are briefly introduced, namely, *correlational ranking*, *entropy feature ranking*, and *PCA for feature reduction*. In the following sections, they will be further analyzed and compared.

Correlational ranking [47] Correlational ranking(CC) is a commonly used supervised feature selection method. Correlational ranking ranks features by their correlational coefficients with the data label. Correlational coefficient, commonly known as Pearson's correlation, measures the strength of the linear relation between variables.

$$\text{Pearson's correlation } r = \frac{\sum \vec{X}\vec{Y} - \frac{\sum \vec{X}\sum \vec{Y}}{M}}{\sqrt{(\sum \vec{X}^2 - \frac{\sum \vec{X}^2}{M})(\sum \vec{Y}^2 - \frac{\sum \vec{Y}^2}{M})}} \quad (2.5)$$

where $\vec{X}$, $\vec{Y}$ are two vectors, and M is the dimension of the vector.

The Pearson correlation first aligns the two vectors to zero mean and unit standard deviation, and then sums up the product moment [48]. If the two vectors are perfectly linear correlated, they should have a Pearson's correlation of 1 or -1; or if the two vectors are totally independent from each other, they should have a Pearson's correlation of 0. With the thought that genes which are highly correlated with the label have more discrimination power, correlational ranking first picks up a predefined number of top ranked genes, or set predefined correlation coefficient thresholds. The filtered genes subset is a candidate for further *wrapper* feature selection methods. Since the filtered subset usually contains less than 100–200 genes, exhaustive search or floating search becomes possible. Correlation ranking together with wrapper

methods have achieved quite good results on many expression data sets. In Golub(1999)'s work [24] on leukemia, a number of 50 informative genes is selected to build the neighborhood classifier, and the selection is based on the *signal to noise ratio(SNR)*, which can be considered a variation of correlational coefficient.

$$Signal/Noise = \frac{mean(\vec{A}) - mean(\vec{B})}{\sqrt{\frac{var_A}{n_A} + \frac{var_B}{n_B}}} \quad (2.6)$$

where $\vec{A}$, $\vec{B}$ denotes substrings of the feature vector corresponding to class A and B.

Entropy feature ranking The algorithm to use entropy as a metric to rank individual feature's discrimination capability is proposed by Dash&Liu [17]. Although the rank is assigned to individual features, it is evaluated based on the performance of feature subset, and this point will be further illustrated later.

Consider a dataset consisting of several distinct clusters, and each sample in the dataset is described by a set of features. Some of the features are relevant, and some are irrelevant. The irrelevant features have no contributions in distinguishing the clusters. The entropy ranking of features is based on the idea that *removing irrelevant features from the feature subset has no impact on underlying concept of the data clusters*. Consider a set of N samples in an M dimensional hyper-space, where each instance $\vec{X}_i$ is represented by a vector of M values (i.e., $x_{i1}, x_{i2}, \dots, x_{iM}$). Removing a variable from an original set of variables (i.e., decreasing the dimensionality of the hyper-space by one) is to project the data onto the $M - 1$ dimensional hyper space. If the distinctness among the clusters remains after projecting samples from M to (M-1) dimensions, then the variable is not important in describing the

underlying structure of the data.

The distinctness of the clusters is measured by the concept of *entropy*. Data has orderly configurations if it has distinct clusters, and has disorderly or chaotic configurations otherwise. From entropy theory, we know that entropy is low for orderly configurations and high for disorderly configurations for the simple reason that there are fewer orderly configurations compared to disorderly configurations.

PCA for feature extraction PCA is a classical linear feature extraction method. It is readily performed by solving an eigenvalue problem, or by using iterative algorithms which estimate principal components [49] [50]. PCA performs an orthogonal transformation of the coordinate system in which we describe our data. The new coordinate values by which we represent our data are called *principal components*. It is often the case that a small number of principal components is sufficient to account for most of the structure in the data. These are sometimes called the factors or latent variables of the data.

Given a set of N centered observations $\vec{X}_k, k=1, \dots, N, \vec{X}_k \in R^M, \sum_{k=1}^N \vec{X}_k = 0$, PCA diagonalizes the covariance matrix

$$C = \frac{1}{N} \sum_{j=1}^N \vec{X}_j \vec{X}_j^T. \quad (2.7)$$

To do this, one has to solve the eigenvalue equation

$$\lambda V = CV \quad (2.8)$$

The solved eigenvectors $\vec{V}_k$, corresponding to eigenvalues $\lambda_k, k=1, \dots, M$, with $\lambda_k \geq \lambda_{k-1}$, form the new coordinate system which gives the maximum variance in each successive coordinate axis.

The lower dimensional representation of the data are obtained by projection to the new coordinates,

$$Y = X * V \quad (2.9)$$

where

$$V = \begin{pmatrix} \vec{V}_1 & \vec{V}_2 & \dots & \vec{V}_m \end{pmatrix} \quad (2.10)$$

Note that in Eq 2.10, we use the the projection matrix V which only contains the first m eigenvectors corresponding to the m largest eigenvalue in the belief that the first m principal directions have summarized most of the variance among the data points. By such a way, the original data points in R^M space are projected into the R^m space while keeping majority of the variance information.

2.4.5 Classification techniques used

To verify the effectiveness of the feature reduction techniques, classifiers are employed. An effective feature reduction technique should improve the classification accuracy of the classifier with fewer but relevant features, or at least, it shall not degrade the classification performance significantly. In the following chapters, we have proposed and implemented several feature reduction techniques and used the SVM classifiers to verify them.

Support Vector Machine SVM was originally introduced by Vapnik et al. It is widely used in data mining applications [51]. Given a number of binary-class training vectors, each tuple $\vec{X}_i$ can be seen as points in high dimensional space, and each point is associated with a class label $c_i \in \{1, -1\}$. In linear separable case, there exists a hyperplane, $(\vec{W}, b)$, where $\vec{W}$ is a vector and b

is a scalar such that for each point $\vec{X}_i$,

$$\vec{W}^T \vec{X}_i + b \geq 0 \quad \text{for } c_i = 1, \quad (2.11)$$

$$\vec{W}^T \vec{X}_i + b \leq 0 \quad \text{for } c_i = -1. \quad (2.12)$$

Hyperplanes satisfying such criteria are many, SVM aims to find a hyperplane with maximum margin that separates the positive labeled points from the negative ones. Those points that lie closest to this max-margin hyperplane are defined as *support vectors* and the classifier only makes use of these support vectors to classify test vectors.

In the non-linear separable case, the data points are first mapped into higher dimensional feature space from input space [10] Let's define mapping $\Phi : I \subseteq R_m \rightarrow F \subseteq R_M$, and $\langle \vec{X}, \vec{Y} \rangle$ denotes the dot product of vector $\vec{X}$ and $\vec{Y}$. Then the new max-margin hyperplane is defined as the plane that maximizes the following function

$$\gamma = \min_{i=1}^n c_i (\langle \vec{W}, \Phi(\vec{X}_i) \rangle - b) \quad (2.13)$$

where the value $(\langle \vec{W}, \Phi(\vec{X}_i) \rangle - b)$ is the distance between the point $\vec{X}_i$ and the hyperplane in the feature space. The sign of $(\langle \vec{W}, \Phi(\vec{X}_i) \rangle - b)$ denotes which side the point resides with respect to the hyperplane. Multiplied by the label c_i , the product is positive when $\vec{X}_i$ is correctly classified and negative if $\vec{X}_i$ is misclassified [10].

Multi-class classification The SVM is specially tailored for binary-class classification problem. In the case of multi-class problems, we split the problem into k binary-classification problems, where k equals to the number of classes, and each class is classified versus all other classes in the dataset.

The classification accuracy is calculated by adding up the correctly classified samples over all classes.

Cross validation The gene expression dataset usually contains very few data samples, and to analyze the performance of the classification, we need to estimate the generalized classification error. Cross validation is such a technique based on methods of re-sampling. In this paper, we use *Leave-one-out* cross-validation.

In Leave-one-out cross-validation, the classifier is trained k (k equals the number of samples) times, each time leaving out one sample from training, but using only the omitted sample to compute whatever error criterion that interests us.

Chapter 3

Entropy ranking for feature selection

Entropy ranking for feature selection was proposed by Dash&Liu [17]. The basic idea of entropy ranking is mentioned in Section 2.2. Entropy ranking is a purely unsupervised feature selection strategy which originates from the definition of entropy in information theory. In information theory, given a set of objects, the entropy is defined to be low for orderly configurations, and high for disorderly configurations [52]. In other words, entropy is a description of the degree of randomness. Claude E. Shannon [53] defined the measure of entropy

$$H = - \sum P_i \log_2 P_i \quad (3.1)$$

where P_i denotes the probability of appearance of object i . This definition, when applied to an information source, could determine the minimum channel capacity required to reliably transmit the source as encoded binary digits. Shannon's entropy [53] measure came to be taken as a measure of the

uncertainty about the realization of a random variable.

3.1 Entropy description of cluster

The concept of entropy is analogized in the field of clustering of data points by Dash&Liu [17]. In the case of clustering, distinct cluster separation is defined as orderly configuration, and disorderly configuration otherwise. To make clusters of data samples distinct, the pairwise distance shall be either very large (in case of different clusters) or very small (in case of within the same cluster). Then entropy value can be assigned to pairwise distance among data samples: the entropy should be low for very close or very distant pairwise instances, and very high for mean pairwise distance. Eq 3.2 was proposed by Dash&Liu [17] to formulate the concept,

$$e = -[S_{i,j} \times \log S_{i,j} + (1 - S_{i,j}) \times \log(1 - S_{i,j})]$$

where $S_{i,j}$ denotes the pairwise distance between point i and j . Fig 3.1 illustrates how the entropy e varies with pairwise distance $S_{i,j}$.

For a data set of N instances in M dimensions, the entropy measure is given as:

$$E = - \sum_{i=1}^N \sum_{j=1}^N (S_{i,j} \times \log S_{i,j} + (1 - S_{i,j}) \times \log(1 - S_{i,j})) \quad (3.3)$$

where the similarity S and distance D are defined as follows for continuous data,

$$S_{i,j} = e^{-\alpha \times D_{i,j}} \quad (3.4)$$

$$D_{i,j} = \left[\sum_{k=1}^M \left(\frac{g_{k,i} - g_{k,j}}{\max_k - \min_k} \right)^2 \right]^{1/2} \quad (3.5)$$

Figure 3.1: Illustration of how entropy varies with pairwise distance

where $g_{k,i}$ denotes the value of (patient)instance i on (gene)dimension k as illustrated in the gene matrix in Eq 2.2, and $D_{i,j}$ denotes the normalized distance between instance i and j .

3.2 Using entropy to rank features

Since the calculation of Eq 3.3 does not involve any label information of data samples, it is a purely unsupervised description of clusters. Using this entropy definition Eq 3.3 as an unsupervised description of the cluster, we can find subsets of features which make distinct separation of clusters. The entropy ranking takes a Gene subset ranking(GSR) approach. To demonstrate the rationale to find the effective features in a group wise way rather than individual wise way, Dash&Liu used a graph shown in Fig 3.2 to illustrate

the two dimensional clustering problem in two scenarios:

Figure 3.2: Illustration of features' discrimination capability

Scenario 1: A single feature is important by itself only: Consider Fig (a), feature F2 is uniformly distributed, while feature F1 takes values in two separate ranges. It can be clearly seen that F1 has more discrimination power than F2.

Scenario 2 Two features are necessarily important and any individual feature is useless in defining clusters: Consider Fig (b), both F1 and F2 are uniformly distributed. It can be seen that both F1 and F2 are necessary for clustering and any feature alone is *useless*.

Therefore the entropy ranking is rendered in a floating search fashion. The search starts with the subset of all features loaded, and drops less informative features iteratively. In each iteration, the entropy is calculated for the data samples on the current features subset with each feature taken out. The entropy shall be less affected with the removal of a less important feature, so the feature without which gives the lowest entropy value is deemed as the worst feature in this subset and taken out at the end of current iteration. After this, a descending ranking of features based on their informativeness is

obtained, i.e., the 1st feature shall be the one with the most information.

In this project, based on the idea of floating search, we feed subsets of features to a classifier. The subset may start with an empty feature set, and features are added in continuously in descending ranking order. The classification performance is expected to grow with the increment in the number of features, and achieve its best at some point, and may start to drop with more features added in.

3.3 A fast implementation

First two sections have highlighted the working principle of entropy based feature selection [17]. However, the generation of a feature ranking list is a highly computationally complex process. Given a dataset with dimensionality M , the feature ranking takes M iterations, and in each iteration, the entropy value shall be calculated m times, where m equals the number of features in the current feature subset. So the computational complexity will be $O(M^2)$.

$$M + (M - 1) + \dots + m + \dots + 1 = (M + 1) \times M/2 = M^2/2 + M/2 \quad (3.6)$$

In case of gene expression data, the dimensionality M is usually in the range of several thousands, and $O(M^2)$ becomes extremely large. Since the number of iterations to calculate entropy is fixed, we need to deliberately simplify the steps to calculate entropy.

From Eq 3.3, formulation of entropy contains items of Eq 3.4, and S is chained to D in Eq 3.5. Note the normalized Euclidean distance between data samples, Euclidean distance is symmetrical for pairwise distance $D_{i,j}$

and $D_{j,i}$, using this special property, we can solely calculate the distance $D_{i,j}$, with $(i \leq j)$, which saves half the computational cost.

Note in Eq 3.5, the normalization items max_k and min_k , are just the maximum and minimum of the gene vector kas defined in Eq 2.2, so the item $(max_k - min_k)$ does not change with the removal of any gene vector. Therefore, we will try to make the data normalized before any distance and entropy calculation is carried out, thus to save the repetitive computational cost.

$$g_{k,i} = \frac{g_{k,i}}{max_k - min_k} \quad (3.7)$$

After we have done the normalization, Eq 3.5 can be simplified as,

$$Dsums_{i,j} = \sum_{k=1}^M (g_{k,i} - g_{k,j})^2 \quad (3.8)$$

$$D_{i,j} = Dsums_{i,j}^{1/2} = \left[\sum_{k=1}^M (g_{k,i} - g_{k,j})^2 \right]^{1/2} \quad (3.9)$$

which is the standard Euclidean form.

The most time consuming part of the entropy calculation is the calculation of $Dsums$ in Eq 3.8. Expand $Dsums$, we get

$$Dsums_{i,j} = (g_{1,i} - g_{1,j})^2 + (g_{2,i} - g_{2,j})^2 + \dots + (g_{k,i} - g_{k,j})^2 + \dots + (g_{M,i} - g_{M,j})^2 \quad (3.10)$$

Each element of matrix $Dsums$ is a sum of squared differences in every feature. In each iteration, the removal of certain feature has the effect to eliminate the item in Eq 3.10 associated with the feature. Therefore, in each iteration, we can calculate the value of $Dsums_{i,j}$ and store the matrix. To calculate the entropy of the data sets with the removal of feature k , we just minus the item $(g_{k,i} - g_{k,j})^2$ from each element in the stored matrix. In

this way, in each iteration we only need to do the summation in Eq 3.10 once, and the computation cost is greatly reduced. The proposed entropy calculation procedure does not reduce the number of iterations, however, it greatly shortens the running time of each iteration. The effectiveness will be demonstrated in the experimental part in Section 3.4.

3.4 Experiments

3.4.1 Dataset

The proposed feature selection algorithm was implemented and tested on three microarray datasets, *Leukemia*, *Lymphoma*, and *Colon*. All these datasets are publicly available for downloading. The links to the datasets are summarized in Table 3.1.

[Leukemia]: Leukemias are primary disorders of bone marrow. They are malignant neoplasms of hematopoietic stem cells. The total number of genes to be tested is 7129, and number of samples to be tested is 72, which are all acute leukemia patients, either acute lymphoblastic leukemia (ALL) or acute myelogenous leukemia (AML).

[Lymphoma]: Lymphoma is a broad term encompassing a variety of cancers of the lymphatic system. Total number of genes to be tested is 4026 and the number of samples to be tested is 62. There are all together three types of lymphomas. The first type, Chronic Lymphocytic Lymphoma(CLL) has 11 patients, the second type Follicular Lymphoma(FL) has 42 and the third type Diffuse Large B-cell Lymphoma(DLBCL) has 9. From an oncology point of view, DLBCL could be further explored to its two subtypes, however,

in this project, we have limited our discussions to treat DLBCL as a single category.

[Colon]: Colon tumor is a disease in which cancerous growths (tumors) are found in the tissues of the colon. This dataset contains 62 samples. Among them, 40 tumor biopsies are from tumors (labeled as “negative”) and 22 normal (labeled as “positive”) biopsies are from healthy parts of the colons of the same patients. The total number of genes to be tested is 2000.

Characteristics of the datasets are summarized in Table 3.2.

Table 3.1: Sources of datasets	
Dataset	website
Leukemia	http://www.broad.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43
Lymphoma	http://lmpp.nih.gov/lymphoma
Colon	http://microarray.princeton.edu/oncology/affydata/index.html

Table 3.2: Gene expression datasets				
Dataset	Classes	No. of genes	No. of Samples	Origin
Leukemia	2	7129	72	Golub et al. [24]
Lymphoma	3	4026	62	Alizadeh et al. [25]
Colon	2	2000	62	Alon et al. [54]

3.4.2 Software and environment

The whole program is implemented in Matlab(version 6.5) environment. To improve the running speed of the entropy ranking, we have purposely written

the optimized entropy calculation procedure in C code. The programming environment is Microsoft Visual C++ Ver 6.00. The hardware configuration is Intel P4 2.4GHz CPU, and 1.25 Gb RAM. The operating system is Microsoft Windows XP.

3.4.3 Dataset preprocessing and normalization

The data preprocessing and normalization steps are illustrated in Figure 3.3. The raw gene expression datasets are normally provided by biological research labs, and such datasets usually contain many text biological attributes which are irrelevant to classification purpose and also can not be directly imported to Matlab environment. Therefore, we have manifested the datasets into pure matrix form by removing such fields. The processed dataset files contain matrix with genes in rows and samples in columns as described in Section 2.3.

Figure 3.3: Data preprocessing and normalization steps

Note that, for *leukemia* dataset, the raw data has more than 7000 features, due to the memory limitation, we have used the preprocessing techniques as described in Section 5.1.1 to filter the data feature size from 7129 to 2978.

Data normalization

Features(genes) are normalized to have zero mean and standard deviation of one.

```

for i=1:no_var
    mean_row = mean(rdata(i,:));
    rdata(i,:) = rdata(i,:) - repmat(mean_row,1,no_sam);
end
for i=1:no_var
    rdata(i,:) = rdata(i,:)/std(rdata(i,:));
end

```

3.4.4 SVM parameter settings

The working principle of Support Vector Machine(SVM) has been described in Section 2.4.5. The SVM we used is the SVM^{light} software package version 5.00 by Joachims [55]. It is available at <http://svmlight.joachims.org/>. The SVM is an executable program, we have used the script from Anton Schwaighofer to interface it with Matlab [56].

SVM kernel options for SVM^{light} [55]

-t int	- type of kernel function:
0:	linear (default)
1:	polynomial $(sa * b + c)^d$
2:	radial basis function $\exp(-gamma a - b ^2)$
3:	sigmoid $\tanh(s a*b+c)$
4:	user defined kernel from kernel.h
-d int	- parameter d in polynomial kernel
-g float	- parameter gamma in rbf kernel
-s float	- parameter s in sigmoid/poly kernel
-r float	- parameter c in sigmoid/poly kernel
-u string	- parameter of user defined kernel

To define an SVM with RBF kernel($\gamma = 0.5$) in Matlab via the interface by Anton Schwaighofer [56].

```
options = svmlopt('ExecPath',' ./svm-win',' Kernel', 2,' Kernel Param', 0.5);
```

To train the SVM with Leave-one-out methodology,

```

svmlwrite(train, sampleddata(:,1:(end-1)), sampleddata(:,end));
svmlwrite(test, testdata(:,1:(end-1)), testdata(:,end));
svm_learn(options, 'train.dat', 'model');
svm_classify(options, 'test.dat', 'model','predictions');
pred(j,no_f) = load('predictions');
if(testdata(:,end)==sign(pred(j,no_f)))
    correct_count=correct_count+1;
    r_or_w(j,no_f) = 1;
end

```

The function *svmlwrite*(...) converts all the data produced by Matlab to files, 'train.data' and 'test.dat', which can be recognized by *SVM^{light}* executables, and the function *SVM-learn*(...) trains the SVM with the 'train.data' to build the classifier and stores the parameters in the 'model' file. The function *svm-classify*(...) reads such parameters, tries to classify the unlabeled data in 'test.data', and stores the prediction results in file 'predictions'.

3.4.5 LOOV and accuracy statistics

As introduced in Section 2.4.5, we have used Leave-one-out strategy to find the generalized classification error. Each sample in the dataset is taken as the test sample for once with all the rest as training samples. The final accuracy counting is taken to be the number of correctly identified samples.

```
for j=1:no_sam
    .....
    testdata = t_data(j,:);
    sampledata= [t_data(1:(j-1),:);t_data( (j+1):end,:)];
    .....
end
```

For multiclass classification case, each class of samples is classified versus all other classes.

```
for t=1:max(label)
    bin_label=label;
    index=find(bin_label ==t);
    bin_label(index)=-1;
    index=find(bin_label==t);
    bin_label(index)= 1;
    .....
    pred = load('predictions');
    if(testdata(:,end)==sign(pred)==t)
        correct_count=correct_count+1;end
    end
end
```

3.4.6 Results

The algorithm is tested against three microarray datasets. For each of the datasets, the entropy ranking algorithm will generate a ranking list to represent the informativeness of the features. The effectiveness of the ranking is verified by SVM classifier. Starting with a feature subset of the most informative features, the data samples are classified with more features added in descending order from the ranking list. The accuracy of classification is continuously monitored. The SVM classification accuracy is plotted versus the number of features employed in Fig 3.4 - Fig 3.6. Since the SVM classifier does not converge until a substantial amount of useful features are feed, we only plot the classification accuracy starting from 15 features. It is observed that initially the classification accuracy increases with the increase in the number of features, and reaches the peak at somewhere between 30 - 100. Beyond this point, the accuracy starts to fall. This is consistent with our previous assumption about feature selection, i.e., the majority of the features in the original feature set are irrelevant to classification, and the inclusion of irrelevant features will have negative effect on classification performance.

To demonstrate the effectiveness of our proposed optimization procedure, we have compared its running time with the original implementation. Table 3.3 shows the time consumed to generate the feature ranking list using our proposed procedure and the original procedure.

To demonstrate that the feature selection process helps to boost the classification performance, we have compared the classification accuracy with and without entropy ranking feature selection. The results are listed in Table 3.4.

We have also compared our feature selection effectiveness with other clas-

sical feature selection algorithms. The results are summarized in Table 3.5. Since the comparison is focused on the effectiveness of the feature selection process, only results that employed the same classifier with the same kernel will be chosen. Recently, Cho et al [27] have done a survey of microarray classification techniques on these microarray datasets, also using SVM(RBF kernel). We have extracted some of their testing results for comparison. The feature reduction techniques being compared include unsupervised methods, such as principal components(PCA), and supervised methods, such as signal to noise ratio(SNR) and correlational coefficient(CC). Also PCA can be considered as a GSR approach, while SNR and CC belong to IGR approaches. From the results, we could observe that, entropy ranking has outperforms the classical feature selection techniques for the same number of features. However, to achieve the best performance, it requires much more number of features.

The survey in [27] reported testing results with 25 features. We also observe in Table 3.6 that, prior to the peak performance, entropy could also achieve quite good results with just 25 top features used. This could be explained as that the first few top ranked features are the most informative features in the datasets using entropy ranking.

Table 3.3: Comparison of time consumed to generate complete ranking list with and without optimization

Dataset	No. of features	time (with opt.)	time (without opt.)
Leukemia	2978	160 mins	more than 1440 mins
Lymphoma	4026	215 mins	more than 1440 mins
Colon	2000	52 mins	more than 1440 mins

Figure 3.4: Performance of entropy ranking on leukemia dataset

3.4.7 Discussions

In this chapter, we have reported our results on using a recently proposed feature selection algorithm *entropy ranking* for microarray data classification. The feature ranking ranks each feature’s discrimination capability by its contribution to the cluster separation. It is a purely unsupervised and subset ranking approach. The original implementation is computationally prohibitive for ultra high dimensional microarray data. We have made several optimizations in implementation, which greatly reduce the time consumed for entropy calculation. The fast feature ranking algorithm is implemented and tested on three publicly available microarray datasets, and it is observed that this algorithm has shown advantage in boosting classification performance compared to several other classical feature selection techniques. As a

Figure 3.5: Performance of entropy ranking on lymphoma dataset

Figure 3.6: Performance of entropy ranking on colon dataset

purely unsupervised feature dimension reduction technique, entropy ranking has outperformed some classical supervised methods, although with more features needed. However, the actual running time of this algorithm is still non-trivial and the number of features required is also quite large.

Table 3.4: Comparison of classification accuracy with and without entropy
rankinn feature selection

Dataset	Feature selection	No. of features	Accuracy
Leukemia	N.A.	2978	95%
	Entropy ranking	95	97%
Lymphoma	N.A.	4026	88%
	Entropy ranking	30	84%
Colon	N.A.	2000	80%
	Entropy ranking	90	85%

Table 3.5: Comparison of classification accuracy between Entropy ranking
with other classical methods using SVM(RBF kernel)

Dataset	Feature selection	No. of features	Accuracy
Leukemia	Entropy	95	97%
	PCA	25	79%
	SNR	25	59%
	CC	25	85%
Lymphoma	Entropy	30	84%
	PCA	25	60%
	SNR	25	76%
	CC	25	56%
Colon	Entropy	90	85%
	PCA	25	65%
	SNR	25	65%
	CC	25	65%

Table 3.6: Comparison of classification accuracy between Entropy ranking with other classical methods with 25 features using SVM(RBF kernel)

Dataset	Feature selection	No. of features	Accuracy
Leukemia	Entropy	25	84%
	PCA	25	79%
	SNR	25	59%
	CC	25	85%
Lymphoma	Entropy	25	79%
	PCA	25	60%
	SNR	25	76%
	CC	25	56%
Colon	Entropy	25	74%
	PCA	25	65%
	SNR	25	65%
	CC	25	65%

Chapter 4

Locally linear embedding – a feature extraction approach

In Chapter 3, a feature selection algorithm based on concept of entropy [17] has been analyzed and implemented. Now, we are going to switch our work to the *feature extraction* approach. A customized Locally Linear Embedding(LLE) [18] algorithm based on fractional metrics will be described and experimented with. In this chapter, the original LLE algorithm proposed by Roweis&Saul will be discussed in details, and then we will show how the fractional metrics could be incorporated in the implementation. Lastly, in the experiment part, we will present the testing results.

4.1 Locally linear embedding

Locally linear embedding(LLE) was first proposed by Roweis&Saul [18] as a nonlinear dimensionality reduction algorithm for face recognition applications. Unlike most classical linear dimensionality reduction methods, LLE

does NOT make any linear assumption about the distribution of data points. LLE assumes that data points lie on curved/distorted manifolds in the high dimensional space, and data dimensionality reduction could be realized by proper unfolding of the manifolds. The actual process of unfolding carries out in a 'local-to-global' way, i.e., by carefully preserving the neighborhood geometry of each local patch of the manifolds, the global geometry of the data points can be recovered in lower dimensional space through the connection of overlapping local patches.

Specifically, each data point along with its k nearest neighbors forms one piece of these local patches. The local geometry of this patch is to be calculated and stored in a set of coefficients, and a new set of lower dimensional data points are to be found which precisely preserve such geometry characterized by such coefficients. Given a dataset which consists of N data points in real value, each of dimensionality D . Assume that N is sufficiently large, so that the manifold which the data points lie on is well sampled by these N samples. The local geometry of the data points is captured by a set of linear coefficients W which linearly reconstruct such data points from their neighbors. Note that the linear relationship is only assumed and applied in the *local* geometry, and NOT in the global geometry. This is fairly reasonable given the fact that the local patch is sufficiently small and relatively flat. The following function has been proposed by Roweis&Saul to measure the reconstruction cost [18],

$$\epsilon(W) = \sum_i^N \left| \vec{X}_i - \sum_j^k W_{ij} \vec{X}_j \right|^2 \quad (4.1)$$

The reconstruction cost $\epsilon(W)$ is a sum of all the squared distances be-

tween all data points and their k neighbors. The reconstruction weight W_{ij} summarizes the contribution from the j th data point to i th data point to reconstruct the local patch centered around the i th data point. The proper weight coefficient W shall minimize the total reconstruction cost $\epsilon(W)$. This problem could be solved with two constraints, first, the local patch for each point only includes its k nearest neighbors, therefore if data point $\vec{X}_j$ does not belong to the k nearest neighbors of data point $\vec{X}_i$, the reconstruction weight W_{ij} is defined to be 0. Second, the reconstruction weights associated with each data point shall sum to one: $\sum_{j=1}^k W_{ij} = 1$. The second constraint serves to make sure that the reconstruction weights are invariant to rotations, re-scalings, and translations of the data points, which will be shown in Section 4.2. The optimal weights W subject to these constraints are found by solving a least-squares problem.

With the reconstruction weights W calculated, each high dimensional point $\vec{X}_i$ can be mapped to a new vector $\vec{Y}_i$ in lower dimensional space by minimizing a new reconstruction cost function proposed by Roweis&Saul [18],

$$\Phi(Y) = \sum_i^N \left| \vec{Y}_i - \sum_j^k W_{ij} \vec{Y}_j \right|^2 \quad (4.2)$$

The reconstruction weights W are kept the same in this new function defined in Eq 4.2, therefore, by minimizing this new reconstruction cost $\Phi(Y)$, a group of lower dimensional vectors $\vec{Y}_i$ can be found in the lower d dimensional space which will share the same local geometry as the high dimensional points $\vec{X}_i$. Mathematically, such vectors could be found by solving a sparse $N \times N$ eigenvalue problem as further illustrated in Section 4.4.

The actual process of LLE [18] is as follows:

- (1) Assign neighbors to each data point $\vec{X}_i$ (for example by using the k nearest neighbors).
- (2) Compute the weights W_{ij} ($j = 1, 2, \dots, k$) that best linearly reconstruct $\vec{X}_i$ from its neighbors, solving the constrained least-squares problem in Eq 4.1.
- (3) Compute the low-dimensional embedding vectors $\vec{Y}_i$ with dimension of d which are best reconstructed by W_{ij} , minimizing Eq 4.2 by solving a linear algebra problem.

The programme flow is illustrated in Figure 4.1 where q , W , and M are defined in Eq 4.39, Eq 4.36 and Eq 4.26.

The locally linear embedding algorithm has many intrinsic advantages over many traditional data dimensionality reduction techniques. 1) LLE does not make linear assumption about the data distribution, therefore it is capable of detecting non-linear subspace structure within the data. 2) The LLE implementation has very few parameters and they are very easy to tune. 3) It is a purely unsupervised feature extraction technique, the data dimensionality reduction process does not involve any classifier, and it is non-iterative, therefore it is very fast. 4) The LLE implementation does not rely on the data sample similarity measurement metrics, and any similarity criteria may be incorporated.

D.Gering [57] has done the mathematical proof about LLE's invariant properties to scaling, rotation and translation, and has provided derivations of the reconstruction weights and lower dimensional coordinates in [57]. In Section 4.2 to Section 4.4, we will include those materials here by following

Figure 4.1: Flow chart of LLE programme

the steps of derivation in [57], but using notations from LLE's original work for the sake of completeness.

4.2 Invariance to scaling, rotation and translation

We have mentioned in Section 4.1 about the invariant properties of the LLE algorithm to scaling, rotation and translation of data points. The invariant properties are necessary because the weights W used to construct each data point in high dimensional space are the same ones used to reconstruct the local patches in lower dimensional manifolds, therefore, the reconstruction weights must be invariant to any linear mapping [18].

The reconstruction weights shall minimize the cost $\epsilon(W)$ in Eq 4.1, which is a sum of non-negative items. Therefore, the reconstruction weight associated with each data point shall satisfy the following relationship:

$$\|\vec{X}_i - \sum_{j=1}^k w_j \vec{X}_j\| = 0 \quad (4.3)$$

$$\vec{X}_i = \sum_{j=1}^k w_j \vec{X}_j \quad (4.4)$$

4.2.1 Scaling and rotation

Consider multiplying each input data point $\vec{X}_i$ by the rotation and scaling matrix R , it can be shown that the weights, W , corresponding to the original data points equal the weights, V , corresponding to the rotated and scaled points [57].

Since weights V shall minimize the reconstruction cost for data RX , it must also satisfy Eq 4.4, then,

$$R\vec{X}_i = \sum_{j=1}^k v_j R\vec{X}_j = v_1 R\vec{X}_1 + v_2 R\vec{X}_2 + \dots + v_k R\vec{X}_k \quad (4.5)$$

$$R^{-1}R\vec{X}_i = \sum_{j=1}^k v_j R^{-1}R\vec{X}_j \quad (4.6)$$

$$\vec{X}_i = \sum_{j=1}^k v_j \vec{X}_j \quad (4.7)$$

$$\sum_{j=1}^k w_j \vec{X}_j = \sum_{j=1}^k v_j \vec{X}_j \quad (4.8)$$

$$W = V \quad (4.9)$$

4.2.2 Translation

Suppose vector $\vec{T}$ is added to each input point $\vec{X}_i$ and its k neighbors:

$$\vec{X}_i + \vec{T} = \sum_{j=1}^k v_j (\vec{X}_j + \vec{T}) \quad (4.10)$$

$$\sum_{j=1}^k w_j \vec{X}_j + \vec{T} = \sum_{j=1}^k v_j \vec{X}_j + \sum_{j=1}^k v_j \vec{T} \quad (4.11)$$

Given Eq 4.9, the first item on both sides of Eq 4.11 could be balanced. Then, to hold the above identity, all rows of the weight matrix ($V = W$) must sum to 1, which is the first constraint $\sum_{j=1}^k w_j = 1$ we have mentioned earlier. Therefore, the weights can be invariant to translations of the data point and its neighbors if and only if the following constraint hold [57]:

$$\sum_{j=1}^k w_{ij} = 1 \quad \text{for any } i \quad (4.12)$$

4.3 Solution for reconstruction weights

To find the reconstruction weights W , we need to solve the cost function Eq 4.1. Making use of the constraint Eq 4.12, we can rewrite the cost term for data point $\vec{X}_i$ as,

$$\epsilon^i(W) = \left| \vec{X}_i - \sum_{j=1}^k W_{ij} \vec{X}_j \right|^2 \quad (4.13)$$

$$= \left| \sum_{j=1}^k W_{ij} (\vec{X}_i - \vec{X}_j) \right|^2 \quad (4.14)$$

Use X to represent the matrix of columns $\vec{X}_i$ repeated k times, and $\vec{W}_i$ to be a column vector by transposing the i th row vector of W , and Z to be matrix of columns formed by neighbors of point $\vec{X}_i$, we write the above equation in simpler form,

$$\epsilon^i(W) = ((X - Z)\vec{W}_i)^T((X - Z)\vec{W}_i) \quad (4.15)$$

$$= \vec{W}_i^T (X - Z)^T (X - Z) \vec{W}_i \quad (4.16)$$

$$= \vec{W}_i^T S \vec{W}_i \quad (4.17)$$

$$S = (X - Z)^T (X - Z) \quad (4.18)$$

This minimization problem can be solved using Lagrange multiplier. Adding the identity constraint in Eq 4.12, the Lagrange multiplier is constructed,

$$L(\vec{W}_i) = \vec{W}_i^T S \vec{W}_i + \lambda (\vec{W}_i^T I - 1) \quad (4.19)$$

The problem is then to find the zero-gradient condition of the multipliers.

$$\frac{\partial L}{\partial \vec{W}_i} = 0 = 2S\vec{W}_i + \lambda I \quad (4.20)$$

$$S\vec{W}_i = cI \quad (4.21)$$

By above equation the solution of $\vec{W}_i$ may be found by inverting S , which is a computationally complex linear algebra problem. However, in practice, it could be solved by keeping the constant c arbitrarily as 1, and then scale $\vec{W}_i$ to sum to 1 [57].

4.4 Solution for lower-dimensional coordinates

Define $\vec{I}_i$ to be the i th column of the identity matrix, Y to be the matrix in columns $[Y_1 \ Y_2 \ \dots \ Y_N]$, and $\vec{W}_i$ to be a column vector by transposing the i th row vector of W ,

$$\Phi(Y) = \sum_i^N \left| \vec{Y}_i - \sum_j W_{ij} \vec{Y}_j \right|^2 = \sum_i^N \left| Y \vec{I}_i - Y \vec{W}_i \right|^2 = \sum_i^N \left| Y (\vec{I}_i - \vec{W}_i) \right|^2 \quad (4.22)$$

Given a matrix A , denote its i th column vector as $\vec{A}_i$, the following identity can be found easily,

$$\sum_i \left| \vec{A}_i \right|^2 = \sum_i \vec{A}_i^T \vec{A}_i = \text{trace}(A^T A) = \left| A \right|^2 \quad (4.23)$$

Therefore Eq 4.22 can be simplified as,

$$\Phi(Y) = \left| Y(I - W) \right|^2 \quad (4.24)$$

Again making use of the identity $\left| A \right|^2 = \sum a_{ij}^2 = \sum a_{ji}^2 = \left| A^T \right|^2$, $\Phi(Y)$ can be further transformed as,

$$\Phi(Y) = \left| (I - W)^T Y^T \right|^2 = \text{trace}(Y(I - W)(I - W)^T Y^T) = \text{trace}(Y M Y^T) \quad (4.25)$$

while

$$M = (I - W)(I - W)^T \quad (4.26)$$

It appears that if W were equal to identity matrix, then M would be a zero matrix, and $\Phi(Y)$ would vanish to zero. However, given the constraint that a point can not be its own neighbor, this will never appear.

The minimization problem of $\Phi(Y)$ is solved in the same way as the minimization of $\epsilon^i(W)$ using Lagrange multipliers. To uniquely determine the Y in the lower dimensional space, two constraints must be enforced to reduce the degree of freedom of Y by defining its origin and orientation in the lower dimensional space,

$$\sum_i \vec{Y}_i = 0 \quad (4.27)$$

$$\frac{1}{N-1} \sum_i \vec{Y}_i \vec{Y}_i^T = I \quad (4.28)$$

The problem is then solved by construction of a Lagrange multiplier subject to the above constraints [57],

$$L(Y) = YMY^T + \lambda(YY^T - (N-1)I) \quad (4.29)$$

To find the zero-gradient condition of the Lagrange multiplier, its derivative is computed,

$$\frac{\partial L}{\partial Y} = 0 = 2MY^T + 2\lambda Y^T \quad (4.30)$$

$$MY^T = \lambda Y^T \quad (4.31)$$

Then, L can be minimized with the columns of Y^T (rows of Y) being the eigenvectors associated with the lowest eigenvalues of M . The first constraint Eq 4.27 can be imposed by discarding the eigenvectors associated with eigenvalue 0 (free translation). To find the lower dimensional vector $\vec{Y}_i$ in the d -dimensional space, only eigenvectors $\vec{U}_i$ associated with the bottom

d nonzero eigenvalues are kept. These produce the d rows of the d -by- N output matrix Y :

$$Y = \begin{bmatrix} | & | & & | \\ \vec{Y}_1 & \vec{Y}_2 & \dots & \vec{Y}_N \\ | & | & & | \end{bmatrix} = \begin{bmatrix} - & \vec{U}_1 & - \\ - & \vec{U}_2 & - \\ & \vdots & \\ - & \vec{U}_d & - \end{bmatrix} \quad (4.32)$$

Matrix W is of size $(N \times k)$, where N is the number of data samples, and k is the number of neighbor points selected to form each local patch, so is matrix $(I - W)$, thus M in Eq 4.26 will be a matrix of $(N \times N)$. Solution of lower dimensional vectors resolves to a linear algebra problem of eigen-decomposition of a $(N \times N)$ matrix. For microarray data, N is usually within the range of 40 - 100, which makes the solution very fast.

4.5 Fractional metrics

Charu C. Aggarwal et al. [58] have discussed the behavior of relative contrast with different distance metrics, and proposed to use fractional metrics for high dimensional data.

Consider a dataset consisting of n points, where each point is described by a d -dimensional vector. Define distance measurement

$$dist = \left[\sum_{i=1}^d (x_i - y_i)^{1/k} \right]^k \quad (4.33)$$

between two data points x and y . And relative contrast

$$r_j = \frac{Dmax_{d,j}^k - Dmin_{d,j}^k}{Dmin_{d,j}^k} \quad (4.34)$$

for each point j using the distance measurement defined in Eq 4.33.

where $Dmax_{d,j}^k$ denotes the farthest distance from all other points to the point j we consider, $Dmin_{d,j}^k$ denotes the nearest distance similarly. When $k=1/2$, $dist$ is the Euclidean distance, which is a special case of fractional metrics.

For most of the data mining algorithms, the distance measurement metric is necessary and crucial. Unless otherwise specified, most of the time Euclidean distance is chosen. However, in the high dimensional space, the contrast nature of distance measurement become shattered with increase of dimensionality.

It has been proved [58] that the relative contrast r will degrade with the increase of dimensionality d . To make the distance metrics a more meaningful proximity measure, we have to take a practical approach, which relaxes the k to take value other than $1/2$ as for Euclidean distance. The relative contrast r on the specific dataset can be used as a guide for choice of a specific K .

$$K = \arg \max_k \sum_{j=1}^N \frac{Dmax_{d,j}^k - Dmin_{d,j}^k}{Dmin_{d,j}^k} \quad (4.35)$$

In implementation, we can fine tune the fraction value k in the range $0.1 \sim 0.5$ by step of 0.1 , and pick up the one which reaches peak performance.

4.6 LLE from pairwise distance

Note that the reconstruction cost function Eq 4.1 is a sum of cost items associated with each single data point. The cost associated with each point depends on the geometry of the neighborhood of the point only, therefore, we can take care of them individually. In an explanatory report, Saul&Roweis

[59] has expressed the reconstruction cost in a different form as in Eq 4.36 and Eq 4.37. The reconstruction cost $\epsilon(W)$ in Eq 4.1 is expressed as the sums of individual reconstruction costs $\epsilon^i(W)$ from each data point [59],

$$\epsilon^i(W) = \left| \sum_{j=1}^k W_{ij} (\vec{X}_i - \vec{X}_j) \right|^2 = \sum_{j=1}^k \sum_{m=1}^k W_{ij} W_{im} Q_{jm}^i \quad (4.36)$$

where

$$Q_{jm}^i = (\vec{X}_i - \vec{X}_j)^T (\vec{X}_i - \vec{X}_m) \quad (4.37)$$

Figure 4.2: illustration of triangle

The original cost function Eq 4.36 contains the item Q_{jm}^i , which is expressed in the form of the inner product of two vectors as in Eq 4.37. Data point X_i along with two of its nearest neighbors X_j and X_m forms a triangle in hyper space. By definition of inner product,

$$\vec{a} \cdot \vec{b} = |\vec{a}| |\vec{b}| \cos \theta \quad (4.38)$$

where θ is the angle between $\vec{a}$ and $\vec{b}$.

Therefore, the inner product of vector $(\vec{X}_i - \vec{X}_j)$ and vector $(\vec{X}_i - \vec{X}_m)$ contains the information about the distances from point $\vec{X}_i$ to $\vec{X}_j$ and $\vec{X}_m$, and the angle at point X_i inside this triangle. Calculating the local patch centered at each point $\vec{X}_i$ thus keeps information which uniquely characterizes this triangle. It is important that the distance and angle information

with respect to the data points is the inherent information which belongs to the data points distribution, and it is independent to the coordinate system chosen. Therefore, such information are to be kept when the data points are projected from high dimensional space to lower dimensional space.

In an attempt to incorporate a more flexible distance measure while keeping such inherent information associated with the geometry of the triangle, we try to characterize this triangle in another form q_{jm}^i to replace Q_{jm}^i as,

$$q_{jm}^i = (D_{i,j} + D_{i,m} - D_{j,m})/2 \quad (4.39)$$

where $D_{i,j}$ and $D_{i,m}$ denote the distances from point $\vec{X}_i$ to $\vec{X}_j$ and $\vec{X}_m$, respectively, and $D_{j,m}$ denotes the distance from point $\vec{X}_j$ to $\vec{X}_m$.

$$q_{im}^j = (D_{j,i} + D_{j,m} - D_{i,m})/2 \quad (4.40)$$

$$q_{ij}^m = (D_{m,i} + D_{m,j} - D_{i,j})/2 \quad (4.41)$$

The rationale is justified as follows. The q value for all three points forming the triangle will be calculated as in Eq 4.39 to Eq 4.41. With these three equations together, we could get back the value of pairwise distances, $D_{m,i}$, $D_{m,j}$ and $D_{i,j}$. Therefore the triangle side lengths information could be remained by q . From basic geometry knowledge, the unique shape of a triangle could be determined by either angle with length of two adjacent sides, or the length of all three sides. So shape information of the triangle may be captured by the q calculation. Thus in this way, the geometry of the local patches could be remained. Since the value of q actually carries the geometry information of the local patch, the calculated coefficient $W_{i,j}$ may be able to remain and reconstruct this local neighborhood geometry. Substitute Q_{jm} in

Eq 4.37 by q_{jm} in Eq 4.39, we could make the cost function of reconstruction expressed in terms of pairwise distances. The flexible distance measure of pairwise distances D as in Eq 4.33 could enable us to find the fractional coefficient which best describes pairwise distance in high dimensional space, which will be illustrated in the experimental part. The mathematical study is out of the scope of this thesis.

4.7 Experiment and discussion

For description of datasets, preprocessing steps, hardware platforms, classifier settings, and cross validation strategies, please refer to Section 3.4.1 ~ Section 3.4.5.

In our designed experiment, LLE is first applied to the expression data to reduce the dimensionality from several thousands to around 10. Then SVM classification is carried out, and the Leave-one-out classification accuracy is used to evaluate the effectiveness of the feature reduction performance.

47.1 Comparison of LLE+SVM with SVM alone

The performance of LLE is first evaluated by comparison of SVM classification results with and without feature reduction. With LLE feature selection, the classification accuracy is plotted versus the number of features employed in the classification process. For all the three datasets, the classification accuracy achieves the peak with around 2~10 features. Without feature selection, we have input all the features into the SVM classifier, the comparison results are listed in Table 4.1. From all the three datasets tested,

we can see that the performance of classification is not degraded after we reduce the dimensionality of the data from several thousands to less than 10. In addition, with the dimensionality reduced tremendously, we have also cut down the computational cost of SVM much. So generally, the revised LLE has kept the useful information among the features while significantly reducing the dimensionality.

Table 4.1: Comparison of classification accuracy with and without feature selection

Dataset	Feature selection	No. of features	Fraction, K	Accuracy
Leukemia	N.A.	2978	0.5	95%
	LLE	6	0.4	95%
Lymphoma	N.A.	4026	0.5	88%
	LLE	2	0.4	88%
Colon	N.A.	2000	0.5	80%
	LLE	8	0.3	91%

Table 4.2: Time consumed for each run of LLE feature reduction

Dataset	Feature selection	No. of neighbors	Time consumed
Leukemia	LLE	3	less than 2 min
Lymphoma	LLE	3	less than 2 min
Colon	LLE	3	less than 2 min

4.7.2 Comparison of LLE with other feature reduction methods

We have compared the performance of the revised LLE with some classical feature reduction techniques on three microarray datasets discussed in Section 3.4.1. The results of other techniques are extracted from a survey reported by Cho(2003) [27]. The results are listed in Table 4.3. It is observed that our proposed LLE methods are consistently better than the above methods for all three gene expression datasets. Also note, as a feature extraction method, it has achieved correct classification with much less features as required by typical feature selection methods.

4.7.3 Comparison of Fractional metrics with Euclidean distance

We have also compared the performance of our revised LLE using fractional metrics with the original LLE implementation using the Euclidean distance. The classification accuracy are plotted on the same graph in Fig 4.3-4.5 for the three datasets respectively. The improvement in classification accuracy is obvious when k takes value other than 0.5(as for Euclidean).

Table 4.3: Comparison of classification accuracy between LLE and other classical methods using SVM(RBF kernel)

Dataset	Feature selection	No. of features	Accuracy	Fraction, K
Leukemia	LLE	6	95%	0.4
	PCA	25	79%	N.A.
	SNR	25	59%	N.A.
	CC	25	85%	N.A.
Lymphoma	LLE	2	88%	0.4
	PCA	25	60%	N.A.
	SNR	25	76%	N.A.
	CC	25	56%	N.A.
Colon	LLE	8	91%	0.3
	PCA	25	65%	N.A.
	SNR	25	65%	N.A.
	CC	25	65%	N.A.

4.7.4 Discussions

The feature reduction effectiveness can also be viewed graphically. Fig 4.6 - 4.8 shows the scattering of data samples in 2d space by features generated from LLE. Intuitively, the data points form two clusters with each on one side. The misclassified points are just the points on the overlapping borders. For leukemia and colon data, the cluster separation is distinct, while for lymphoma, it is not so good, which is consistent with the classification result in Table 4.3.

This LLE feature extraction process is purely unsupervised, and it does

Figure 4.3: The performance of LLE feature extraction on leukemia dataset

not need the existence of the class labels. Also, the feature extraction calculation is a simple linear algebra problem, and it does not involve any training or iteration process, such that the process can be extremely fast.

Compared to the entropy ranking algorithm we reported in earlier chapter, LLE shows evident advantage in its fast running speed, the LLE projects a micorarray datset in minutes, while the entropy ranking needs at least 1 – 3 hours to generate the feature ranking sequence. However, as a feature extraction technique, the obvious disadvantage is that it loses the direct physical interpretation of the original features after projection.

Figure 4.4: The performance of LLE feature extraction on colon dataset.

Figure 4.5: The performance of LLE feature extraction on lymphoma dataset.

Figure 4.6: 2d scattering of leukemia classes generated by LLE.

Figure 4.7: 2d scattering of colon classes generated by LLE.

Figure 4.8: 2d scattering of lymphoma classes generated by LLE.

Chapter 5

Feature grouping framework – a feature selection approach

In Chapter 2, it was mentioned that there are two kind of problems associated with gene expression data: the grouping of genes of similar function with samples(conditions) as features; and the grouping of samples of similar tumor status with genes as features. Most of the time, research on grouping of samples focuses on the latter problem only. However, recent research has tried to find some linkage between them.

Early researchers have tried to make the work of gene grouping and sample grouping interrelated. Once informative genes have been identified, then it is relatively easy to use conventional clustering algorithms to cluster samples. On the other hand, once samples have been correctly partitioned, some supervised methods such as t-test scores and separation scores [68] can be used to rank the genes according to their relevance to the partition [60].

Xing et al. [61] had proposed a novel interrelated algorithm CLIFF(CLuster via Iterative Feature Filtering) for gene expression data feature selection.

Their work was summarized by Jiang [60] as follows. CLIFF first builds a two-component Gaussian model to rank all the genes according to their discriminability. A set of most discriminant genes are chosen as candidates for further processing. Then CLIFF uses an NCut (normalized cut) graph-theory based algorithm to generate an initial partition and enters an iterative filtering process. For each iteration, a sample partition C and a set of genes are taken as input. During each iteration, a subset of most important genes are selected according to their contribution to the sample partition C based on the 'information gain'. The NCut algorithm then generates a new sample partition C' based on the selected genes. The new sample partition C' and the selected genes are passed as input to the next iteration. The iteration process continues until the newly generated partition C' is identical to the previous partition C . The effectiveness of CLIFF method depends on the result of the NCut algorithm, which is sensitive to noise and outliers in the samples. Therefore the performance of CLIFF is also affected by the presence of noise and outliers contained by the samples [60].

The above mentioned methods can be seen as global methods, in the sense that "the feature space is globally determined and is shared by all resulting clusters, and the resulting clusters are exclusive and exhaustive" [60]. However, from a molecular biological point of view, for any cellular process, there are only a small subset of genes participated, and it only takes place in a subset of the samples. Furthermore, a single gene may participate in multiple pathways that may or may not be coactive under all conditions, so that a gene can participate in multiple clusters or in none at all." [60] Recently a series of subspace clustering methods have been proposed [62] [63] [64] to

capture coherence exhibited by the “blocks” within gene expression matrices.

In this project, a new framework is proposed and tested based on the idea of subspace method. The framework is built on the belief that only a small group of similar genes are involved in the tumor development metabolic process. We use clustering methods to group similar genes into several clusters using samples(conditions) as features [19]. After the feature grouping, we use correlational analysis to pickup one of the feature groups which are believed to be responsible for tumor development as candidate subset for further study. Information compression methods are carried out to pick up representative genes from this group. In actual implementation, we do a ranking of genes, and pick up the top ranked ones as the representatives. The effectiveness of this representation is verified by the sample classification performance by SVM classifier. The framework defines the objectives of each stage of work, and in real implementation, different algorithms may be adopted.

Under this framework, feature selection is carried out in 5 steps,

1. **Data pre-processing** The pre-processing stage aims to truncate extreme values.
2. **Filtering** Genes are filtered by individual feature’s statistics.
3. **Gene clustering** Genes are clustered into subsets by similarity.
4. **Cluster validation** The best gene subset is selected as candidate for further processing.
5. **Cluster reduction** The genes in the candidate subset are ranked, and the best subset are taken from the best ranked genes.

Figure 5.1: Illustration of the framework

5.1 Implement at ion of the framework

The framework is a *feature selection* approach, and it takes advantage of both *filter* and *wrapper* approach. The filtering stage does not need the help of classifier, while in the cluster validation stage it will use the clustering algorithm to choose the best feature subset. In the last stage, feature reduction is a purely *unsupervised* and *filter* approach. Overall, the whole process does not need any training and iteration, so it is simple and fast.

5.1.1 Pre-processing

The pre-processing stage aims to filter out extreme values and select genes which have comparably large variance. By observing the range of data, some extreme values are truncated by their relative low frequency of appearance,

such as

$$\max(g_{i,j}) \leq 16,000 \quad (5.1)$$

$$\min(g_{i,j}) \geq 100 \quad (5.2)$$

is used for leukemia dataset.

5.1.2 Individual filtering

The data are then filtered by the range spread, and relative variance according to following objective:

$$\max(\vec{g}_i) - \min(\vec{g}_i) \geq \alpha \quad (5.3)$$

$$\max(\vec{g}_i)/\min(\vec{g}_i) \geq \beta \quad (5.4)$$

$$\text{variance}(\vec{g}_i)/\text{mean}(\vec{g}_i) \geq \theta \quad (5.5)$$

where $\vec{g}_i$ is the i th gene vector and M is the number of gene vectors, $i \in 1, 2, \dots, M$.

5.1.3 Clustering by genes

The filtered genes are grouped into subsets by clustering tools. The clustering of genes is implemented by KMEANS algorithm using Euclidean distance between gene vectors as similarity measure. The KMEANS algorithm takes in parameter k and tries to divide the n objects into k clusters. It works in the following way:

k initial clusters are randomly chosen as cluster centers;
 each object is assigned to the nearest cluster center;
 the cluster center are then re-calculated;
 each object is re-assigned;
 the process continues, until the sum of the squared error converges.

$$E = \sum_{i=1}^k \sum_{\vec{P} \in C_i} | \vec{P} - \vec{M}_i |^2 \quad (5.6)$$

where $\vec{P}$ is any vector inside cluster C_i , and $\vec{M}_i$ is the center point representing this cluster.

5.1.4 Clusters validation

The candidate cluster is chosen for further refinement according to the following criteria:

1. By clustering performance, each cluster is used as input individually for sample clustering, and the cluster which produces the best accuracy is chosen as candidate.
2. By correlation measure, based on the belief that good feature subset contains features highly correlated with the class, yet uncorrelated with each other [47].

5.1.5 Gene cluster shrinking

Once the candidate feature subset is chosen, we wish to further reduce the size of the feature subset. In the implementation, we have used *PCA ranking* and *entropy ranking*.

PCA ranking We have mentioned in Section 2.2 about Principal Component Analysis as a feature extraction method. However, instead of using the PCs as the new variables, this ranking method uses the information in the PCs to find important variables in the original feature set. The importance of variables are ranked by their contribution to the principal components(PCs). As before, first calculate the PCs, then consider the eigenvector corresponding to the smallest eigenvalue (the least important PC), and discard the variable that has the largest (absolute value) coefficient in that vector. Then, consider the eigenvector corresponding to the second smallest eigenvalue, and discard the variable contributing the largest (absolute value) coefficient to that eigenvector, among the variables not discarded earlier [65].

Entropy ranking The idea of entropy ranking is introduced in Section 2.2. the actual algorithm is carried out as follows:

Feature Ranking Algorithm:

T = Original Variable Set

for $k = 1$ to $(M-1)$

for every variable v in T

$$T_v = T - v$$

Calculate entropy E_{T_v} based on the variable set T_v

end for

Let v_k be the variable that minimize E_{T_v}

$$T = T - v_k$$

end for

After we have ranked the features' effectiveness in classification, we may start with an empty feature set, add in features in descending rank order, and use the classification accuracy of the classifier to verify the necessary number of features to represent the feature subset. The details of the classifier has been introduced in Section 2.4.5.

5.2 Prototyping and discussion

For description of datasets, preprocessing steps, hardware platforms, classifier settings, and cross validation strategies, please refer to Section 3.4.1 ~ Section 3.4.5.

To demonstrate the effectiveness of the framework, we have implemented a prototype of the framework and tested using the Leukemia dataset.

5.2.1 Prototype

Pre-processing The framework is implemented and tested on the leukemia dataset. The original gene expression data contains the expression level of 7129 genes across 72 samples(patient tissues). First we filter out genes which are not active throughout the experiment, i.e. genes which have expression values less than 100 across all the 72 tissues are taken out,

$$\max(\vec{g}_i) > 100 \quad (5.7)$$

Then we filter out the extreme values in the data, by setting a ceiling threshold and a floor threshold,

$$\max(g_{i,j}) \leq 16,000 \quad (5.8)$$

$$\min(g_{i,j}) \geq 100 \quad (5.9)$$

Individual filtering To eliminate the genes which do not have enough variance across samples, we have set three filtering criterias,

$$\max(\vec{g}_i) - \min(\vec{g}_i) \geq 500 \quad (5.10)$$

$$\max(\vec{g}_i)/\min(\vec{g}_i) \geq 5 \quad (5.11)$$

$$\text{variance}(\vec{g}_i)/\text{mean}(\vec{g}_i) \geq TOP2000 \quad (5.12)$$

where $i \in 1, 2, \dots, 7129$. The three criterias are checked consecutively, and for the last criteria, we aim to select the top 2000 features, which have better normalized variance.

Feature grouping After the individual filtering, we use clustering methods to group the genes in several clusters. In this implementation, we have used *KMEANS* to find two gene clusters of 168 and 1832 each. Using the mean

to represent each cluster, we find the former one has better discrimination power. We want to further shrink this cluster in following stage. To verify that the 168 sized gene subset has kept enough classification information, we use PCA to summarize the feature subset in two principal directions; In Fig 5.2, we can see the clear boundary of the two clusters of AML and ALL, which strongly indicates this feature subset has kept the classification information.

Figure 5.2: Data scattering of leukemia

Feature cluster shrinking The 168 feature subset contains still a large number of features, so we wish to further compress this feature subset, and use an even smaller number of features to represent it. We have tried the PCA ranking on this feature subset, the features are ranked according to their contribution to the principal components. To test the effectiveness of

this ranking, we use SVM classifier to classify the data samples according to the ranking order, In Fig 5.3, we can see that with the continuous increment

Figure 5.3: Performance comparison of feature grouping on leukemia

in the number of features, the classification accuracy also increases, until it reaches a maximum performance at around 46 features, beyond that point, the classification accuracy does not improve any further. Up to this point, we say the 68 cluster can be further compressed to a feature subset containing less than 50 features.

To illustrate this top ranked 50 features are the essence of the 168 feature subset, we also plot the classification accuracy of the reverse ranked top 50 features. For ease of comparison, the classification accuracy is plotted on the same graph with the forward ranked accuracy in Fig 5.3.

To illustrate this subset of features is far more effective than the original 2000 feature set, we randomly pickup another 50 features from the original feature set and plot the classification accuracy for comparison. The results are illustrated in Fig 5.4.

Figure 5.4: Performance comparison of feature grouping on leukemia2

Chapter 6

Conclusion and Future work

6.1 Conclusion

In recent years, gene expression data classification has emerged as a hot frontier in data mining research. The ultra high dimensionality of gene expression data has created many practical difficulties for the classification task. Effective data dimensionality reduction tools are in great need. In this project, we have tackled the problem from both feature selection and feature extraction direction.

Entropy ranking Entropy ranking [17] is a purely unsupervised feature selection algorithm. It uses the concept of entropy to describe the degree of separation between the clusters. The entropy is expressed in terms of pairwise distances among data samples, and it will be low for very close or very distant pairwise distance(good class separation) and high for mean distance(poor class separation). The feature selection algorithm iteratively tests different subsets of features based on the entropy of the underlying clusters formed by these subsets of features. The features are ranked accord-

ing to their informativeness in forming distinct clusters. Candidate features are selected from the top ranked ones with the help of sequential floating search. The actual number of features needed is monitored by the classification accuracy of a proper classifier. However, the process to generate the ranking list is computationally prohibitive with the dimensionality of gene expression data. We have tentatively analyzed the most time consuming part of the implementation and given specific suggestions of optimization. The proposed optimization does not reduce the number of iterations, however, it greatly shortened the running time of each iteration. The optimized algorithm is implemented and tested on three microarray datasets. The program running time is reduced to several hours, which is tolerable to typical applications. The experiments also show that entropy ranking as a feature reduction method is effective on microarray datasets.

Revised LLE method Locally linear embedding(LLE) [18] is an unsupervised feature extraction technique. Assuming the data samples reside on a smooth low dimension manifold, using sophisticated linear algebra, LLE tries to summarize the manifold by local geometry. The local geometry of data point neighborhood is represented by a group of reconstruction weights. Then LLE tries to reconstruct the original manifold in lower dimensional space while keeping the same local geometry by careful manipulation of the reconstruction weights. We have analyzed the original algorithms and adopted a more flexible distance measure in the implementation. The revised version of the algorithm is implemented and tested against the original one and also, other classical feature reduction algorithms. Generally, the results are

promising. To the best of our knowledge, we are the first to apply LLE to microarray data analysis.

Feature grouping framework Feature grouping framework is based on the biological hypothesis that among thousands of genes, only a small group of similar genes are relevant for tumor development. Under the framework, the dataset is first preprocessed to remove the extreme values, which are assumed to be caused by experimental error and sample contamination. Then, statistical tools are employed to filter the features by their individual variances. After preprocessing, features are grouped into subsets by clustering tools based on similarity. The best gene subset is selected as the candidate for further shrinking. Information compression techniques, such as PCA are used to summarize the subset by fewer genes. We have successfully implemented a prototype of the framework and achieved good results on Leukemia dataset.

We have reported the application of using entropy ranking and revised LLE algorithms for gene expression data feature reduction. The entropy ranking takes a feature selection approach, while the LLE takes a feature extraction approach. In terms of computational cost, entropy ranking is much higher, it needs hours to reduce a gene expression data set from several thousands dimensions to 30-100 dimensions, while the LLE algorithm reduce the dataset to 3-10 dimensions within minutes. However, as a feature selection method, entropy ranking has the inherited advantage of preservation of interpretation. In terms of classification performance, the LLE performs the same or a bit better on certain data sets with fewer features than entropy ranking, which agrees with our early discussions in the review part that, feature extraction methods normally could achieve better results with fewer

features than selection methods.

6.2 Further research

Upon accomplishing the project, we strongly feel several issues may be addressed for feature reduction in the future,

- In the literature, there has not been much discussions on determination of the necessary number of features. In this project, we have used the classifier performance as a guide to find the necessary number of features that generate best performance. However, it is better if we could decide this from the dataset itself. In feature extraction, the problem is easier, it is referred to as implicit dimension and the normal solution is to set a threshold of how much variance needs to be reserved from the original features. In feature selection, this area may be explored.
- Compared to feature selection, feature extraction has one inherent disadvantage, i.e., features extracted lose the direct physical interpretation of original features. However, was there a way to refer every significant extracted feature to certain individual or group of original features, the problem may be remediable.
- Specific to the framework of feature grouping, one important problem is yet to be solved, i.e., determination of the number of gene groups. The number of targeted gene groups directly influence the feature reduction performance. It is suggested to use hierarchical clustering methods to find a proper level of division of clusters, in other words, the number of clusters, hence to determine the number of gene groups.

Author's publications

■ Shi Chao, Chen Lihui, “High dimensional gene expression data dimension reduction,” in *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems (CIS)*, pp.451-455, Dec 2004.

[2] Shi Chao, Chen Lihui, “Feature dimension reduction for microarray data analysis using Locally Linear Embedding,” in *Proceedings of the 3rd Asia-Pacific Bioinformatics Conference, Series on Advances in Bioinformatics and Computational Biology*, vol. 1, pp.211-217, Imperial College Press, 2005.

Bibliography

- [1] William K. Purves, David Sadava, Gordon H. Orians and H. Craig Heller, *Life: The Science of Biology, Seventh Edition*. Sinauer Associates, Inc, 2003.
- [2] M. Farabee. (2002) On-line biology book. [Online]. Available: <http://www.emc.maricopa.edu/faculty/farabee/BIOBK/BioBookTOC.html>
- [3] Azuaje F, “Interpretation of genome expression patterns: computational challenges and opportunities,” *IEEE Eng. Med. Biol*, vol. 19, no. 6, p. 119, 2000.
- [4] Pomeroy SL, Tamayo P, Gaasenbeek M, Sturla LM, Angelo M, McLaughlin ME, Kim JY, Goumnerova LC, Black PM, Lau C, Allen JC, Zagzag D, Olson JM, Curran T, Wetmore C, Biegel JA, Poggio T, Mukherjee S, Rifkin R, Califano A, Stolovitzky G, Louis DN, Mesirov JP, Lander ES, and Golub TR, “Prediction of central nervous system embryonal tumour outcome based on gene expression,” *Nature*, vol. 415, no. 6870, pp. 436–442, 2002.

- [5] Zajchowski DA, Bartholdi MF, Gong Y, Webster L, Liu HL, Munishkin A, Beauheim C, Harvey S, Ethier SP, and Johnson PH, “Identification of gene expression profiles that predict the aggressive behavior of breast cancer cells,” *Cancer Res*, vol. 61, no. 13, pp. 5168–78, 2001.
- [6] Sorlie T, Perou CM, Tibshirani R, Aas T, Geisler S, Johnsen H, Hastie T, Eisen MB, van de Rijn M, Jeffrey SS, Thorsen T, Quist H, Matese JC, Brown PO, Botstein D, Eystein Lonning P, and Borresen-Dale AL, “[Geneexpression patterns of breast carcinomas distinguish tumor sub-classes with clinical implications,” *Proc Natl Acad Sci*, vol. 98, no. 19, pp. 10869–10874, 2001.
- [7] W. Dubitzky, M. Granzow and D. Berrar, *Comparing Symbolic and Sub-symbolic Machine Learning Approaches to Classification of Cancer and Gene Identification*. Dordrecht: Kluwer Academic, 2001.
- [8] L J Van’t Veer and D De Jong , “The microarray way to tailored cancer treatment,” *Nut. Med*, vol. 8, no. 1, pp. 13–14, 2002.
- [9] DeRisi J, Penland L, Brown PO, Bittner ML, Meltzer PS, Ray M, Chen Y, Su YA, and Trent JM, “Use of a cDNA microarray to analyse gene expression patterns in human cancer,” *Nat. Genet*, vol. 14, no. 4, pp. 457–460, 1996.
- [10] Y. Lu and J. Han, “Cancer classification using gene expression data,” *Information Systems*, vol. 28, no. 4, pp. 243–268, 2003.

- [11] Michael F. Ochs and Andrew K. Godwin, "Microarrays in cancer: Research and application~*Microarrays and Cancer: Research and Applications, A Supplement to BioTechniques*, vol. 34, pp. 4–15, 2003.
- [12] Donna K. Slonim, Pablo Tamayo, Jill P. Mesirov, Todd R. Golub, and Eric S. Lander, "Class prediction and discovery using gene expression data," *Proceedings of the fourth annual international conference on Computational molecular biology*, vol. 4, pp. 263–272, 2000.
- [13] S. Lakhani, "Microarray and histopathological analysis of tumours the future the past," *Nat. Rev. Can.*, vol. 1, no. 2, pp. 151–157, 2001.
- [14] D. V. Nguyen and D. M. Rocke, *Classification of Acute Leukemia based on DNA Microarray Gene Expressions using Partial Least Squares*. Dordrecht: Kluwer Academic, 2001.
- [15] Anton Berns, "Cancer: gene expression in diagnosis," *Nature*, vol. 403, pp. 491–492, 2000.
- [16] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *Journal of Computational Biology*, vol. 6, no. 3/4, pp. 281–297, 1999.
- [17] M. Dash and H. Liu, "Feature selection for clustering," *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications*, pp. 110–121, 2000.
- [18] S.T.Roweis and L.K.Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, 2000.

- [19] S. Dudoit, J. Fridlyand, and T. P. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data," *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 77–87, 2002.
- [20] Yang YH, Dudoit S, Luu P, Lin DM, Peng V, Ngai J, and Speed TP, "Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation," *Nucleic Acids Research*, vol. 30, no. 4, 2002.
- [21] D. A. Lashkari, J. L. DeRisi, J. H. McCusker, A. F. Namath, C. Gentile, S. Y. Hwang, P. O. Brown, and R. W. Davis, "Yeast microarrays for genome wide parallel genetic and gene expression analysis," *Proc. of the Natl. Acad. of Sci*, vol. 94, no. 24, pp. 13057–13062, 1997.
- [22] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. of the Natl. Acad. of Sci*, vol. 95, no. 25, pp. 14863–14868, 1998.
- [23] DeRisi JL, Iyer VR, and Brown PO, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, vol. 278, no. 5338, pp. 680–686, 1997.
- [24] Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, and Lander ES, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

- [25] Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X, Powell JJ, Yang L, Marti GE, Moore T, Hudson J Jr, Lu L, Lewis DB, Tibshirani R, Sherlock G, Chan WC, Greiner TC, Weisenburger DD, Armitage JO, Warnke R, Levy R, Wilson W, Grever MR, Byrd JC, Botstein D, Brown PO, and Staudt LM, "Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling," *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.
- [26] Ross DT, Scherf U, Eisen MB, Perou CM, Rees C, Spellman P, Iyer V, Jeffrey SS, Van de Rijn M, Waltham M, Pergamenschikov A, Lee JC, Lashkari D, Shalon D, Myers TG, Weinstein JN, Botstein D and Brown PO, "Systematic variation in gene expression patterns in human cancer cell lines," *Nature Genetics*, vol. 24, no. 3, pp. 227–234, 2000.
- [27] Sung Bae Cho and Hong Hee Won, "Machine learning in DNA microarray analysis for cancer classification," *Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics*, vol. 19, pp. 189–198, 2003.
- [28] Benny Y. M. Fung and Vincent T. Y. Ng, "Classification of heterogeneous gene expression data," *ACM SIGKDD Explorations Newsletter*, vol. 5, no. 2, pp. 69–78, 2003.
- [29] D. Koller and M. Sahami, "Towards optimal feature selection in machine learning," *Proceedings of 13th International Conference*, pp. 284–292, 1996.

- [30] W. Siedlecki and J. Sklansky, "On automatic feature selection," *Int. J. Pattern Recognition Artif. Intell.*, vol. 2, pp. 197–220, 1988.
- [31] E. P. Xing, M. I. Jordan, and R. M. Karp, "Feature selection for high-dimensional genomic microarray data," *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 601–608, 2001.
- [32] C. Campbell, Y. Li, and M. Tipping, "An efficient feature selection algorithm for classification of gene expression data," *NIPS 2001 Workshop on Machine Learning Techniques for Bioinformatics*, 2001.
- [33] F. Ferri, P. Pudil, M. Hatef, and J. Kittler, "Comparative study of techniques for large-scale feature selection," *Pattern Recognition in Practice IV, Multiple Paradigms, Comparative Studies and Hybrid Syst.*, pp. 403–413, 1994.
- [34] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 2, pp. 153–158, 1997.
- [35] J. Mao and A. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Trans. Neural Networks*, vol. 6, pp. 296–317, 1995.
- [36] S. K. Alain Biem and B.-H. Juang, "Pattern recognition using discriminative feature extraction," *IEEE Trans. Signal Processing*, vol. 45, no. 2, pp. 500–504, 1997.

- [37] C. Lee and D. Landgrebe, “Decision boundary feature extraction for neural networks,” *IEEE Trans. Neural Networks*, vol. 8, no. 1, pp. 75–83, 1997.
- [38] N. Intrator, “Feature extraction using unsupervised neural network,” *Neural Computation*, vol. 4, pp. 98–107, 1992.
- [39] K. Y. Yeung and W. L. Ruzzo, “An empirical study on pca for clustering gene expression data,” *Technical Report UW-CSE-00-11-03, department of computer science and engineering, University of Washington*, 2001.
- [40] —, “Principal component analysis for clustering gene expression data,” *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.
- [41] L. J. Cao, K. S. Chua, W. K. Chong, H. P. Lee, and Q. M. Gu, “A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine,” *Neurocomputing*, vol. 55, pp. 321–336, 2003.
- [42] Rosipal R., Girolami M., Trejo L.J., and Cichocki A., “Kernel PCA for feature extraction and de-noising in non-linear regression,” *Neural Computing & Applications*, vol. 10, no. 3, pp. 231–243, 2001.
- [43] H. Liu and H. motoda, *Feature extraction construction and selection: A data mining perspective*. United States of America: Kluwer academic publishers, 1998.
- [44] Andrew D Keller, Michel Schummer, Walter L Ruzzo, and Lee Hood, “Bayesian classification of DNA array expression data,” *Technical Report UW-CSE-2000-08-01, August*, 2000.

- [45] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant features and the subset selection problem,” *International Conference on Machine Learning*, pp. 121–129, 1994.
- [46] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.
- [47] M. A. Hall, “Correlation-based feature selection for discrete and numeric class machine learning,” *Proc. 17th International Conf. on Machine Learning*, pp. 359–366, 2000.
- [48] R. A. Johnson and G. K. Bhattacharyya, *Statistics: principles and methods*. United States of America: John Wiley and Sons, Inc, 1996.
- [49] Bernhard Scholkopf, Alexander Smola and Klaus-Robert Muller, “Non-linear component analysis as a kernel eigenvalue problem,” *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.
- [50] I. Jolliffe, *Principal Component Analysis*. Springer Verlag, 1986.
- [51] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.
- [52] M. Dash, H. Liu, and J. Yao, “Dimensionality reduction for unsupervised data,” *In Ninth IEEE International Conference on Tools with AI, ICTAI*, vol. 19, 1997.
- [53] Shannon, “C.e. a mathematical theory of communication,” *The Bell Systems Technical Journal*, vol. 27, pp. 379–423, 1948.

- [54] U. Alon, N. Barkai, D. A. Notterman, K. Gish, D. M. S. Ybarra, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon cancer tissues probed by oligonucleotide arrays," *Proc Natl Acad Sci*, vol. 96, no. 12, pp. 6745–6750, 1999.
- [55] T. Joachims. (2002) Svm light support vector machine. [Online]. Available: <http://svmlight.joachims.org/>
- [56] A. Schwaighofer. Matlab interface to svm light. [Online]. Available: <http://www.cis.tugraz.at/igi/aschwaig/software.html>
- [57] D. Gering, "Linear and nonlinear data dimensionality reduction," *Area Exam Report, MIT*, 2002.
- [58] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim, "On the surprising behavior of distance metrics in high dimensional spaces," *ICDT 01: Proceedings of the 8th International Conference on Database Theory*, vol. 1973, no. 420-434, 2001.
- [59] L. Saul and S. Roweis, "An introduction to locally linear embedding," <http://www.cs.toronto.edu/roweis/lle/>.
- [60] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, 2004.
- [61] E. P. Xing and R. M. Karp, "CLIFF: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts,"

Proceedings of The Ninth International Conference on Intelligence Systems for Molecular Biology, ISMB, 2001.

- [62] G. Getz, E. Levine, and E. Domany, “Coupled two-way clustering analysis of gene microarray data,” *Proc. Natl. Acad. Sci USA*, vol. 97, no. 22, pp. 12079–12084, 2000.
- [63] Y. Cheng and G. M. Church, “Biclustering of expression data,” *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, no. 8, pp. 93–103, 2000.
- [64] Lazzeroni L and Owen A, “Plaid models for gene expression data,” *Statistica Sinica*, vol. 12, no. 1, pp. 61–86, 2002.
- [65] Fodor I. K, “A survey of dimension reduction techniques,” *LLNL technical report UCRL-ID-148494*, 2002.