The Pennsylvania State University

The Graduate School

College of Engineering

AN ANALYTICAL FRAMEWORK FOR THEORETICAL ANALYSES IN BINARY CLASSIFIER ENSEMBLES AND A STUDY OF ISSUES IN CLUSTER VALIDATION FOR GENOMIC DATA

A Thesis in

Computer Science and Engineering

by

Anand M. Narasimhamurthy

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2006

The thesis of Anand M. Narasimhamurthy was reviewed and approved* by the following.

Raj Acharya
Professor of Computer Science and Engineering
Thesis Adviser
Chair of Committee
Head of the Department of Computer Science and Engineering

Piotr Berman
Associate Professor of Computer Science and Engineering

Rajeev Sharma
Associate Professor of Computer Science and Engineering

Jia Li
Assistant Professor of Statistics

Rangachar Kasturi
Professor of Computer Science and Engineering
Douglas W. Hood Professor and Chairman
Department of Computer Science and Engineering,
University of South Florida,Tampa
Special Member

*Signatures are on file in the Graduate School.

Abstract

Classification and clustering subsume a large number of pattern recognition tasks. The contribution of this work is two-fold. The first part relates to classification, more specifically, to classifier ensembles (multiple classifier systems) for binary classification (two-class) problems. In the second part of this work, we explore some of the issues in cluster validation as relates to genomic data.

Classifier ensembles have proved to be promising and useful in various applications. The basic idea is to build a team of classifiers and combine their outputs in order to obtain a more ”robust” classification, as opposed to relying on the output of a single classifier. The outputs of the classifiers could be combined in a number of ways. Majority voting is a simple yet useful combination scheme. Our contribution in the area of multiple classifier systems includes the formulation of the problem of computing the upper and lower bounds of majority voting accuracy for an ensemble of binary classifiers as a linear program (LP). The resulting analytical framework can be used for performing a variety of analyses related to voting. Diversity and complementarity are considered as desirable properties in an ensemble of classifiers, however there is no widely accepted characterization of these concepts, thus making an objective evaluation difficult. Many of the measures defined in the literature are formulated in terms of correct/incorrect classifications, these are referred to as error-diversity measures. We show that the analytical framework mentioned above can be used effectively to evaluate error-diversity measures and explore whether there is a useful relationship between the selected diversity measures and the ensemble accuracy.

Next, we explore some of the issues in cluster validation in the context of microarray data. Clustering is often an important first step in the analysis of genomic data, cluster validation is an important step in cluster analysis. We assess the suitability of standard cluster validation techniques for microarray data. Often an important goal in clustering genomic data is to group genes based on underlying biologically relevant criteria such as functions. It is thus of interest to compare a clustering result with an external clustering, for instance comparing the grouping of genes obtained by applying a clustering algorithm on microarray data against a reference grouping such as a grouping based on biological functions derived from existing biological literature. We examine some of the recent cluster validity measures proposed in the literature which may be suitable for this purpose. We propose a measure for the distance between two membership matrices and suggest when this could be a suitable choice for the above purpose. We use standard network flow algorithms for finding the measure. We also formulate related theoretical problems as network flow problems.

Table of Contents

List of Tables

List of Figures

Acknowledgments

A number of people desrve to be acknowledged for this work. First of all, I would like to acknowledge my parents for all their support during this long journey. A number of friends, colleagues and officemates at Penn State, too many to list individually, made my rather long stay at Penn State pleasant.

I would like to thank my adviser Prof. Raj Acharya and all my committee members, namely Dr. Berman, Dr. Jia Li, Dr. Rajeev Sharma and Prof Kasturi for all the help they provided. Prof. Raj Acharya used to make himself as accessible as possible to his students inspite of his busy schedule and numerous administrative commitments as the department head. Prof Rangachar Kasturi, my former adviser and Dr. Rajeev Sharma both members of my committee, deserve special mention. Prof Kasturi was instrumental in me joining the graduate program at the CSE department at Penn State. He provided a lot of useful advice throughout the duration he was at Penn State as well as later. As long as I was under their supervision, Prof Kasturi and Dr. Sharma ensured that my funding was taken care of. This enabled me to focus on my research.

I would also like to thank Dr. Jia Li and Dr. Francesca Chiaromonte of the Statistics department. Dr. Jia Li provided a lot of technical help both before and after she was a committee member. Also, sitting in her data mining class helped me quite a bit. Sitting in the Bioinformatics II course taught by Francesca helped me gain background knowledge and focus my thesis research.

Special thanks to the entire staff of the CSE department who were always accessible and willing to help, most notably Vicki Keller, Beth Kennedy and Karen Corl. All of them were

of immense help during the entire duration of my graduate studies at Penn State. Whether it was help with finding office supplies or administrative stuff related to courses I taught, Karen was available when needed. Vicki ensured that all the formalities related to the department or the International Students and Scholars Office were promptly taken care of. As for Beth, I have lost count of the number of times I asked her to check up Dr. Acharya's schedule and make an appointment if necessary. Needless to say she would patiently oblige everytime.

This acknowledgement section would be incomplete without thanking the janitorial staff who worked the evening shift in the IST building. I could count on them to be around at odd hours for the occasional chat, this was especially helpful during frustrating periods of my Ph.D.

Chapter 1

Introduction

We motivate the discussion for the rest of the thesis in this chapter. The contribution of this work is two-fold. The first part relates to classification, more specifically, to classifier ensembles (multiple classifier systems) for binary classification (two-class) problems. In the second part of this work, we highlight some of the issues in cluster validation for genomic data. Each chapter in this thesis is self contained, each individual chapter has a summary and a conclusions section.

This chapter is organized as follows. Section 1.1 provides an overview of classifier ensembles and sets the scene for chapters 2 and 3. In section 1.2 we briefly discuss cluster validation and motivate the discussion for Chapters 4 and 5. The chapterwise contributions are outlined in section 1.3.

1.1 Background : Multiple classifier systems

Classifier ensembles have proven to be useful in various applications. The basic idea is to build a team of classifiers and combine their outputs in order to obtain a more ”robust” classification, as opposed to relying on the output of a single classifier. In this work, we focus on the binary classification (two class) problem. Although in the general case there are multiple classes, two-class problems nevertheless constitute an important special case. Many real-life

classification applications fall into this category e.g. fraud detection from credit card transaction logs (fraud or genuine), medical diagnosis of tumors (tumor or no tumor).

Xu *et al.* [60] categorize the output information that classification algorithms supply (or are able to supply) as :

- Abstract level : The classifier only outputs a unique label for each sample indicating the class to which it belongs to.
- Rank level : The classifier outputs a ranked list of labels, the label with the best rank is the top choice of the classifier for the class the sample belongs to.
- Measurement level : The output is a set of measurement values one for each label, signifying the degree that the given sample belongs to that particular class.

Accordingly, the outputs of the classifiers could be combined in a number of ways. Xu *et al.* [60] categorize combination methods as Type 1, Type 2 or Type 3 according to whether the combination is made based on information at the abstract, rank or measurement level respectively. In the case of Type 3 combination all classifiers are required to provide measurement level outputs, and if they are measurement vectors of different kinds, they need to be transformed into the same kind. Depending on the kind of measurement values output by the classifiers, a number of Type 3 combination methods are possible. For instance, when all classifiers are Bayes classifiers, averaged Bayes classifier and its variants are possible combination schemes.

Type 1 is the most general since the classifiers could be entirely different in their methodologies, in fact an abstract level output can be obtained from either rank level or measurement

level output. Type 1 combination is sometimes also referred to as decision level fusion. Examples of Type 1 combination schemes include the majority and pluraity voting, Behaviour Knowledge Space (BKS) [29] and so on. Majority voting is one of the popular schemes employed in classifier combination. A significant amount of literature comprising theoretical and experimental analyses, has addressed majority voting ([32, 39, 38, 37, 35]) in particular and voting schemes in general. In our work we formulate the problem of determining the upper and lower bounds of majority voting performance for a binary classification problem, as an optimization problem with linear constraints. The resulting analytical framework may be used for performing a number of analyses. This is discussed in Chapter 2.

A very important consideration in designing classifier ensembles is to obtain a high classification accuracy. Diversity in the classifier ensemble is considered a desirable attribute towards this end. Intuitively if all the classifiers commit the same errors, there is no benefit gained from the ensemble. On the other hand if different classifiers commit errors on different samples, the outputs could be potentially combined in order to obtain a classification accuracy better than the best single classifier in the ensemble.

One possible way of characterizing diversity is in terms of errors in classification by the different classifiers in the ensemble, on a set of samples. A number of diversity measures proposed in different contexts can be adapted as error diversity measures in classifier ensembles. Although this seems simple, characterization of diversity in classifier ensembles even for the binary classification case is not straightforward and hence an objective evaluation of relationship between diversity and ensemble accuracy is tricky. Also previous studies have highlighted the ambiguous relationship between these diversity measures and ensemble accuracy and have questioned the utility of error diversity measures in building an ensemble [54, 38]. We show

that the same analytical framework for majority voting may be used for analyses related to error diversity measures. We use the framework mentioned above to explore whether there is a useful relationship between various error diversity measures and the accuracy of the ensemble. This is discussed in Chapter 3.

1.2 Background : Issues in cluster validation for genomic data

The second part of this thesis deals with cluster validation issues in the context of gene expression data. For the discussion in the succeeding chapters, we use the term clustering to refer both to the process as well as the result. We use the term *hard clustering* to mean a clustering in which an object (data point) is assigned to one cluster only. In soft clustering, the degree to which an object is associated with a cluster is indicated by a membership grade.

The advent of technologies such as microarrays has enabled the generation of expression patterns of thousands of genes in a single experiment. Clustering usually constitutes the initial step in the analysis of gene expression data, the aim being the identification of groups of genes for further analysis. The hope is that similarity with respect to expression is often indicative of similarity with respect to more fundamental traits such as function.

Cluster validation constitutes an important step in cluster analysis. Loosely speaking, the aim of cluster validation is to assess the quality of a given clustering. This could then be used to determine the ”correct” number of clusters. Cluster validation procedures may be broadly categorized as internal and external validation. Internal validation procedures look at the data alone and typically employ criteria such as inter and intra cluster distances. External validation on the other hand involves comparing the given clustering with an external clustering.

We could look at the overall cluster validation process involving genomic data as comprised of two stages. In the first stage, the ”true” number of clusters in the data is estimated and thus the ”optimal clustering” which reflects the structure in the data as best as possible is determined. This could potentially be accomplished by the use of cluster validity indices (predominantly internal CVIs). Different distance measures and clustering algorithms should be tried out in order to see which of them can find the ”natural clusters” in the data. Chapter 4 mainly deals with the ”first stage”. We outline many of the practical issues encountered in cluster validation. We discuss some of the representative cluster validity indices proposed in the literature and examine their usefulness and shortcomings.

In the second stage the optimal clustering thus determined is compared against an appropriately chosen reference clustering. Consider for instance the application of a clustering algorithm on gene expression data. It may be of interest to compare the groups of genes obtained as a result of the clustering with a grouping of the genes derived from the biological literature in order to assess whether in the given dataset similarity in expression reflects a biologically relevant similarity such as co-regulation or similarity in function. Chapter 5 mainly focusses on this problem from a cluster validation perspective.

Many clustering methods used for analysis of genomic data employ hard clustering i.e. a gene is assigned to one cluster only. A hard clustering of genes could be very restrictive for the following reasons. Genes are often coordinated by multiple regulatory mechanisms and could participate in different functions. Genes were frequently found to be correlated with multiple classes [13, 12]. Hence it may be unrealistic to assign genes to one cluster only. In such a scenario hard partitioning schemes may impose arbitrary boundaries and may not capture the information from a functional point of view. Also gene expression data obtained from microarrays,

is often noisy, thus calling into question any hard assignments. More sophisticated approaches (eg [53]) allow genes to be assigned to multiple clusters. In the most general case, both the clustering obtained by application of a clustering algorithm and the reference clustering could be soft i.e. the degree to which an object is associated with a cluster is indicated by a membership grade which can have a value between 0 and 1. It is reasonable to expect that a general cluster validation methodology must incorporate methods able to deal with the problem of overlapping and/or non-exhaustive clusters. We discuss the Mallows based distance proposed in the recent literature [62] and suggest that this could be more suitable for cluster validation in the context of genomic data, especially if the problem involves comparing sets of overlapping clusters. We also propose a minimum cost matching distance suitable for the same purpose.

Network flow problems are a powerful algorithmic and conceptual tool. The computation of both the Mallows based distance and the minimum cost matching distances can be formulated as flow problems. A related seemingly combinatorial problem that can also be formulated as a flow problem is the following. Although the choice of the reference clustering depends on the task at hand, typically groupings of genes such as functional groupings or those based on gene ontology for instance, are comprised of overlapping clusters. However, a number of clustering methods group data points into mutually exclusive partitions. Thus the theoretical minimum distance between the two clusterings cannot be zero under any distance measure. Under the minimum cost matching distance measure for a fixed reference clustering, we also compute the theoretical minimum distance between the reference clustering and a clustering of a fixed size. The minimum distance could be used as a baseline for evaluating the performance of the clustering algorithm producing mutually exclusive partitions. These aspects are discussed in

detail in Chapter 5. We suggest that the flow problem formulation could be a useful framework for cluster validation problems.

1.3 Summary of contributions

The different contributions of this work may be summarized as follows, along with the relevant publications.

1. Multiple classifier systems

   (a) **Computing the upper and lower bounds of majority voting performance for binary classifier ensembles** (Chapter 2)

   Related publications [43, 45]

   The problem of determining the theoretical upper and lower bounds of majority voting accuracy for an ensemble of binary classifiers can be formulated as a Linear Program (LP) [43, 45]. The resulting framework can be used for performing various theoretical analyses.

   (b) **Analysis of error-diversity measures for classifier ensembles** (Chapter 3)

   Related publication : [44]

   The LP framework mentioned above may be used to analyze the role of “diversity“ in binary classifier ensembles. In this work we focus on the so called error-diversity measures i.e. those that are defined in terms of correct/incorrect classification combinations. We provide a qualitative and a quantitative analysis of error-diversity measures. In particular, we explore whether there is a useful relationship between

the diversity of a classifier ensemble (as defined by a suitable diversity measure) and the accuracy of the ensemble.

2. Cluster validation for genomic data

(a) **Utility analysis of Cluster Validity Indices:** (Chapter 4)

Clustering genes based on expression data often constitutes a first step in the analysis of genomic data since co-expression may be indicative of co-regulation and other biologically relevant criteria such as functions. Cluster validation is often a necessary step in the process of clustering. A number of cluster validity indices (CVIs), both internal (those that look at the data only) and external (those that involve comparing a clustering result with an external clustering) have been proposed, each has its own advantages and limitations. We explore some of the issues in cluster validation in the context of microarray data. Many internal CVIs implicitly assume that the data resides in a Euclidean space. While this may be reasonable for gene expression data, this could be problematic for other types of data such as sequence data. Chapter 4 focusses on internal CVIs. We examine the suitability and limitations of some of the more well known cluster validity indices (CVIs) for genomic data, and suggest which CVIs may be applicable for heterogeneous data types and non-Euclidean spaces.

(b) **External cluster validity indices for genomic data and formulating theoretical problems in cluster validation as flow problems :** (Chapter 5)

Chapter 5 focusses on external cluster validity indices. It is often useful to compare the grouping of genes generated by a clustering algorithm against a reference grouping (a “gold standard”). Some of the standard external CVIs such as the Rand

Index [48] may be used, however many of them have limitations which make them not entirely suitable for genomic data. We examine external validity methods from the recent literature which overcome some of the limitations of previous methods and may hence be more suitable for genomic data. We also propose a measure for the distance between two membership matrices and suggest when this could be a suitable choice for the above purpose. We show the relationship between this measure and a Mallows distance metric proposed by Zhou *et al.* [62]. We use standard network flow algorithms [2] for computing the measure. We also formulate related theoretical problems as appropriate network flow problems.

Chapter 2

Theoretical bounds of majority voting accuracy for a binary classifier ensemble

Summary

In this chapter we formulate the problem of determining the theoretical bounds of majority voting performance as an optimization problem with linear constraints. A number of earlier studies that have attempted a theoretical analysis of majority voting assume independence of the classifiers. No assumptions on the independence of classifiers are made in our formulation. For a binary classification problem, given the accuracies of the classifiers in the team, the theoretical upper and lower bounds for performance obtained by combining them through majority voting are shown to be solutions of the corresponding optimization problem. The objective function of the optimization problem is non-linear in the case of an even number of classifiers when rejection is allowed, for the other cases the objective function is linear and hence the problem is a linear program (LP). The framework developed enables a variety of analyses and investigations.

Related publications : [43, 45]

2.1 Introduction

Majority voting is a very popular combination scheme both because of its simplicity and its performance on real data. The performance of majority voting has been demonstrated experimentally in a number of studies such as handwriting recognition ([60, 27]), person authentication

[32] and so on. A simple analytical justification for majority voting may be given by the well known Condorcet's theorem [5]. Under the assumption of independent classifiers, if the individual classifier error rate $e < 0.5$ (assume for simplicity that all classifiers have the same error rate), for odd number of classifiers(voters) N, the correct decision rate increases with increasing N. A number of studies have addressed the problem of a theoretical analysis of majority voting ([32, 39, 38, 37, 35]). In [39] Lam and Suen provide an analysis of majority voting under the assumption that the classifiers are independent. In [32] Kittler et al. develop a theoretical framework for combining classifiers which use distinct pattern representations. Majority voting is shown to be a special case of the sum rule. The sum rule is developed under the assumptions of statistical independence and that the a posteriori probabilities computed by the respective classifiers do not deviate significantly from the prior probabilities. They show that the sum rule is more resilient to estimation errors compared to the other combination strategies discussed in their paper. In a more recent work [35] Kuncheva et al. attempt to address the performance of majority voting empirically. They also provide some insights based on pairwise dependence statistics, since majority vote with dependent classifiers can offer improvement over independent classifiers and over the individual accuracies.

In this work we formulate the problem of determining the theoretical bounds of majority voting performance for a binary classification problem for both odd and even number of classifiers and also the case where rejection is allowed. The optimization problem is non-linear when we allow rejection i.e. we allow a sample to be not classified in case of a tie (this possibility arises only for an even number of classifiers). We also use the framework developed, for analyzing the relationship between two candidate measures of diversity proposed in the literature and majority vote accuracy.

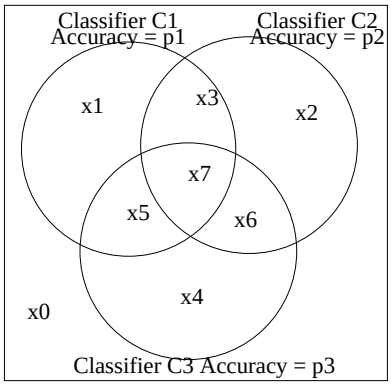
This chapter is organised as follows. In section 2.2 the problem of determining the theoretical upper and lower bounds of majority voting accuracy is formulated as an optimization problem. The results are shown in Section 2.3. The results pertaining to theoretical bounds are discussed in Section 2.3.1. Discussion related to the improvement of classification accuracy is presented in Section 2.3.2. Conclusions are presented in section 2.4.

2.2 Problem formulation

In this section we derive the theoretical upper and lower bounds of majority voting performance for a two class problem, given a set of classifiers whose accuracies are known. Specifically we address the following problem : *"For a binary classification problem, given a set of K classifiers with known accuracies $p_1, \ldots, p_K$ respectively, what are the theoretical upper and lower bounds for the majority voting scheme which combines the outputs of the individual classifiers."*

2.2.1 Notation and representation

We use the following notation for the rest of the chapter. Let $bit(i, K)$ denote the K bit binary expansion of i. Each classifier is represented by a bit (1 or 0) with 1 indicating that the classifier is correct and 0 indicating that it is incorrect. We follow the convention that if there are K classifiers $C1, C2, \ldots, CK$, $C1$ is the LSB (least significant bit) and CK is the MSB (most significant bit) . Let x $= [x_0, x_1, \ldots, x_{(2^K-1)}]^T$ be the vector of joint probabilities (since there are 2^K possible combinations of correct/incorrect classifications for K classifiers). The joint probabilities (dependencies) of the classifiers may be shown by Venn diagrams as in Table 2.1 which illustrates the joint statistics for 3 classifiers.



(A)

Regions correspond to bit combinations.
x_i is the probability associated with region i
i.e. the bit combination $bit(i, K)$.

Example:
For K=3 classifiers, (shown left)
x=$[x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7]^T$
Region 5 corresponds to the bit combination 101
i.e. it corresponds to (C3 correct,C2 wrong,C1 correct)
and x_5 = Prob(C3 correct,C2 incorrect,C1 correct).

(B)

Table 2.1. Diagram illustrating joint statistics of 3 classifiers. (A) Venn Diagram (B) Illustration

2.2.2 The optimization problem

The optimization problem may be derived as follows. Let $g(\mathrm{x})$ represent the probability of correct classification of majority vote. This is the objective function to be maximized/minimized subject to certain constraints. We note that the sum of joint probabilities where classifier r is correct must equal p_r (marginalization of joint probabilities over other classifiers). This can be represented as K constraints of the form $\boldsymbol{b_r^T}$ x = p_r $(1 \leq r \leq K)$ or as a single matrix equation (2.2a) where A_{eq} is a $K \times 2^K$ matrix whose rows $\boldsymbol{b_1^T}, \boldsymbol{b_2^T}, \ldots, \boldsymbol{b_K^T}$ correspond to the classifiers $C1, \ldots, CK$ (accuracies $p_1, \ldots, p_K$ respectively). Additionally, $\sum_{i=1}^{2^K-1} x_i = 1$ and $0 \leq x_i \leq 1$ where $0 \leq i \leq 2^K - 1$. The **optimization problem** for the general case is given by equations (2.1) - (3.5). The **objective function** $g(\mathrm{x})$ given by eqn. (2.8) is linear in x for cases 1) and 2) and non-linear for case 3); the **constraints** (eqns. 2.2a-2.2c) are linear and same for all 3 cases below.

Case 1) Odd number of classifiers (K).

Case 2) Even number of classifiers, no rejection permitted. (Assuming equal priors for the two classes a random decision is made in case of a tie.)

Case 3) Even number of classifiers, rejection permitted in case of a tie : We assume that we are interested only in the accuracy among the number of samples not rejected (= $\boldsymbol{f}^{\boldsymbol{T}}_{\boldsymbol{majority}}$ x + $\boldsymbol{f}^{\boldsymbol{T}}_{\boldsymbol{minority}}$ x)

The optimization problem : max /min $g(\mathrm{x})$ (2.1)

$$\text{s.t.} \quad A_{eq}\mathrm{x} = \mathrm{d} \tag{2.2a}$$

$$0 \leq x_i \leq 1 \;\; (0 \leq i \leq 2^K - 1) \tag{2.2b}$$

$$\vec{1}^T \mathrm{x} \;=\; 1 \tag{2.2c}$$

where,

$$\mathrm{d} = [p_1, p_2, \ldots p_K]^T \quad \text{(vector of classifier accuracies.)} \tag{2.3a}$$

$$\mathrm{x} = [x_0, x_1, x_2, \ldots, x_{(2^K-1)}]^T \quad \text{(vector of joint probabilities.)} \tag{2.3b}$$

and $A_{eq} = [\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_K}]^T$ ($K \times 2^K$ matrix, row r corresponds to r^{th} classifier) (2.3c)

$\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_K}$ are as in equation (3.6) and $g(\mathrm{x})$ is as in equation (2.8). The vectors $\boldsymbol{f_{majority}}, \boldsymbol{b_1}, \ldots \boldsymbol{b_K}$ are infact bit strings of length 2^K. It can be easily seen that,

$$\begin{aligned} \boldsymbol{b_1} &= [0\ 1, \ldots, 0\ 1]^T \\ \boldsymbol{b_2} &= [0\ 0\ 1\ 1, \ldots, 0\ 0\ 1\ 1]^T \\ &\vdots \\ \boldsymbol{b_K} &= [\overbrace{0\ 0 \ldots 0\ 0}^{2^{(K-1)}}, \overbrace{1\ 1 \ldots 1\ 1}^{2^{(K-1)}}]^T \end{aligned} \tag{2.4}$$

Let bit(i, K) denote the K bit binary representation of integer i and let $\boldsymbol{f_{majority}}$(i) denote the entry at ith position in $\boldsymbol{f_{majority}}$ ($0 \leq i \leq 2^K - 1$). We define 3 vectors as in equations (2.5) -(2.7).

$$\boldsymbol{f_{majority}}(i) = \begin{cases} 1 & \text{if no. of 1s in } bit(i, K) > K/2 \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

$$\boldsymbol{f_{minority}}(i) = \begin{cases} 1 & \text{if no. of 1s in } bit(i, K) < K/2 \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

$$\boldsymbol{f_{tie}}(i) = \begin{cases} 1 & \text{if no. of 1s in } bit(i, K) = K/2 \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

$$g(\mathrm{x}) = \begin{cases} \boldsymbol{f}^{\boldsymbol{T}}_{\boldsymbol{majority}}\,\mathrm{x} & \text{odd K. (Case 1)} \\ \boldsymbol{f}^{\boldsymbol{T}}_{\boldsymbol{majority}}\,\mathrm{x} + 0.5\boldsymbol{f}^{\boldsymbol{T}}_{\boldsymbol{tie}}\,\mathrm{x} & \text{even K, no rejects, equal priors (Case 2)} \\ \dfrac{\boldsymbol{f}^{\boldsymbol{T}}_{\boldsymbol{majority}}\,\mathrm{x}}{(\boldsymbol{f}^{\boldsymbol{T}}_{\boldsymbol{majority}}\,\mathrm{x}+\boldsymbol{f}^{\boldsymbol{T}}_{\boldsymbol{minority}}\,\mathrm{x})} & \text{for even K with rejection (in case of a tie) (Case 3)} \end{cases} \tag{2.8}$$

From Table 2.1 it can be verified that for K = 3 classifiers, $g(\mathrm{x}) = \boldsymbol{f}^{\boldsymbol{T}}_{\boldsymbol{majority}}\,\mathrm{x} = x_3 + x_5 + x_6 + x_7$ and

$$A_{eq} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

We state the following proposition. The proof is given in the appendix.

PROPOSITION 1. *Let $s_{min}(d)$ and $s_{max}(d)$ represent the lower and upper bounds of majority voting i.e. solutions of the LP given by Equations (2.1- 3.5) with the objective minimized and maximized respectively where $d = [p_1, p_2, \ldots p_K]^T$. Then,*

$$s_{min}(d) + s_{max}(\vec{1}\text{-}d) = 1 \tag{2.9}$$

This proposition holds for case 1 (odd no. of classifiers) and case 2 (even no. of classifiers, equal priors and no rejection permitted). Visually, it means that the plots of the upper and lower bounds are inverted mirror images of each other as can be seen in Figures 2.1 and 2.2.

2.3 Results and discussion

In this section we describe the results of our experiments. The results may be categorized as :

1. **Theoretical upper and lower bounds.**

2. **Relationship between diversity measures and majority voting performance.**

   We use the Linear Programming framework described in this chapter for analyzing whether there is a useful relationship between selected classifier diversity measures and majority voting accuracy. This is skipped here but discussed in detail in the next chapter.

3. **Improving classification accuracy.**

We define an **Improvement pattern** as one in which the majority vote is greater than that of the best classifier in the pool (improvement over the best classifier) and a **non-improvement pattern** as one in which majority vote accuracy is not greater than the best classifier in the pool.

2.3.1 Theoretical upper and lower bounds

The theoretical upper and lower bounds of majority voting for 3,5,7 and 9 classifiers are illustrated in Figure 2.1. The plots corresponding to even (4,6,8 and 10) classifiers (no rejection) are shown in Figure 2.2. The objective function in the case of even classifiers with rejection is non-linear and the solution may involve a number of local minima. We note that $g(\mathrm{x}) = \frac{f^T_{majority} x}{f^T_{majority} x + f^T_{minority} x} = \frac{1}{1+\frac{f^T_{minority} x}{f^T_{majority} x}}$. We could for instance separately minimize $f^T_{majority} x$ subject to constraints given by (2.2) and maximize $f^T_{minority} x$ subject to the same

constraints (2.2). If $s1_{min}$ and $s2_{max}$ are the corresponding solutions the quantity $\frac{1}{1+\frac{s2_{max}}{s1_{min}}}$ would then be a conservative lower bound on $g(\mathrm{x})$. In the results illustrated in Figures 2.1 and 2.2 p (x-axis) represents the accuracy of a single classifier in the pool. For purposes of illustration, all classifiers are assumed to have the same accuracy p. p was varied from 0 to 1 and the corresponding lower and upper bounds of majority voting were determined for each value of p. The curve inside the region bounded by the upper and lower bounds represents the majority vote accuracy if the classifiers were independent. The results indicate that even with a set of reasonably competent classifiers it is theoretically possible to obtain dramatic improvement in accuracy. It is observed that the difference between the upper and lower bounds decreases for increasing p indicating that higher the competence of the experts in the pool, better the worst case scenario is.

With respect to the role played by the number of classifiers the results shown in Figures 2.1 and 2.2 may be intuitively explained as follows. The number of constraints is linear in K (number of classifiers) whereas the dimension of the vector x is exponential in K . Thus with increasing K the ”degrees of freedom” increase exponentially. Hence for a particular accuracy of the classifiers (p) if we increase the number of classifiers, we should be able to obtain a ”better” solution (i.e. a lower theoretical minimum and a higher theoretical maximum). The above argument also explains the fact that a given theoretical maximum (say 1) can be obtained for successively smaller values of classifier accuracy (p). For example, the smallest value of p for which the theoretical maximum accuracy is 1 is p=0.66,0.6,0.58 and 0.55 for 3,5,7 and 9 classifiers respectively. It is also easily seen that for a given value of p the majority vote accuracy for independent classifiers increases with K. This still leaves open the question of how to enforce

statistical independence in a classifier ensemble. Therefore, these results by themselves do not provide persuasive arguments for deciding the number of classifiers.

2.3.2 Improvement in classification accuracy

An important goal in building a classifier ensemble is to obtain improvement over the best classifier in the pool (i.e. to obtain an improvement pattern as defined earlier). Let p_{maj}^{min} and p_{maj}^{max} be the theoretical lower and upper bounds respectively (as determined by the solutions of the respective linear programs). Assuming that the joint probability vector x (section 2.2.1) is uniformly distributed over the space of feasible solutions we have, Probability of improvement $= \frac{p_{maj}^{max} - p_{best}}{p_{maj}^{max} - P_{maj}^{min}}$, where p_{best} is the accuracy of the best classifier in the pool.

The plot of probability of improvement vs. classifier accuracy (p) (assuming uniform distribution over the space of feasible solutions) for 3,4,5 and 6 classifiers are shown in Figure 2.3. It can be seen that the maximum probability of improvement occurs for p (accuracy of a single classifier) in the range [0.7 : 0.9]. This corresponds to a scenario of having a team of reasonably competent classifiers (significantly better than random but not very high), indicating that a high accuracy could be potentially obtained from a team of reasonably competent classifiers.

The theoretical bounds of majority vote accuracy may be found as solutions to the linear program given by equations (2.1) - (3.5), however this does not provide a direct answer as to how to achieve these bounds. Tumer and Ghosh [59] emphasize that the selection and training of classifiers that will be combined is as critical an issue as selection of the combination mechanism. If different classifiers do not misclassify the same set of data points, higher is the accuracy of the ensemble since misclassifications by one classifier could be compensated for by the others. Hence the key is enforcing ”diversity”/complementarity in a classifier ensemble. We

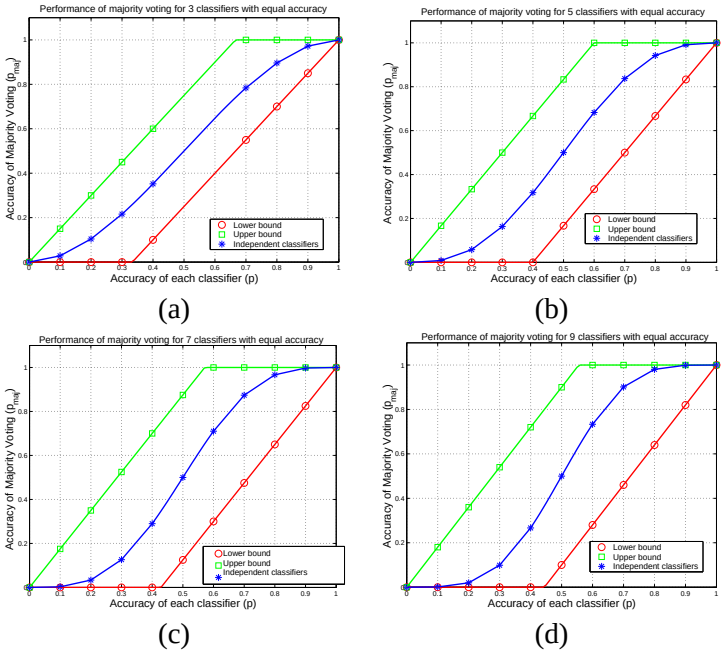


Fig. 2.1. Majority voting accuracy(Odd number of classifiers) : Theoretical lower,upper bounds and independent classifiers for (a) 3 Classifiers (b) 5 classifiers (c) 7 Classifiers (d) 9 classifiers with equal accuracy

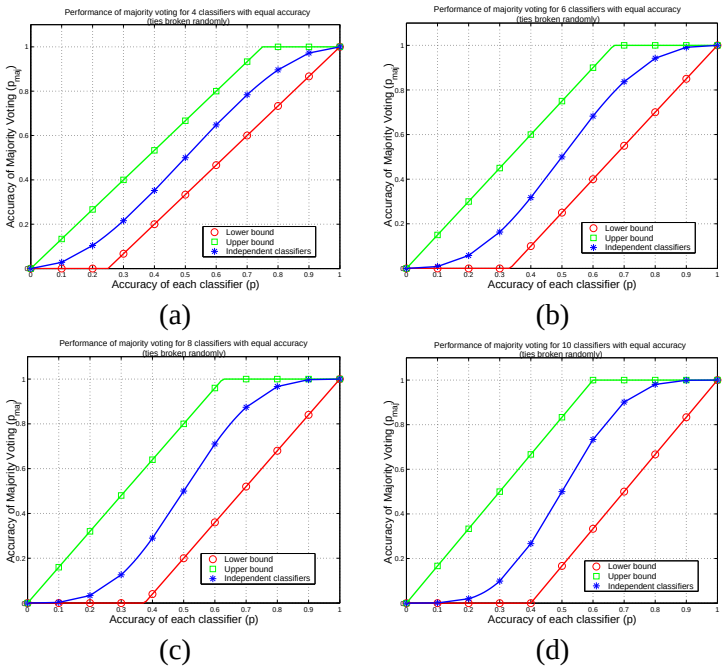


Fig. 2.2. Majority voting accuracy (Even classifiers, no reject) : Theoretical lower,upper bounds and independent classifiers for (a) 4 Classifiers (b) 6 classifiers (c) 8 Classifiers (d) 10 classifiers with equal accuracy

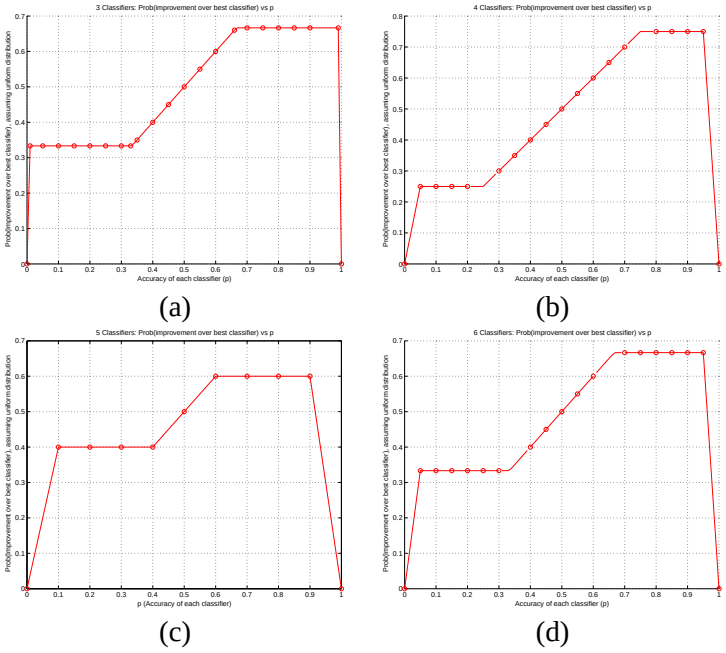


Fig. 2.3. Probability of improvement, assuming a uniform distribution over the space of feasible solutions for (a) 3 (b) 4 (c) 5 (d) 6 classifiers

could hope to achieve this by employing classifiers based on different methodologies, different feature sets or even different sensors. For example, Chandroth [11] investigates methodological diversity for engine fault diagnosis. Four sets of fault classifiers are designed, three sets based on vibration data and employing different features, and the fourth based on pressure data (different sensor). Their results show that classifiers chosen from different methodologies tended to yield better results when combined using majority voting than classifiers chosen from within the same methodology.

In bagging [8] an ensemble is formed by classifiers trained on different training sets obtained from the original training set by a bootstrap sampling procedure. The outputs of the classifiers are combined by the plurality vote. The ”diversity” in the ensemble is created by using the different training sets, the process of creating them imitates random generation from the data distribution. A boosting strategy such as the AdaBoost [22] is another practical method to enforce complementarity, since successive classifiers are trained on patterns that have been misclassified by earlier classifiers. For a detailed analysis of these methods the reader may consult papers such as [3] where an empirical comparison of voting classification algorithms is performed.

2.4 Conclusions

In this chapter we formulate the problem of finding the theoretical lower and upper bounds of majority vote performance for a binary classification problem, as an optimization problem with linear constraints. The problem formulation makes no assumptions on the independence of classifiers. The theoretical bounds could serve as a baseline for evaluation and the framework may be used for performing a variety of analyses related to voting. Using the

framework we provide a few insights about majority voting as described in the relevant sections and also investigate the relationship between two classifier diversity measures and majority vote accuracy.

Proof of Proposition 1 (section 2.2 equation 2.9)

For the proof of the proposition, we need to prove a property of the vectors $\boldsymbol{b_1}, \ldots \boldsymbol{b_K}$, $\boldsymbol{f_{majority}}$ and $\boldsymbol{f} = \boldsymbol{f_{majority}} + 0.5\boldsymbol{f_{tie}}$.

Let $\boldsymbol{b}$ be a bit string and let $\boldsymbol{b}$(i) denote the bit in the $i\ th$ position of $\boldsymbol{b}$ ($0 \leq i \leq 2^K - 1$), let $\boldsymbol{b^c}$ denote a bit string obtained by switching bit at position i with bit at position $2^K - 1 - i$, i.e.$\boldsymbol{b^c}$(i) = $\boldsymbol{b}(2^K - 1 - i)$. For a vector $\boldsymbol{v}$ which is linear combination of such bit strings, i.e. if $\boldsymbol{v} = c_1\boldsymbol{b_1} + \ldots + c_K\boldsymbol{b_K}$ we define $\boldsymbol{v^c}$ to be $c_1\boldsymbol{b_1^c} + \ldots + c_K\boldsymbol{b_K^c}$.

PROPERTY 1.

$$(\boldsymbol{f^c})^{\boldsymbol{T}} x = 1 - \boldsymbol{f^T} x \tag{2.10}$$

Proof : From equation 3.6 it can be verified that all the vectors $\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots \boldsymbol{b_K}$ satisfy Property 1. For $\boldsymbol{f_{majority}}$ noting that if number of 1s in $bit(i, K) > K/2$ then number of 1s in $bit(2^K - i - 1, K) < K/2$, since $2^K - i - 1$ is in fact the 1s complement of i, we have,

if $\boldsymbol{f_{majority}}$(i) = 1 then $\boldsymbol{f_{majority}}$($2^K$ - i - 1) = 0. Hence $(\boldsymbol{f^c_{majority}})^{\boldsymbol{T}}$ x = 1 - $\boldsymbol{f^T_{majority}}$ x

Consider the vector $\boldsymbol{f}$ where $\boldsymbol{f^T}$ x= $\boldsymbol{f^T_{majority}}$ x + 0.5 $\boldsymbol{f^T_{tie}}$ x. For odd K $f_{tie}(i)$ is always 0. i.e. if the number of bits is odd, there can never be a tie. For even K we observe that if number of 1s in $bit(i, K) = K/2$ then number of 1s in $bit(2^K - i - 1, K)$ is also $K/2$, and

thus if $f_{tie}(i) = 1$ then $f_{tie}(2^K - 1 - i) = 1$. Note that $f_{majority}$(i) and f_{tie}(i) cannot both be 1 simultaneously. We have the following possibilities:

- $\boldsymbol{f_{majority}}$(i) = 0,$\boldsymbol{f_{tie}}$(i) = 0 Here $\boldsymbol{f}$(i) = 0 and,

$$\boldsymbol{f}(2^K - 1 - i) = \quad \boldsymbol{f_{majority}}(2^K - 1 - i) + 0.5\boldsymbol{f_{tie}}(2^K - 1 - i) \quad = 1$$

- $\boldsymbol{f_{majority}}$(i) = 1,$\boldsymbol{f_{tie}}$(i) = 0 Here $\boldsymbol{f}$(i) = 1 and,

$$\boldsymbol{f}(2^K - 1 - i) = \quad \boldsymbol{f_{majority}}(2^K - 1 - i) + 0.5\boldsymbol{f_{tie}}(2^K - 1 - i) \quad = 0$$

- $\boldsymbol{f_{majority}}$(i) = 0,$\boldsymbol{f_{tie}}$(i) = 1 Here $\boldsymbol{f}$(i) = 0.5 and,

$$\boldsymbol{f}(2^K - 1 - i) = \quad \boldsymbol{f_{majority}}(2^K - 1 - i) + 0.5\boldsymbol{f_{tie}}(2^K - 1 - i) = \quad 0 + 0.5 = 0.5$$

In all the cases, f(2^K-1-i) = 1-f(i) and thus it follows that $(\boldsymbol{f^c})^{\boldsymbol{T}}$ x = 1 - $\boldsymbol{f^T}$ x

Let LP1 denote the Linear program specified by equation 2.1 and constraints 2.2 for cases 1) and 2) (section 2.2.2). Let $s_{min}(p)$ = min $\boldsymbol{f^T}$ x be the solution of LP1. By property 1, we have $(\boldsymbol{f^c})^{\boldsymbol{T}}$ x = 1 - $\boldsymbol{f^T}$ x and thus max $(\boldsymbol{f^c})^{\boldsymbol{T}}$ x = 1 - min $\boldsymbol{f^T}$ x

Consider the Linear program LP2 where the objective is max $(f^c)^T x$ with the same set of constraints as LP1 (specified by equation 2.2). Taking 1 - both sides, for all constraints we get LP2 as in Table 2.2-(A). It can be seen that Linear Program LP2 is exactly equivalent to LP3 given by Table 2.2-(B). This proves the proposition.

$\max \quad (\boldsymbol{f^c})^{\boldsymbol{T}}\mathrm{x}$ $\text{s.t.} \ (\boldsymbol{b^c_1})^{\boldsymbol{T}}\mathrm{x} \quad = \quad 1-p_1$ $\vdots \quad \vdots$ $(\boldsymbol{b^c_K})^{\boldsymbol{T}}\mathrm{x} \quad = \quad 1-p_K$ $\vec{1}^T\mathrm{x} \quad = \quad 1$ $0 \le x_i \le 1 \qquad (0 \le i \le 2^K-1)$	$\max \quad \boldsymbol{f^T}\mathrm{x}$ $\text{s.t.} \ \boldsymbol{b^T_1}\mathrm{x} \quad = \quad 1-p_1$ $\vdots \quad \vdots$ $\boldsymbol{b^T_K}\mathrm{x} \quad = \quad 1-p_K$ $\vec{1}^T\mathrm{x} \quad = \quad 1$ $0 \le x_i \le 1 \qquad (0 \le i \le 2^K-1)$
(A) Linear Program LP2	**(B) Linear Program LP3**

Table 2.2. **Proof of proposition 1 : Linear Programs LP2 and LP3**

Chapter 3

Evaluation of diversity measures for binary classifier ensembles

Summary

Diversity is an important consideration in classifier ensembles, it can be potentially exploited in order to obtain a higher classification accuracy. There is no widely accepted formal definition of diversity in classifier ensembles, thus making an objective evaluation of diversity measures difficult. We propose a set of properties and a linear program based framework for the analysis of diversity measures for ensembles of binary classifiers. Although we regard the question of what exactly defines diversity in a classifier ensemble as open, we show that the framework can be used effectively to evaluate diversity measures. We explore whether there is a useful relationship between the selected diversity measures and the ensemble accuracy. Our results cast doubt on the usefulness of diversity measures in designing a classifier ensemble, although the motivation for enforcing diversity in a classifier ensemble is justified.

Related publication : [44]

3.1 Introduction

Diversity is defined in different ways in various fields [36, Chapter 10]. Rao [49] provides an axiomatic definition based on a comprehensive study on diversity in life sciences. In Software

Engineering diversity is formulated in terms of coincident failure of different program (software) versions on a random input (e.g. Littlewood & Miller [40] and Partridge & Krzanowski [47]).

In the context of classification, one possible classification of diversity measures is pairwise and non-pairwise measures. In the case of pairwise measures, for a multiple classifier system (MCS) usually the average value of the measure (averaged over the number of pairs) is taken as a measure of diversity of the ensemble. Examples of pairwise diversity measures include the Q-Statistic [61], the Double Fault [24] and the Disagreement measure [55]. Non-pairwise measures are defined on the ensemble as a whole, examples of these are the Kohavi-Wolpert measure [33], the Entropy measure [14] and the measure of difficulty (θ) [25]. Another class of diversity measures arises from the bias-variance decomposition of the ensemble error. Examples of this class of measures include the Coincident Failure Diversity(CFD) and the Distinct Failure Diversity (DFD) due to Partridge & Krzanowski [47].

Kuncheva and Whitaker [38] list measures (both pairwise and non-pairwise) proposed in different contexts and examine how they can be adapted as diversity measures for classifier ensembles. They also examine the relationship between the diversity measures and ensemble accuracy. They conclude that the relationship between the diversity measures and the combination methods is somewhat ambiguous.

Although diversity and complementarity are considered desirable characteristics of a classifier ensemble, a lack of a widely accepted formal definition of diversity renders the evaluation of diversity measures difficult. In [43] we show that the theoretical upper and lower bounds of majority voting performance for a binary classification problem are solutions of a linear program (LP). In this chapter we propose a framework based on the linear programming

formulation for evaluation of diversity measures for binary classifier ensembles. We also propose a set of properties for a diversity measure. Although we regard the question of what exactly defines diversity in a classifier ensemble as open, we show that the framework can be used effectively to evaluate whether diversity measures proposed in different contexts are suitable for classifier ensembles.

This chapter is organized as follows. The framework is described in section 3.2. Diversity measures for classifier ensembles are discussed in section 3.3.1. An important motivation for enforcing diversity in an ensemble is to obtain an improvement in classification accuracy. The characterization of the role of ”diversity” in majority voting (or classifier combination schemes in general) is not straightforward, owing to a lack of a widely accepted formal definition of classifier diversity. The proposed framework enables evaluating whether there is a useful correlation between a given ”diversity” measure and majority voting accuracy. This is discussed in section 3.3.3. Conclusions are presented in section 3.4.

3.2 Description of the framework and proposed properties

We introduce the notation used in the rest of the chapter. Each binary classifier is represented by a bit (1 or 0) with 1 indicating that the classifier is correct and 0 indicating incorrect. The joint statistics can be represented by bit combinations. We follow the convention that if there are K classifiers $C1, C2, \ldots, CK$, $C1$ is the LSB (least significant bit) and CK corresponds to the MSB (most significant bit).

3.2.1 Notation and definitions

DEFINITION 1. *1. Let $bit(i, K)$ represent the K bit binary expansion of i ($0 \leq i \leq 2^K - 1$). and*

2. N(b) = Number of 1s in b, where b is a binary string. (in vector form N(b) = $\vec{1}^T b$).

DEFINITION 2. *Let x=$[x_0, x_1, \ldots, x_{(2^K-1)}]^T$ be the vector of probabilities of the joint correct/incorrect classifications (since there are 2^K possible combinations for K classifiers), where x_i ($0 \leq i \leq 2^K - 1$) is the probability of the correct/incorrect classification of the K classifiers represented by the bit combination $bit(i, K)$. For example if K = 3, x = $[x_0, x_1, \ldots, x_7]^T$; then x_3 = P(bit(3,3)) = P(C3=0,C2=1,C1=1)=P(C3 incorrect,C2 correct,C1 correct).*

DEFINITION 3. *We define a* **configuration** *C as a discrete probability distribution over the set $\{0, 1, \ldots, 2^K - 1\}$ where K is a parameter. i.e. C is a set of tuples of the form $< i, x_i >$ where x_i is the weight (probability) associated with i, $0 \leq i \leq 2^K - 1$ and K is a parameter.*

DEFINITION 4. **Complementary configuration**

If $C = \{< i, x_i >\}$ $0 \leq i \leq 2^K - 1$ is a configuration, the complementary configuration $\bar{C}$ is defined as follows:

$$\bar{C} = \{< i, x_i^{(\bar{C})} >\} \text{where } x_i^{(\bar{C})} = x_{2^K-1-i}^{(C)}$$

3.2.2 Proposed properties

PROPERTY 1. **The diversity measure must have a finite value for all configurations.**

We strongly recommend that a diversity measure satisfy Property 1. In addition we propose the following desirable properties. We do not claim that these are the most useful, rather this is one possible set of intuitive properties.

PROPERTY 2. *A desirable property would be that the measure have a minimum and a maximum value.*

PROPERTY 3. *It is preferable that the diversity measure be capable of being expressed as an* **easily computable closed form function of the joint probability vector x**. *The minimum and maximum values of the measure for a particular ensemble can then be determined as solutions to the optimization problem in section 3.2.3.*

Diversity measures may either be symmetrical or non-symmetrical with respect to correct and incorrect classifications (0 and 1) [51]. Although we list the symmetry property here, not satisfying the symmetry property is not necessarily undesirable or disadvantageous. We state the symmetry property formally below.

PROPERTY 4. **Symmetry property**

A diversity measure satisfies the symmetry property if it is symmetrical with respect to correct or incorrect decisions for the entire configuration. Mathematically this can be stated as : "The diversity measure of a configuration and its complement (defined in section 2.2.1) is the same."

3.2.3 Problem formulation

For an ensemble of K binary classifiers with accuracies $p_1, p_2, \ldots, p_K$, if the diversity measure can be expressed as a closed form function g(x) of the joint probability vector x (Definition 2), the theoretical bounds may be derived as solutions to an optimization problem with linear constraints. g(x) is the objective function to be maximized/minimized subject to the constraints specified by equations (3.2-3.4). For the details the reader may refer [43] where we formulate the optimization problem to determine the theoretical bounds of majority vote accuracy for a given ensemble of classifiers (since the constraints are identical).

$$\text{max (min)} \;\; g(\mathrm{x}) \tag{3.1}$$

$$\textbf{s.t.} \quad A_{eq}\mathrm{x} = d \tag{3.2}$$

$$0 \leq x_i \leq 1 \qquad 0 \leq i \leq 2^K - 1 \tag{3.3}$$

$$\vec{1}^T\mathrm{x} = 1 \tag{3.4}$$

$$\begin{aligned} \text{where } d &= [p_1, p_2, \ldots, p_K]^T \text{(vector of classifier accuracies)} \\ \mathrm{x} &= [x_0, x_1, x_2, \ldots, x_{(2^K-1)}]^T \text{ (vector of joint probabilities)} \\ \text{and } A_{eq} &= [b_1, b_2, \ldots, b_K]^T (K \times 2^K \text{ matrix of equality constraints,} \\ &\quad \text{row } r \text{ corresponds to } r \text{ th classifier.)} \end{aligned} \tag{3.5}$$

$$
\begin{aligned}
\boldsymbol{b_1} &= [0\ 1, \ldots, 0\ 1]^T \\
\boldsymbol{b_2} &= [0\ 0\ 1\ 1, \ldots, 0\ 0\ 1\ 1]^T \\
&\vdots \\
\boldsymbol{b_K} &= [\overbrace{0\ 0 \ldots 0\ 0}^{2^{(K-1)}}, \overbrace{1\ 1 \ldots 1\ 1}^{2^{(K-1)}}]^T
\end{aligned}
\tag{3.6}
$$

Some of the diversity measures discussed in [38] may be expressed as linear functions of x i.e. $g(\mathrm{x}) = \boldsymbol{f^T}\mathrm{x} + c$ where $\boldsymbol{f} = [f_0, f_1, \ldots, f_{(2^K-1)}]^T$. These are listed in Table 3.3. The derivation is omitted due to space constraints. In these cases, the optimization problem is a Linear Program (LP).

3.3 Results and discussion

In this section we evaluate the measures discussed in [38] based on the framework described in section 3.2. We provide short definitions of some of the measures. For a more detailed overview the reader may refer [38, Sections 3,4 and Table 2]. We briefly discuss pairwise measures. Although we conducted similar experiments on pairwise measures, due to space constraints we only present sample results related to two non-pairwise measures, Cunningham measure [14] and measure of "difficulty" [25]. The analysis and experimental procedure are the same for the other measures and many of the discussions and conclusions also apply to them.

3.3.1 Pairwise vs non-pairwise diversity measures

Pairwise classifier diversity measures: Four pairwise measures (for a pair of classifiers) are listed in Table 3.2. They are defined with respect to the confusion matrix shown in table 3.1. For multiple classifiers, the average value of the pairwise diversity measure (averaged over the total number of pairs) is taken as a measure of diversity of the ensemble. Kuncheva and

	Classifier $D_2 \rightarrow$	
Classifier $D_1 \downarrow$	D_2 Incorrect	D_2 Correct
D_1 incorrect	N^{00}	N^{01}
D_1 correct	N^{10}	N^{11}

Table 3.1. Confusion matrix for a pair of classifiers.

Whitaker [38] suggest that pairwise measures may not be useful in the case of unequal pairwise distributions. We suggest other reasons why they may not be suitable and thus do not recommend their use for classifier ensembles.

- The Q-Statistic and the correlation coefficient do not satisfy the Property 1 listed in section 3.2.2. For example if both N^{00} and N^{10} are both 0 the Q-Statistic and correlation coefficient are undefined ($\frac{0}{0}$).

- The Disagreement measure,Q-Statistic and the correlation coefficient are symmetrical for a pair of classifiers, however they are not necessarily symmetrical with respect to the entire

Diversity measure	Definition
Q Statistic [61]	$Q = \frac{N^{11}N^{00} - N^{10}N^{01}}{N^{11}N^{00} + N^{10}N^{01}}$
Correlation coefficient [56]	$\rho = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11}+N^{10})(N^{01}+N^{00})(N^{11}+N^{01})(N^{10}+N^{00})}}$
Disagreement measure [55]	Dis = $\frac{N^{01}+N^{10}}{N^{00}+N^{01}+N^{10}+N^{11}}$
Double fault measure [24]	DF = $\frac{N^{00}}{N^{00}+N^{01}+N^{10}+N^{11}}$

Table 3.2. Definitions of 4 selected pairwise diversity measures

configuration i.e. the average Q-statistic (or correlation coefficient) for a configuration and its complement (defined in section 2.2.1) are not necessarily the same.

- Although the double-fault measure does not violate Properties 1 and 2, it may not capture the aspects of classifier diversity which may be regarded important. Consider for example configurations A_1 and A_2 as below.

  A_1:$x_0 = 0.5, x_1 = 0.3, x_2 = 0.2, x_3 = 0$; A_2:$x_0 = 0.5, x_1 = 0, x_2 = 0, x_3 = 0.5$

  Intuitively, A_1 is more diverse than A_2 (in A_2, there is total agreement with respect to all classifications), however the double fault value is the same for both cases.

- With pairwise measures, it is usually hard to express the diversity measure of a given ensemble as a simple closed form function of x. The objective function g(x) may be complicated, and hence it is difficult to determine the range of values for a given ensemble. Thus, they may not be amenable for analysis and evaluation.

Non-pairwise diversity measures:

1. **Cunningham (Entropy) measure [14]:** In [14] a diversity measure is proposed and is referred to as the Entropy measure. Since it is quite distinct from the entropy function in information theory, we refer to it as the Cunningham measure. In [38] it is defined as follows. Let z_j $j = 1, \ldots, N$ be examples classified by the classifiers and $l(z_j)$ the number of classifiers that correctly classify z_j. Let K be the number of classifiers. The Cunningham measure may be defined as :

$$E = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{(\lfloor K/2 \rfloor)} min\{l(z_j), L - l(z_j)\} \tag{3.7}$$

   In the limiting case the proportion of examples is the probability. With our notation the Cunningham measure may be defined as:

$$E = \frac{1}{\lfloor K/2 \rfloor} \sum_{i=0}^{2^K - 1} min(n, K - n) x_i \tag{3.8}$$

   where $n = N(bit(i, K))$ i.e. n = Number of 1s in K bit binary expansion of i.

2. **Measure of "difficulty" (θ):** Hansen and Salamon [25] propose a measure as follows. Let the number of classifiers be K and let X be the random variable which denotes the fraction of classifiers that correctly classify a random input. Thus X is a discrete random variable which can take on the values $\{\frac{0}{K}, \frac{1}{K}, \ldots, \frac{K}{K}\}$. The variance of X (θ = Var(X)) is proposed as the measure of diversity.

The Cunningham measure satisfies the symmetry property as defined in section 3.2.2 while Hansen's measure of difficulty does not.

Diversity measure	g(x)=$\boldsymbol{f}^{\boldsymbol{T}}$ x+ c	
	$\boldsymbol{f} = [f_0, f_1, \ldots, f_{(2^K-1)}]^T$	c (constant)
Cunningham measure (E) [14]	$f_i = min(n, K-n)/\lfloor K/2 \rfloor$ where $n = N(bit(i,K))$	0
Measure of difficulty (θ) [25]	$f_i = \left(\frac{N(bit(i,K))}{K}\right)^2$	$-p^2_{mean}$ where p_{mean}= Avg. classifier accuracy
Kohavi-Wolpert variance (KW) [33]	$f_i = \frac{N(bit(i,K))}{K}\frac{K-N(bit(i,K))}{K}$	0

Table 3.3. Diversity measures expressed as functions of x

3.3.2 Variation of selected diversity measures with classifier accuracy

The results for the Cunningham measure and Hansen's measure of "difficulty" (θ) are shown in Figure 3.1. The Cunningham measure ranges from 0 (lowest diversity,complete agreement) to 1 (highest diversity). For the measure of difficulty, higher values correspond to lower diversity and vice versa. The curve in the middle corresponds to the value of the diversity measure, if the classifiers were statistically independent.

It is interesting to note that for statistically independent classifiers the value is on the side of higher "diversity". For the Cunningham measure it is much closer to the maximum while for the Hansen's measure of difficulty it is closer to the theoretical minimum (again note that for the Cunningham measure, higher values correspond to high diversity while for the measure of difficulty higher values correspond to lower diversity and vice versa). Although this seems to indicate that statistical independence bodes well for the classifier ensemble, it still leaves open the question of how to enforce statistical independence. For instance, Eckhardt and Lee [19]

point out that software programs developed independently tended to fail on similar inputs, this being related to the difficulty of the specific inputs.

The absolute value of a diversity measure may have limited use even if the measure has a theoretical minimum and maximum value, since the range of values it can take depends on the particlular ensemble. Consider for example, ensembles E1 and E2 each consisting of three classifiers with the following accuracies:

E1 : $p_1 = p_2 = p_3 = 0.9$ and E2 : $p_1 = p_2 = p_3 = 0.6$

Although in general the theoretical minimum and maximum values of the Cunningham measure are 0 and 1 respectively, we can see from Figure 3.1(a) that the corresponding values for ensemble E1 are 0 and 0.3 and for E2 they are 0 and 1 respectively. Let the actual values of the Cunningham measure for E1 and E2 be 0.25 and 0.5 respectively. E1 is *relatively more diverse* than E2 even though the absolute value is smaller, since the diversity value is relatively closer to its theoretical maximum. Thus it may be more useful to consider *relative diversity* as opposed to the absolute value of the diversity measure, especially if we wish to compare different ensembles. For a given set of classifier accuracies, we define the relative diversity (D_{rel}) as: $D_{rel} = \frac{D_{actual} - D_{min}}{D_{max} - D_{min}}$, where D_{min},D_{max} and D_{actual} are the theoretical minimum,theoretical maximum and actual value of the given diversity measure, respectively.

3.3.3 Exploring the relationship between diversity measures and majority voting performance

A motivation for designing a ”diverse” ensemble of classifiers is to obtain an improvement in classification accuracy. Since there is no widely accepted formal definition of ”diversity”, the characterization of the relationship between ”diversity” and majority voting (or

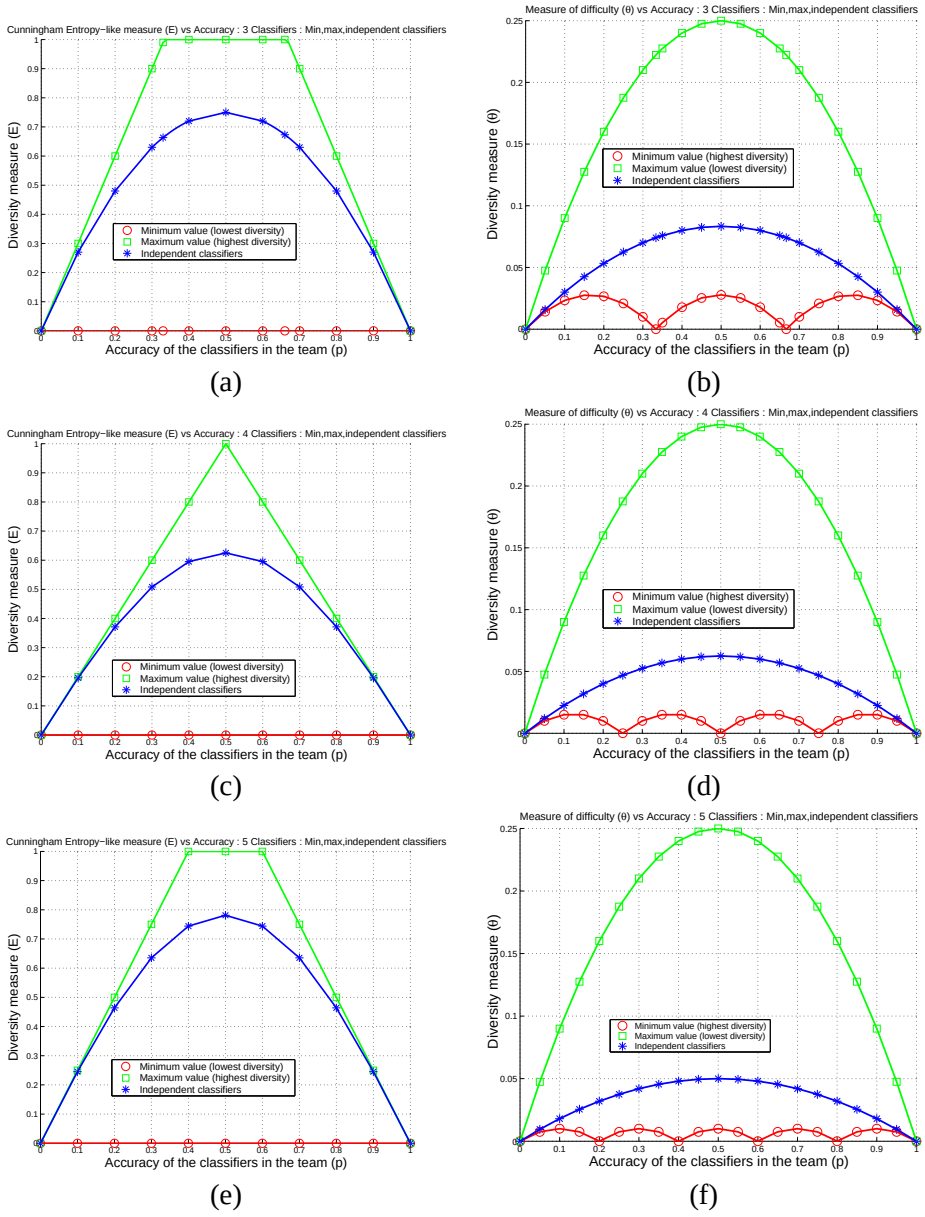


Fig. 3.1. Cunningham Measure (E) and Measure of difficulty (θ) vs classifier accuracy (p) : Lower,upper bounds and independent classifiers for 3 (a),(b) 4 (c),(d) 5 (e),(f) classifiers
E vs p : Figures (a),(c),(e)
θ vs p : Figures (b),(d),(f)

classifier combination schemes in general) is not straightforward. Nevertheless, the framework formulated in the chapter enables evaluating whether there is a useful correlation.

The main problem encountered in analyzing the relationship between a diversity measure and majority vote accuracy (p_{maj}) is to vary a given variable in a systematic manner over its full range in order to compute the corresponding value of the other variable. One could use a simulation scheme and an enumeration experiment such as in [38]. However this does not necessarily provide a means for varying a quantity in a systematic manner over its full range. The optimization problem framework addresses this problem. The experimental procedure is outlined below. For simplicity and purpose of illustration, all classifiers are assumed to have the same accuracy p. Let p_{maj} denote the majority vote accuracy.

Experimental procedure

1. Vary p in steps. For each p determine the majority vote theoretical upper and lower bounds p_{maj}^{max} and p_{maj}^{min} respectively by solving the linear program (LP) in [43] (constraints identical to the optimization problem in section 3.2.3).

2. Vary p_{maj} in steps from p_{maj}^{min} to p_{maj}^{max}. For each p_{maj} obtain a feasible solution i.e. a solution which satisfies the constraints given by Equations (3.2-3.4). Determine the value of the diversity measure (D) corresponding to the solution.

3. Find the coefficient of linear correlation between majority vote accuracy (p_{maj}) and the value of the diversity measure (D).

It is more useful to look at the overall trend instead of the actual values of the diversity measures (especially if the classifier accuracies are not the same). For the most part, the variation of

the diversity measures was in line with their basic motivations. For example, the coefficient of linear correlation between Hansen's measure of "difficulty" (θ) and p_{maj} was mostly negative and that between the Cunningham measure (E) and p_{maj} was positive. However in general the relationship between the diversity measures and majority vote accuracy (p_{maj}) is hard to characterize.

We repeated the experiments by varying the Cunningham measure between its theoretical minimum and maximum values (which may be determined by solving the linear program in section 3.2.3 with g(x) given in table 3.3) and determining the corresponding majority vote accuracy (p_{maj}). The results are shown in Figure 3.2. As can be seen from figure 3.2 there may be no one-to-one correspondence between the diversity measure and majority vote accuracy. It is entirely possible that there is more than one solution x which satisfies the constraints (3.2)-(3.4). For example in Figure 3.2(a) the values of E vary linearly with p_{maj}, the range of values of p_{maj} corresponding to the range [0.4:0.6] of E is approximately [0.7:0.85]. In Figure 3.2(b) there are multiple solutions where p_{maj} is very close to 0.85 while values of E range from 0.4 to 0.6. These results raise questions about the usefulness of diversity measures in designing a classifier ensemble.

3.4 Conclusions

In this chapter we propose a linear program based framework for the analysis of diversity measures for ensembles of binary classifiers and also a set of properties for such a diversity measure. Although we regard the question of what exactly defines diversity in a classifier ensemble as open, we show that the framework can be used effectively to evaluate diversity measures for classifier ensembles. The framework was used for analyzing the relationship between selected

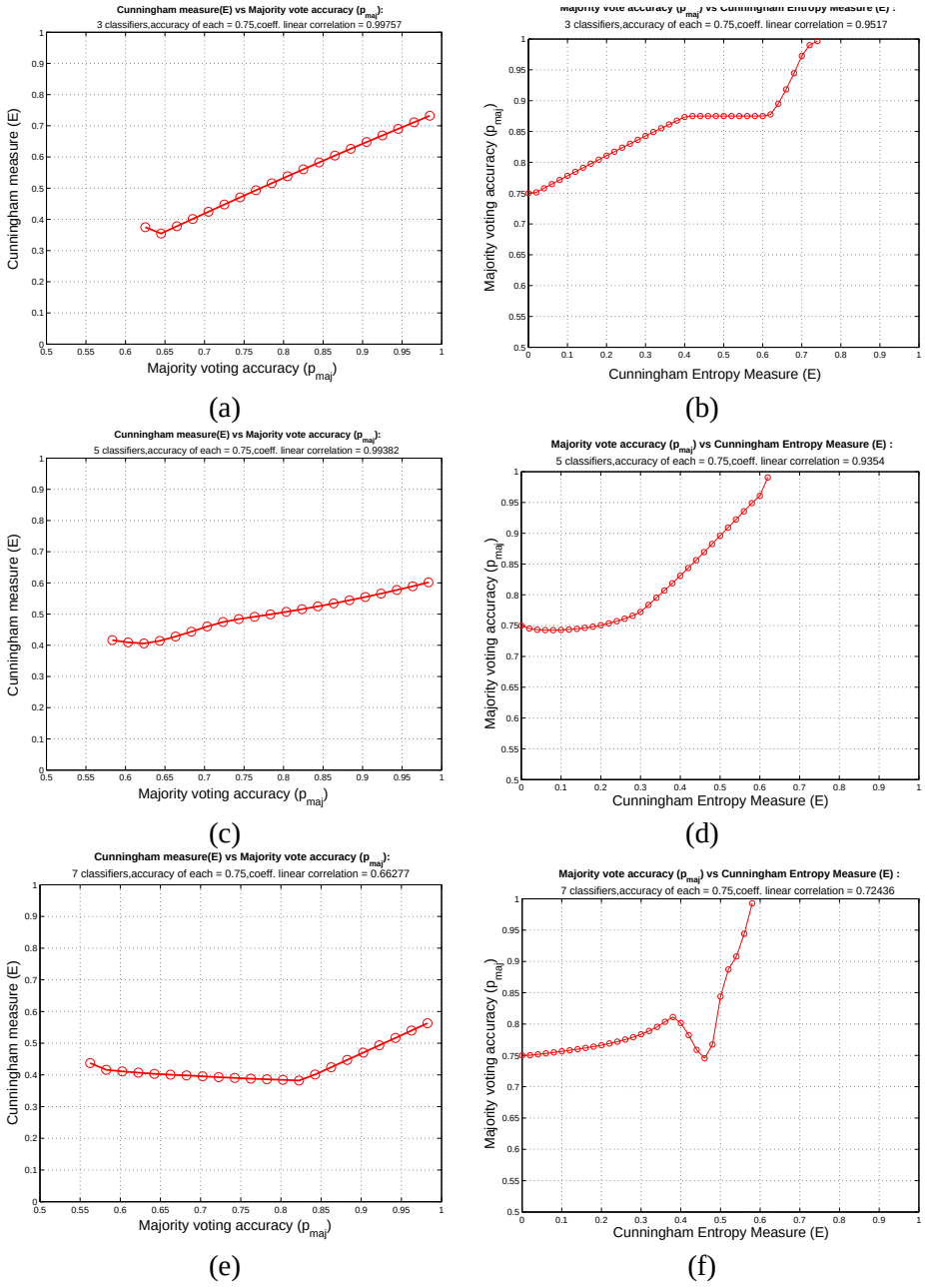


Fig. 3.2. Variation of Majority vote accuracy (p_{maj}) vs Cunningham (Entropy) measure (E) and vice versa :
E vs p_{maj} : Figures (a),(c),(e)
p_{maj} vs E : Figures (b),(d),(f).
for 3 Classifiers (a),(b) 5 classifiers (c),(d) and 7 classifiers (e),(f); with equal accuracy p=0.75

diversity measures and accuracy of the classifier ensemble. Even though the motivation for enforcing diversity in a classifier ensemble is justified,the results cast doubt on whether diversity measures are useful in this regard. Based on our evaluation we suggest that although measures may be useful in the original context they were proposed, caution must be exercised in using them as diversity measures for classifier ensembles.

Chapter 4

Utility analysis of cluster validity indices for genomic data

Summary

In this chapter we examine the suitability and limitations of some of the standard cluster validity indices proposed in the literature for cluster validation in the context of genomic data. We highlight some of the issues in cluster validation through experimental results on yeast datasets. We use the term clustering to refer both to the process as well as the result. Informally, cluster validation may be defined as the process of evaluating the quality of the clusters obtained. This could then be used to select the ”optimal” number of clusters describing the structure of the data as best as possible.

Cluster Validity Indices (CVIs) may be broadly classified as internal, i.e. those which look at the data alone and external i.e. those that involve comparing a clustering against an external clustering. The cluster validation protocol could be considered to be comprised of two stages. The first stage predominantly employs internal CVIs (although in general external CVIs could be used for this stage too) and aims to estimate the ”true” number of clusters in the data which reflects the structure in the data as best as possible. In the second stage, the optimal clustering thus determined may be compared against an appropriately chosen ”reference clustering”. In the context of genomic data, consider for instance the application of a clustering algorithm to gene expression data obtained from microarrays in order to group together genes

with similar expression patterns. The first stage is concerned with identifying the ”optimal” co-expressed clusters, i.e. genes assigned to the same cluster have highly similar expression profiles. Next we may want to compare the clusters thus obtained with a grouping of the same genes based on biological function (this would be the reference clustering), in order to assess whether co-expression in the given dataset relates to similarity in biological function for instance. This would constitute the second stage of the cluster validation protocol. In this chapter we focus on the first stage, in the next chapter we look at the second stage.

Many internal cluster validity indices assume the data to reside in a Euclidean space. Although the assumption of a Euclidean space is somewhat reasonable for gene expression data (even here other distance measures may be more suitable), this may not be the case for other data such as sequence data. Thus it is preferable to include indices applicable to non-Euclidean spaces without modification, among a suite cluster validation methods. Accordingly, we examine which cluster validity indices (internal) are suitable for non-Euclidean spaces.

4.1 Overview of cluster validity indices (CVIs) and techniques

In this section we present a brief overview of some of the existing cluster validation methods.

The literature on cluster validation is quite extensive. One may consult any of the surveys for a detailed overview, for instance Milligan and Cooper [42] conduct a Monte Carlo evaluation of 30 internal indices. Numerous cluster validation techniques especially for gene expression data also abound in the bioinformatics literature. Two examples include a prediction-based re-sampling method for estimating the number of clusters in a dataset, proposed by Dudoit and Fridlyand [17] and a stability based method for discovering structure in clustered data, proposed

by Ben-Hur *et al.* [4]. A number of software packages for clustering, cluster visualization and cluster validation are available. Many of them may be downloaded for free for non-commercial purposes. Cluster and TreeView are an integrated pair of programs developed by Eisen *et al.* [20] used extensively by researchers. They implement many clustering algorithms and provide a graphical visualization of the output. Machaon Cluster Validation Environment [6] is a software package intended for application of different clustering and validation algorithms to gene expression data.

Cluster Validity Indices (CVIs) may be broadly categorized as internal and external, this is by no means an exhaustive categorization of the plethora of cluster validation methods in the literature. Internal CVIs typically optimize a statistic related to the cluster structure such as the tightness of clusters and functions of inter-cluster distances; examples include the Dunn Index [18], Hartigan Index [26], Calinski-Harabasz [10] and the Davies Bouldin Index [15]. Internal indices are often used to determine the ”natural” clusters in the data while rejecting noise. Some internal indices are defined only for the number of clusters ≥ 2, while others are defined even if the number of clusters is 1 (all points assigned to one cluster), still others compare the value of an index for a given clustering with a ”null scenario” (usually the null hypothesis is that of no clustering structure in the data). One such example is the Gap Statistic proposed by Tibshirani [58], a method in which an internal index is compared to its expectation under a reference null distribution. External validation involves comparing a clustering with an external clustering. External CVIs could also be used ”internally” to assess the consistency of partitions generated by different runs of a clustering algorithm for instance. We provide a brief overview of some of the external validity measures in Chapter 5.

How many clusters?

An important consideration in cluster validation is to determine the ”true” number of clusters. One possibility is to use a measure of quality of the clustering and choose the number of clusters which corresponds to an ”optimal clustering”. This could be done by using internal cluster validity indices. Another approach is to use external indices *i*nternally to measure stability and/or internal reproducilibility ([17],[4]). Using the values of the measure of quality, the number of clusters may be determined either by a rule of thumb or by comparing against a reference scenario of no-clustering. Although this sounds simple, it can be quite tricky in practice. There may be no outright winner as to the right number of clusters, there is often more than one reasonable answer. Also different validity indices may not agree on the optimal number of clusters. These problems are illustrated in Section 4.3.

4.2 Limitations of cluster validity indices

We list some of the representative internal cluster validity indices used in our experiments and discuss their usefulness/shortcomings. Many of the CVIs proposed in the literature (both internal and external) have shortcomings which could be particularly restrictive for genomic data. Most internal CVIs assume the data to reside in a Euclidean space, however in many cases other metric spaces are preferable.

- **Hartigan Index [26]**

  Let n be the number of data points and k be the number of clusters. Let $W(k)$ denote the

within clusters sum of squares. The Hartigan Index is defined as :

$$H(K) = \gamma(k)\frac{W(k) - W(k+1)}{W(k)} \tag{4.1}$$

where the correction factor $\gamma(k) = n - k - 1$. Since the within clusters sum of squares is monotonically decreasing, the Hartigan Index indicates the relative improvement when adding an extra cluster (moving from k to $k + 1$). A few salient points of practical importance are summarized below.

- The Hartigan Index is defined for k = 1 (i.e. all points assigned to one cluster). This corresponds to the hypothesis that there is no structure in the data.
- The Hartigan Index itself is not monotone. A rule of thumb would be to identify the cluster number k^* where the index has a high value at $k^* - 1$ and a low value at $k^* + 1$. Although this seems reasonable, often in practice the choices are not clear cut. There may be varying number of cluster numbers which are all reasonable answers.
- The Hartigan Index implicitly assumes data in a Euclidean space. Although one could come up with non-Euclidean equivalents of within cluster sum of squares, the question is whether $W(k)$ is then monotonic.
- Another subtle problem that could be encountered in practice, with algorithms such as k-means for instance is the following. k-means finds the local minimum, thus repeated runs would be preferable. Although the monotonicity is true for global minimum (i.e. $W^*(k) > W^*(k+1)$ where $W^*(k)$ and $W^*(k+1)$ denote the global optima of the within cluster sum-of-squares for k and $k + 1$ clusters respectively),

in practice there may be some cases where the Hartigan Index is negative depending on how $W(k)$ and $W(k+1)$ are computed from the repeated runs.

- **Davies-Bouldin Index [15]**

  Let S_i be an appropriate measure of dispersion of i^{th} cluster and let M_{ij} be the distance between the centres of clusters i and j. Let $R_{ij} = \frac{S_i + S_j}{M_{ij}}$ The Davies Bouldin Index ($\bar{R}$) is given by 4.2.

$$\bar{R} = \sum_{c=1}^{N} R_i \tag{4.2}$$

where,

$$R_i = max_{i \neq j} R_{ij}$$

The Davies Bouldin Index is defined only when the number of clusters ≥ 2. For Euclidean spaces, the standard deviation of the distances of all points of the i^{th} cluster from the respective centroid may be used as the measure of dispersion S_i. Although the Davies-Bouldin Index does not directly impose the restriction of data residing in a Euclidean space, the interpretation of centroid and distances between pairs of centroids is not clear-cut in non-Euclidean metric spaces. Boutin and Hascoet [7] suggest a non-Euclidean version of the Davies Bouldin Index for graph partitioning. They define two intra cluster distances for a cluster C_k which they refer to as the diameter $diam(C_k)$, the complete diameter is the distance between the two most remote nodes and the average diameter as the average distance between all pairs of nodes that belong to the same cluster. They also suggest using for the inter-cluster distance $d(C_i, C_j)$ between clusters C_i and C_j as

single, complete or average linkage. They suggest $diam(C_i)$ for S_i and $d(C_i, C_j)$ for M_{ij}.

- **Silhouette [50]**

  The computation of the silhouette requires as input the dissimilarity between each pair of data points, the dissimilarity can be any appropriate metric and need not be the Euclidean distance. The silhouette may be defined as follows. Let $Cl(i)$ be the cluster to which i is assigned to. Let a_i denote the average dissimilarity between i and all other observations in $Cl(i)$. Let $d(i, C)$ denote average dissimilarity of i to all objects in cluster C. Let

$$b_i = min_{C \neq Cl(i)} d(i, C).$$

  The silhouette width of observation i is then,

$$sil_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{4.3}$$

  The silhouette width lies in the range [-1, 1]. Objects with large silhouette width are better clustered while those with smaller width lie between clusters. Objects with a large negative silhouette width are ”misclustered”. The overall silhouette width is the average sil_i over all observations. Kaufman and Rousseeuw [31] suggest estimating the number of clusters by that which gives the largest average silhouette width.

4.3 Experiments and Results

We show the results corresponding to the cdc cell cycle data by Spellman *et al.* [57] and also a few sample results on the yeast shock data generated by Gasch *et al.* [23]. We performed Principal Component Analysis (PCA) primarily to gain insight into the data and cluster analysis. The experimental procedure for the cluster analysis consisted of two stages. In the first stage we try to estimate the ”true” number of clusters in the data and thus determine the optimal clustering that describes the structure in the data. In the second stage we attempted to compare the ”optimal clustering” obtained in the first stage with a reference clustering. We used the MIPS database [1] to obtain the ”reference clusterings” for each set of genes used in the experiments. In this chapter we focus on the first stage, namely trying to estimate the ”true” number of clusters in a given dataset. The results are described in sections 4.3.1 and 4.3.2 repectively. We used Matlab and R software packages for our results.

4.3.1 Experiments on the yeast cell cycle data by Spellman et al.

In this section we describe experiments and analyses on the cell cycle data illustrated with sample results. The dataset is described below.

cdc yeast cell cycle data by Spellman et al. [57]

The original data consisted of T=15 equi-spaced time points covering more than 2 cycles. The cells were synchronized with the cdc method in order to facilitate observation of cyclic behaviors. The data is from spotted arrays and represents normalized log-ratios. About 800 genes show periodic behavior and are hence cell-cycle related. For our experiments we use a pre processed subset of the data as described below.

- The first 12 time steps of the cdc-cell cycle data were used.
- Among the set of 800 genes, 678 genes without missing values were chosen.
- The data was row standardized i.e. the row mean was 0 and row standard deviation was 1.

Principal Component Analysis (PCA)

The results of Principal Component Analysis (PCA) are shown in Figure 4.1 and summarized below.

- The percent variability and the cumulative percent variability encompassed by the Principal Components (PCs) are shown in Figure 4.1(a) and (b) respectively. We see that the first 3 Principal Components account for about 95% of the variability, strongly suggesting that the data is explained by a relatively small number of ”basic patterns”.
- The first 3 PCs are shown in figure 4.1 (c). The first two PCs are periodic time-varying signals (approximately sinusoidal). This makes intuitive sense since the original data is cell-cycle data. The third PC however is a time-varying signal with increasing amplitude. Since we do not have a reasonable explanation from the experiment for the increasing amplitude we regard it as an artifact and discard it even though the amount of variability is not negligible.

Principal Component Analysis was used for dimensionality reduction. Figures 4.2 and 4.3 show the results of hierarchical clustering on the original 12 dimensional data and on the projections in the space spanned by the first two principal components respectively. A qualitatively better clustering structure was observed when the data was clustered in the low dimensional sub space

spanned by the first 2 Principal Components as compared to clustering in the full 12 dimensions, as suggested by the figures mentioned above. This suggests that discarding the remaining principal components amounts to discarding noise and other artifacts.

Clustering results

The set of genes selected by Spellman et al. were those involved (or at least suspected to be involved) in cell cycle. The cell-cycle consists of five phases each with a time lag with respect to the previous one. Since we hope that the gene expression will reflect the cell cycle behavior, we confine the number of clusters to a window around 5 hence the number of clusters k was varied from 2 to 10. Since there are 5 phases of the cell cycle, it is reasonable to expect 5 clusters. However as we shall see from the results, in many cases 3 and 4 produced much better clusters than 5 clusters.

The data was clustered using the k-means algorithm. 200 runs with different random initializations were conducted for each choice of k. First we clustered the data in the full 12 dimensional space. Next we considered the projections of the data in the space spanned by the first two principal components and clustered this data using k-means following the same procedure as mentioned above (i.e. 200 runs with random initializations). Different Cluster Validity Indices (CVIs) were computed for each of the clusterings. The results are shown in Figures 4.4 (Hartigan), 4.5 (Davies Bouldin) and 4.7 (Silhouette). For each k the Davies Bouldin and the average silhouette width were computed for each of the 200 runs. The mean and standard deviation were computed across the 200 runs. The three bubbles on each stem in Figures 4.5(a)-(b) and 4.7(a)-(b) represent the mean, mean - std. deviation and mean + std. deviation respectively. The Hartigan Index was computed differently, hence there is only one bubble corresponding to each k in Figure 4.4. These computations are explained below.

- **Hartigan Index**

  Figures 4.4(a) and 4.4(b) show the Hartigan Index [26] for the 12 dimensional data and the projections in the space of first two principal components respectively. The Hartigan Index was computed as follows : For a given number of clusters k, among the 200 runs of the k-means algorithm, the lowest value for the within cluster sum of squared distances (SSD) was selected as $W(k)$ and plugged into equation (4.1). Since the Hartigan index for a given k ($H(k)$) indicates the relative improvement when adding an extra cluster (moving from k to $k+1$), one rule of thumb to choose the optimal number of clusters k^*, is to choose k^* where there is a significant drop from $H(k^*-1)$ to $H(k^*)$ but a much smaller drop from $H(k^*)$ to $H(k^*+1)$.

  Using this heuristic, the number of clusters predicted by both clustering using all 12 time points as well as projections in the space of firt 2 principal components is $k^* = 4$ as seen in figures 4.4(a) and 4.4(b). The jumps are more pronounced when clustering is performed in a lower dimensional space as seen in Figure 4.4(b). This is probably due to discarding of noise and other artifacts.

- **Davies Bouldin Index**

  Figures 4.5(a) and 4.5(b) show the Davies Bouldin Index [15] for the 12 dimensional data and the projections in the space of first two principal components respectively. As explained previously, the three bubbles on each stem in Figures 4.5(a) and 4.5(b) represent the mean, mean - std. deviation and mean + std. deviation respectively. Since lower values of the DBI indicate better clustering, a good rule of thumb to choose the ”optimal” clusters, is to choose k^* where there is a trough. For the clustering using 12 time points (Figure

4.5(a)), 3 as well as 4 seem reasonable answers, with the mean value of DBI for 3 clusters being lower. According to Figure 4.5(b), the answer is 4.

This illustrates the practical problems in cluster validation. Depending on the features used, the optimal number of clusters may be different. This is possible, since clustering using different feature sets may yield different structures present in the data. Also there may not be agreement between different cluster validity indices. In the present case, one may argue that there is agreement since k = 4 clusters is a good answer according to both Hartigan and Davies Bouldin indices. Nevertheless, relying on the values of indices alone can lead to ambiguous answers. This could be because there may not be a pronounced clustering structure in the data. The cell cycle data has 5 overlapping phases, thus we may not necessarily see 5 distinct clusters. Figure 4.6 is also illustrative. Here the projections of the data in the 2 dimensional space spanned by the first two principal components are shown. As discussed earlier, using first two principal components is an adequate representation of the data. Different clusters are shown with different colors, the clustering was performed in the 12 dimensional space. When the projections of all data points are considered together, we see that they densely populate a circular area, there are no sharp boundaries, hence clustering the data would be equivalent to ”slicing the pie” into a certain number of pieces.

- **Silhouette**

  We next discuss the results of the Silhouette index. The silhouette permits a visual assessment of the quality of the clustering apart from the values of the index alone. As mentioned before, the k-means algorithm was run 200 times and the average silhouette

width was computed for each run. The average of the 200 values was taken as the average silhouette width for that particular value of k (number of clusters). This was repeated for k varying from 3 to 10. The summarized results are shown in Figure 4.7. In the figure for each k we plot the mean of the average silhouette widths, mean - standard deviation and mean + standard deviation. Sample silhoutte plots are illustrated in Figures 4.8 - 4.9 (these correspond to clustering in the full 12 dimensions) and 4.10 - 4.11 (these correspond to clustering in lower dimensions (2 principal components)). Each sample plot corresponds to one single run.

Relying on the values of the indices to assess the number of clusters can often be deceptive. A case in point is the silhouette results shown in Figure 4.7. The average silhouette width is lower for k = 5 clusters than k = 3 or 4. Hence 5 seems a better answer as compared to 3 or 4. However the sample plots tell a different story. The silhouette permits a visual assessment of the clustering. For example, in the plots corresponding to 3 and 4 clusters, it is seen that the silhouettes are all positive with a few exceptions. The clusters for 3 and 4 are quite balanced, on the other hand for 5 and 6, there are clusters with few points indicating a forcing of clusters. Also the trend is more important than the actual values of the indices. For example, although the average silhouette widths are low for 3,4 and 5 clusters, the quality of clustering for 3 and 4 clusters was significantly better than for 5 or 6 clusters. We ran k-means repeatedly for each number of clusters and compared the consistency of the partitions using the Jaccard index. The partitions showed a high degree of consistency for 3 and 4 clusters (i.e. groups of points tended to be assigned together to the same cluster) but a significantly lower consistency among the partitions in 5 clusters.

4.3.2 Experiments on the yeast shock data by Gasch et al.

Yeast shock data by Gasch et al. [23]

The full dataset consists of gene expression data for 6152 known and putative genes over 140 conditions. We focus on a T=8 time course following a heat shock from 27 to 35C. The 8 time points correspond to 5,10,15,20,30,40,60 and 80 minutes respectively after the heat shock. The values are normalized log-ratios to a baseline obtained pooling equal amounts of all experimental samples.

For our experiments we used log transformed data instead of the original values. The following pre processing steps were performed prior to cluster analysis.

1. **Filtering out genes**

   Filtering out genes is an important pre-processing step. Bryan [9] points out that genome-wide collections of expression trajectories often lack natural clustering structure, prior to ad hoc gene filtering. The gene filters in common use induce a certain circularity to most gene cluster analyses: genes are points in the attribute space, a filter is applied to depopulate certain areas of the space, and then clusters are sought in the ”cleaned” attribute space. As a result, statistical investigations of cluster number and clustering strength are just as much a study of the stringency and nature of the filter as they are of any biological gene clusters.

   We apply the following filtering criterion :

   Genes where there was an expression by a factor of 1.5 over the control condition (i.e. the absolute value of at least $log_2(1.5)$ = 0.585 in the log transformed data) in at least one time point were retained. 4664 genes were retained as a result of this filtering criterion.

As an example, we also applied the following filtering criterion. The coefficient of variation was determined for each gene (row) and the bottom 10 percentile of the genes with the lowest values of the coefficient of variation were discarded from further analysis. The coefficient of variation for gene i was calculated as :

$$c_i = \frac{\sigma_i}{\mu_i}$$

where σ_i was the standard deviation and μ_i was the mean across the 8 time points. 5537 genes were retained as a result of this filtering criterion. The sets of genes obtained by applying the two filtering criteria above were quite different, again emphasizing that the choice of the filter could make a significant difference in the analysis.

2. **Centering and standardization**

   The data corresponding to genes retained after applying the filtering criterion mentioned above was centered. Standardization by row was performed.

Clustering results

We illustrate a few sample results for the yeast shock dataset. All the results shown correspond to the former filtering criterion, namely retaining genes where at least one value was expressed above a certain threshold. The stem plots corresponding to the Davies Bouldin and Hartigan indices are shown in Figure 4.12. Sample silhouette plots are shown in figures 4.13 and 4.14. Applying the rules of thumb described earlier, both the Hartigan and Davies Bouldin index yield the optimal number of clusters as 2 (or at least 2 is a significantly better answer than 3,4,5 etc.). Also the sample silhouette plots indicate 2 clusters as the best answer. For example, in the plot shown in Figure 4.13(a), none of the data points have a negative silhouette width. In Figure

4.13(b) (3 clusters), a small number of data points have a negative silhouette width while this number is much larger in Figure 4.14(a) (4 clusters) and 4.14(b) (5 clusters). Although these plots correspond to a single run, the results were very similar across multiple runs. Thus for this dataset the clustering structure seems to be more pronounced than for the cdc cell cycle dataset.

4.4 Discussion

The results suggest that in real data sets, even though there may be structure in the data, it may not be very clear-cut (cdc cell cycle data for example). Thus choosing the ”correct number” of clusters is not trivial. Cluster validity indices (CVIs) cannot be relied upon exclusively to identify the ”right number” of clusters, knowledge of the data should be used when possible. CVIs are useful in order to assess whether there is a structure in the data and to get a sense of the number of clusters (are there a small number like 3,4 or 5 or a larger number of clusters like 10 or 20). Also similar patterns across different CVIs indicates that there is indeed a structure in the data.

Among the internal cluster validity indices considered, the silhouette is the most general since it does not make any implicit assumptions about the data residing in a Euclidean space. All that is required is a suitable dissimilarity measure between a pair of data points (vectors). The silhouette may be used *as is* to make comparisons between clusters obtained using the various distance measures and/or clustering methods. This could be particularly useful in the case of heterogeneous datasets as discussed below.

Using multiple types of data

In many genomic applications, heterogeneous datasets are typical. Consider for example gene expression data and motif profiles. While the Euclidean distance may be a reasonable choice

in the case of gene expression data, other distance measures such as the Jaccard distance are more appropriate for sequence data such as motif profiles. We may for instance wish to compare the clusters obtained from gene expression data with those obtained from motif profiles using appropriate distance measures and clustering algorithm for each data type. In such cases although any of the CVIs may be used for cluster validation, a CVI which does not incorporate many assumptions is desirable especially if comparisons between different distance/dissimilarity measures and/or different data types are to be made.

As another example, consider for instance a SOM based clustering approach proposed in [30]. This method uses a Self-Organized Map based approach to perform clustering based on more than one data type (referred to as *category*). Different distance measures can be used for each category. The algorithm also permits weighting the data sources, where the weights reflect the relative importance of the similarity with respect to that particular data type. This method is discussed in Section 5.4 in Chapter 5. The reader is referred to [30] for details of the method.

Suppose we performed clustering using gene expression and motif profile data (categories) using the distance measures appropriate for each category (say Euclidean distance for gene expression and Jaccard distance for motif profile). We might want to compare the quality of these clusters with that of clusters obtained using gene expression data alone. In the case of multiple categories, the input to the algorithm is an augmented data vector formed by concatenating the data vectors correspoding to each category. For cluster validation purposes, one option is to pretend that the augmented data vectors reside in a Euclidean space and apply any of the CVIs. On the other hand it would be more appropriate to define a measure of dissimilarity between data points taking into account different distances and weights for each segment of the augmented data vector. In this case the silhouette (or any other method that does not assume data

residing in a Euclidean space) could be applied as is, both to this clustering as well as clustering using gene expression data alone. We could then possibly make direct comparisons between the quality of the clusters in the two cases.

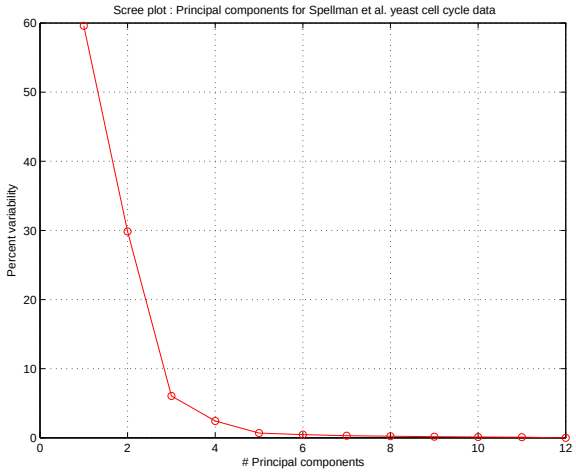
4.5 Conclusions

Cluster validation for genomic data can be quite tricky and involves a number of issues. Cluster Validity Indices (CVIs) although useful, cannot alone be relied upon to determine the optimal clustering that represents the structure in the data as best as possible. Knowledge of data must to be relied on whenever possible. Since clustering is data dependent, the ultimate judgement as to what constitutes a good cluster must be made by the user. At best one could hope to use a CVI which matches the intuition as to what is a good cluster for the task on hand.

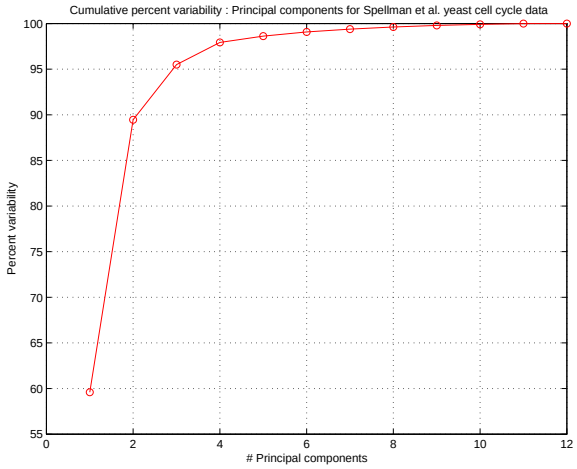
The CVIs could serve as a guideline to assess whether there is a structure in the data, for instance to obtain a rough estimate of the number of clusters (a small number like 2 or 3 or a larger number like 10)? Different CVIs may not agree on the ”right” number of clusters. Consistent patterns across different indices and/or different distance measures are indicative of a stable cluster structure.

Different distance measures may be appropriate for different data types, thus CVIs which do not assume the data to reside in a Euclidean space such as the **silhouette** may be preferable, especially when heterogeneous data are used. This would enable a comparison of clusterings obtained using the different data types. The silhouette also provides a visual assessment of how good the clustering is, thus one need not rely on the values of the index alone for making a judgement on the number of clusters. In any case, further analysis such as stability with respect to perturbation would be preferable in addition to computing cluster validity indices.

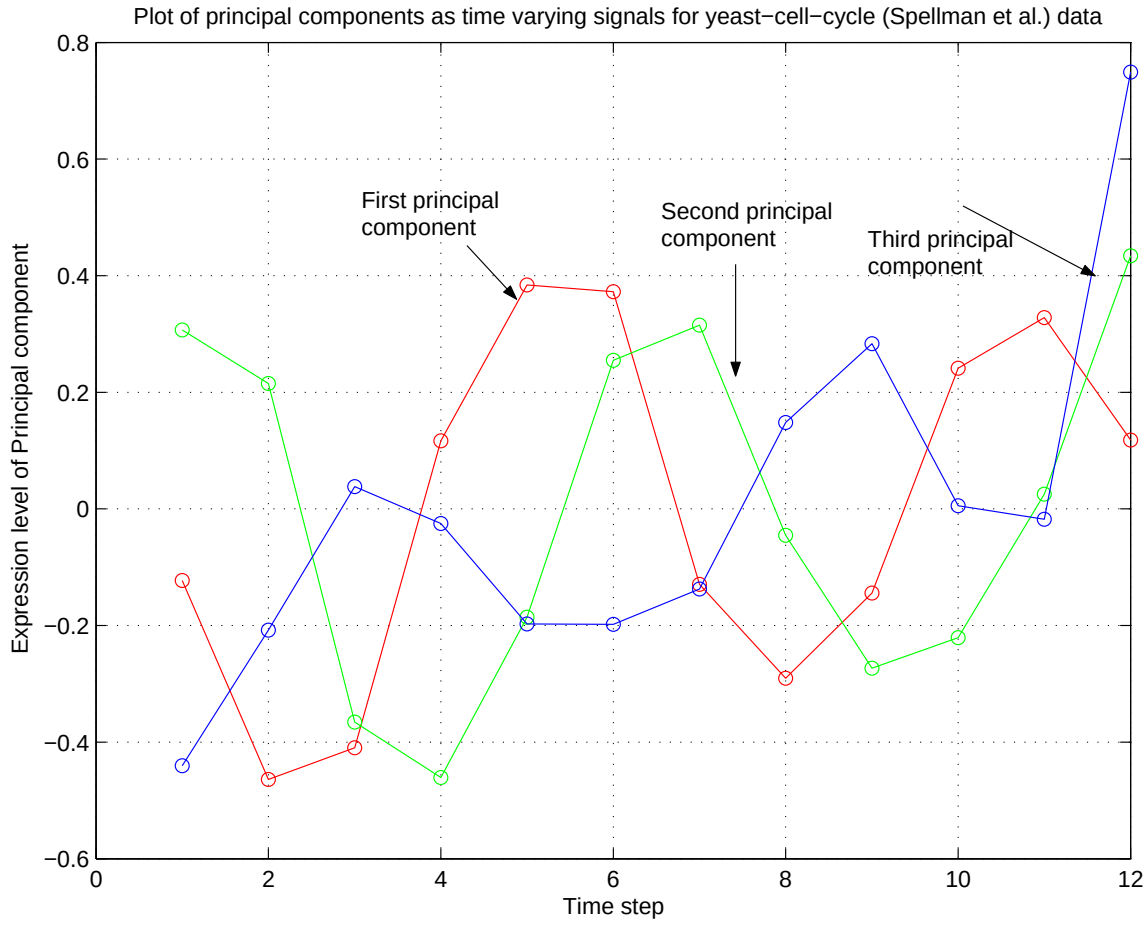
Finally, we point out that genome-wide collections of expression trajectories often lack natural clustering structure, prior to ad hoc gene filtering [9]. Hence filtering of genes and other pre processing could make a significant difference in statistical analysis of genomic data, particularly with respect to the number of clusters and clustering strength. Bryan [9] suggests not looking for consistency from a ”natural clusters” standpoint but instead to seek a ”good segmentation”or try seeded clustering.



(a)



(b)



(c)

Fig. 4.1. Results of Principal Component Analysis on subset of yeast cell-cycle data generated by Spellman et al.
(a)Percent variability corresponding to each PC
(b)Cumulative percent variability corresponding to each PC
(c)First 3 Principal Components

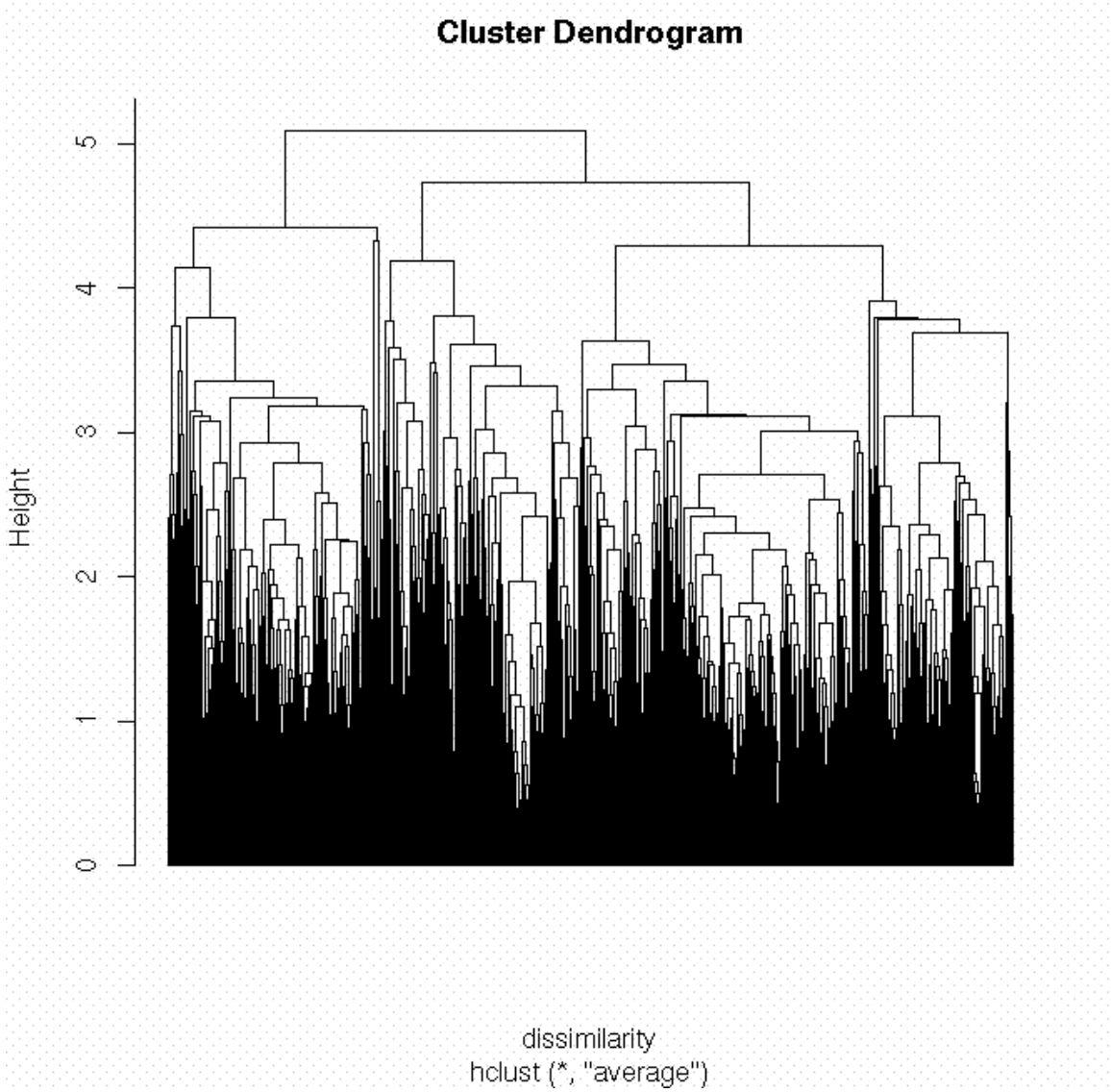


Fig. 4.2. Hierarchical clustering results on subset of yeast cell-cycle data generated by Spellman et al.
Results of hierarchical clustering in 12 dimensions

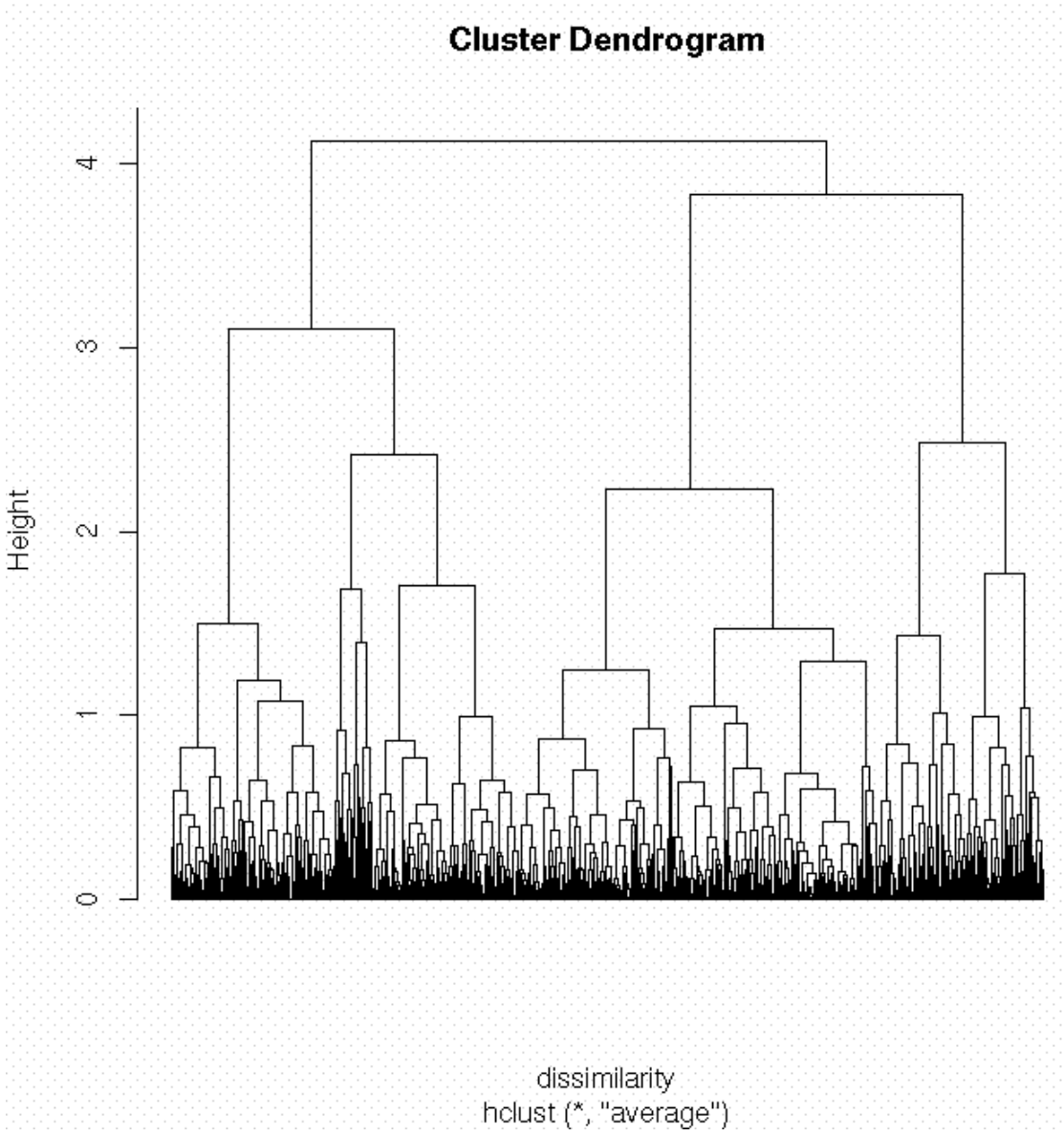
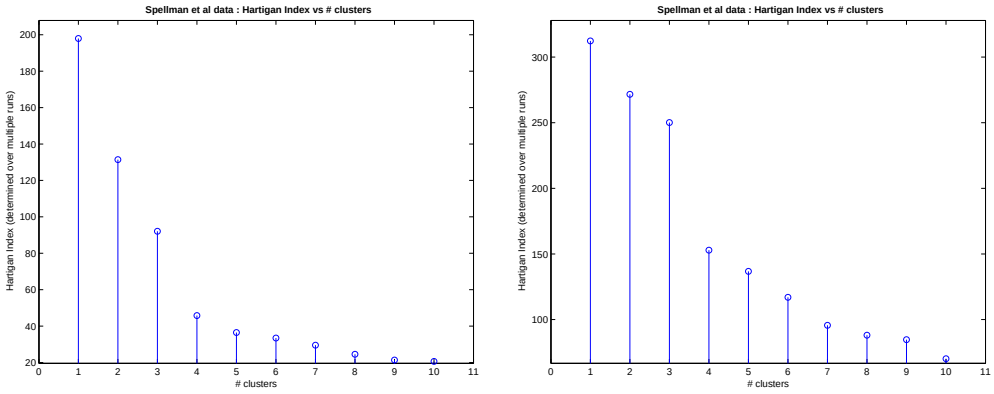
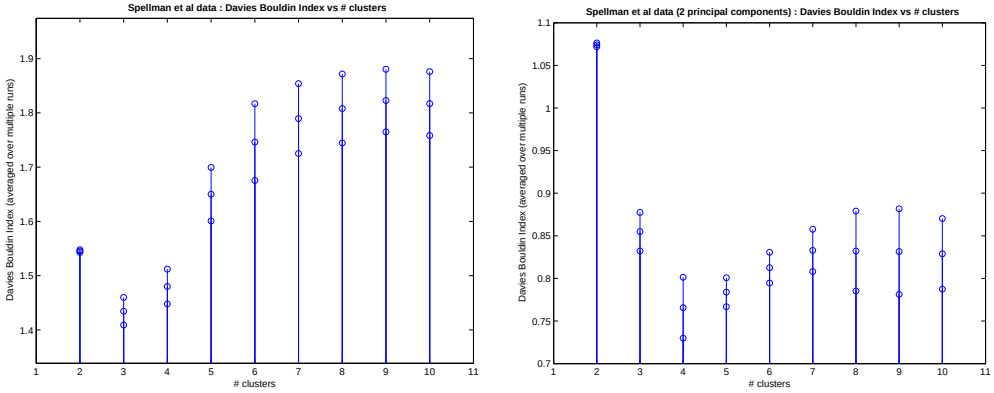


Fig. 4.3. Hierarchical clustering on subset of yeast cell-cycle data generated by Spellman et al. Results of hierarchical clustering in reduced dimensions (first two principal components)



(a) (b)

Fig. 4.4. Spellman et al. data : Hartigan Index for cluster validation
(a) 12 dimensional data (b)Projections in space spanned by first two principal components



(a) (b)

Fig. 4.5. Spellman et al. data : Davies Bouldin Index for cluster validation
(a) 12 dimensional data (b)Projections in space spanned by first two principal components

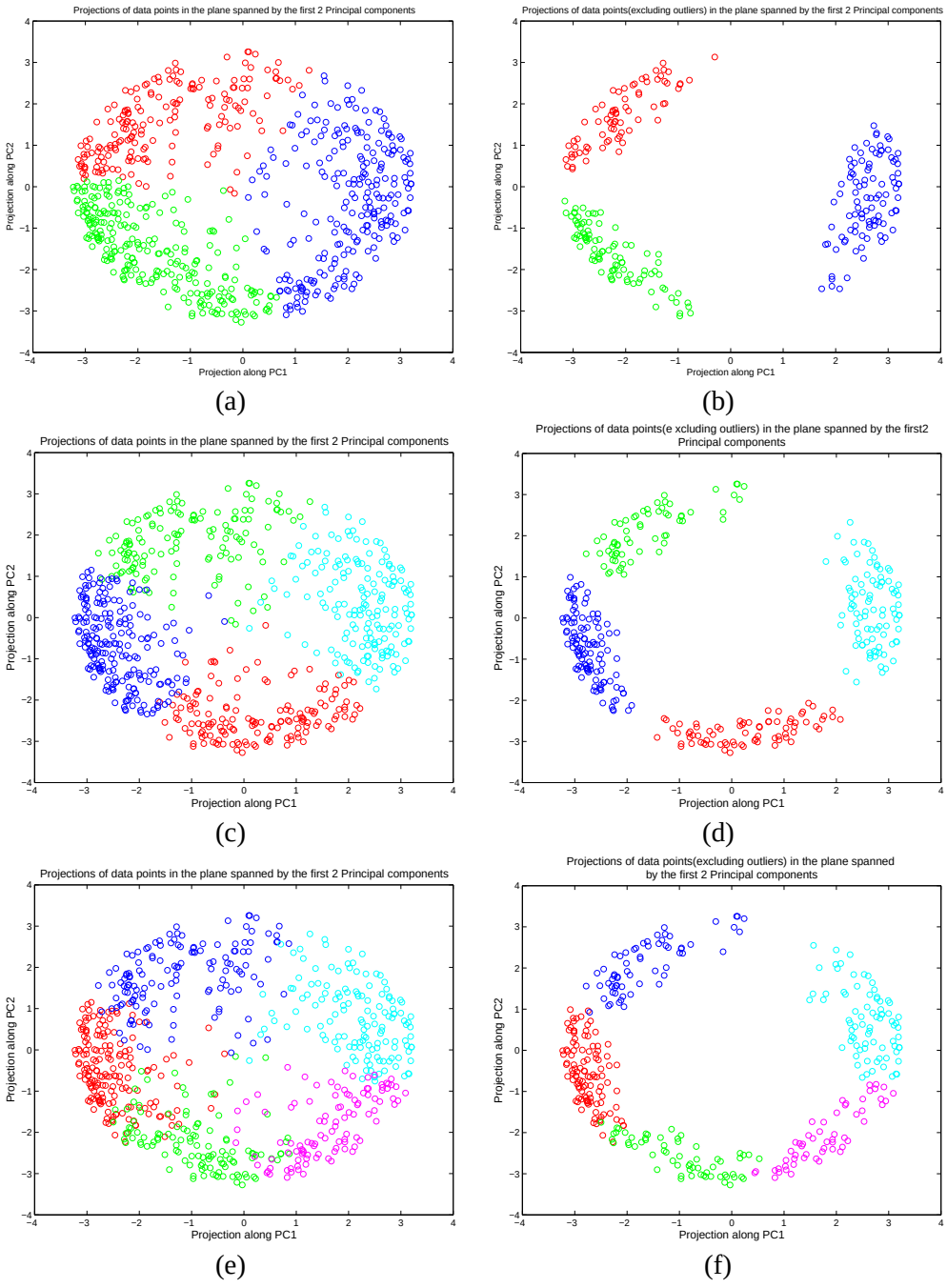


Fig. 4.6. Spellman et al. yeast cell cycle data. Projections in the plane of first two Principal Components
(a) & (b) 3 clusters : (a) All data samples (b) Outliers excluded
(c) & (d) 4 clusters : (c) All data samples (d) Outliers excluded
(e) & (f) 5 clusters : (e) All data samples (f) Outliers excluded

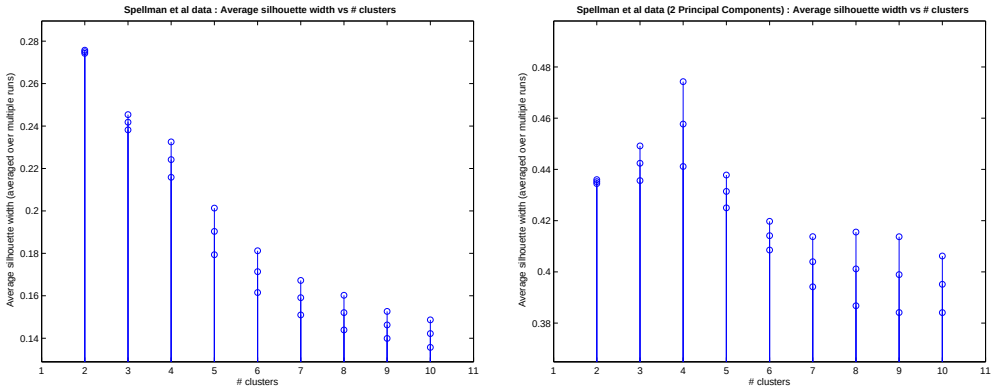
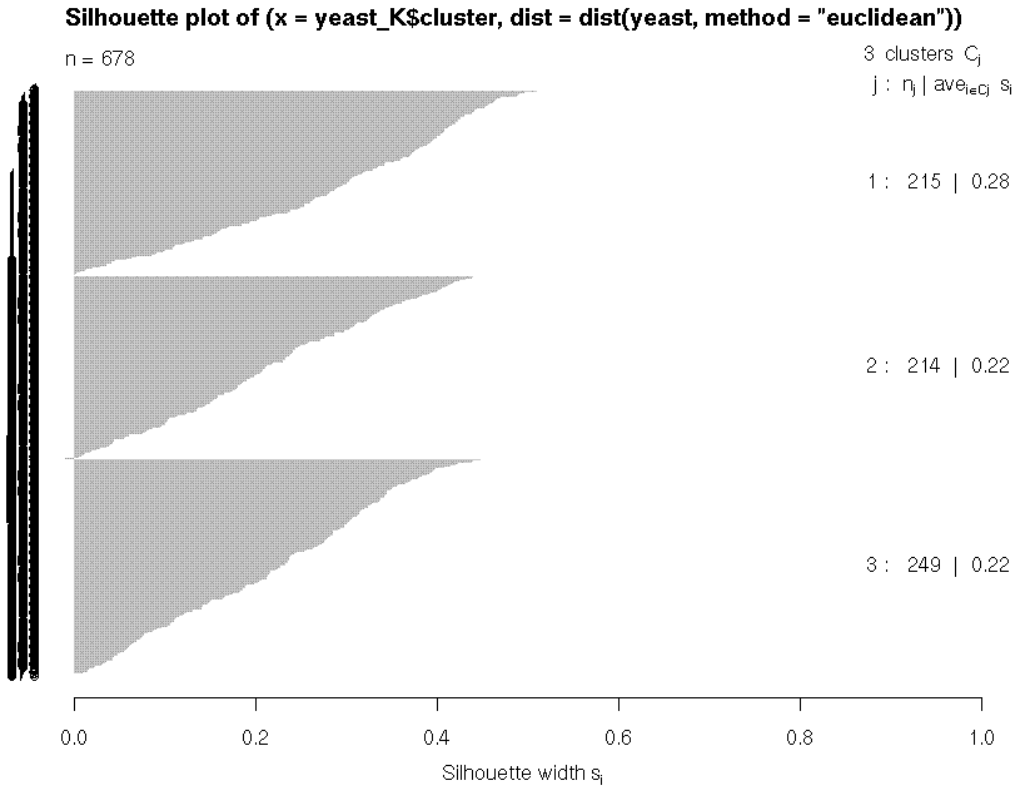
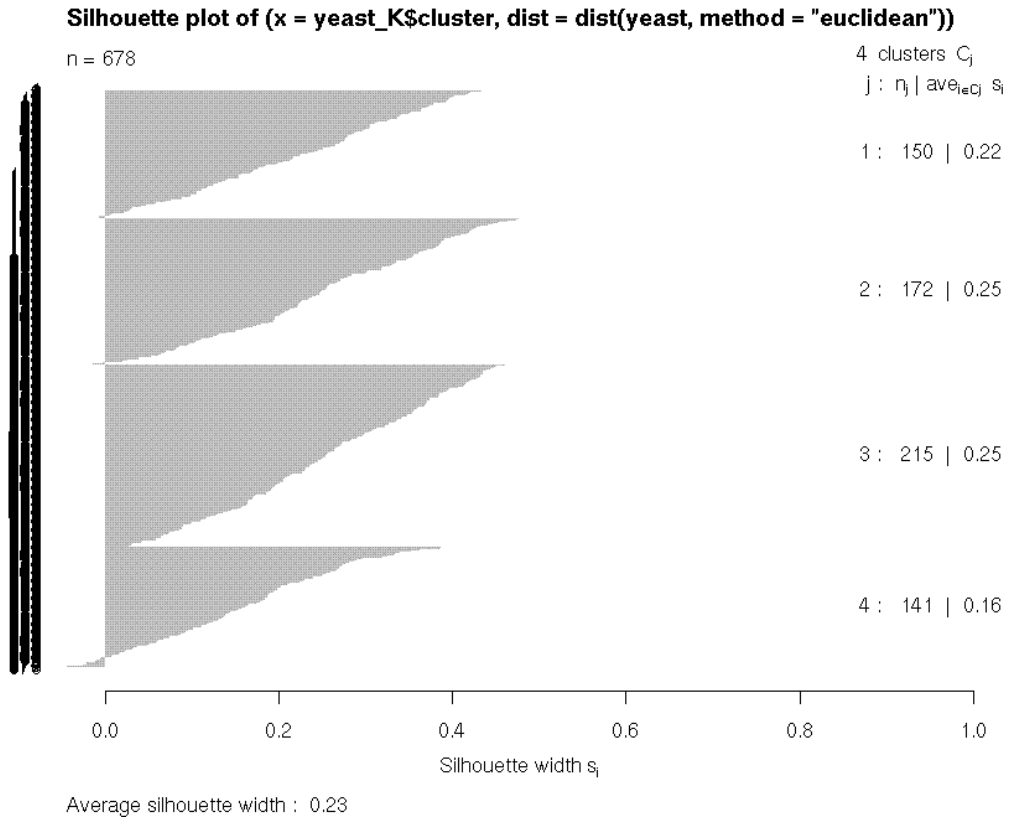


Fig. 4.7. Spellman et al. data : Silhouette Index
(a) 12 dimensional data (b)Projections in space spanned by first two principal components

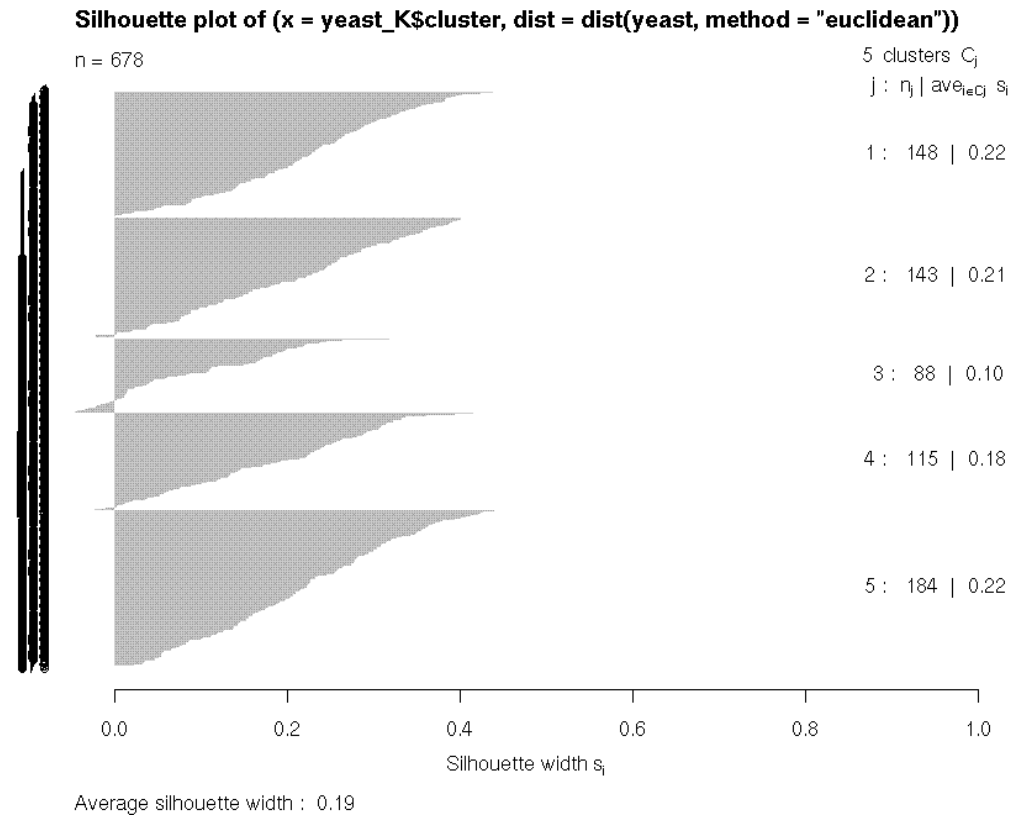


(a)

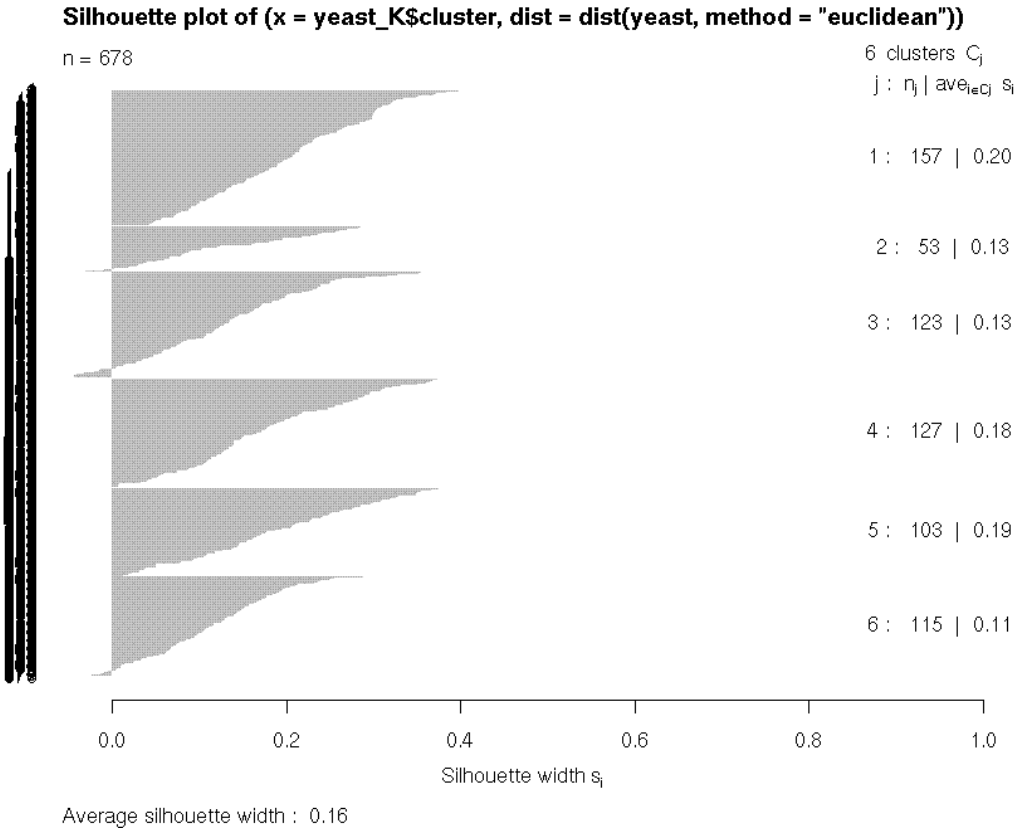


(b)

Fig. 4.8. Spellman et al. data : Clustering using 12 time points) :(a) 3 (b) 4 clusters

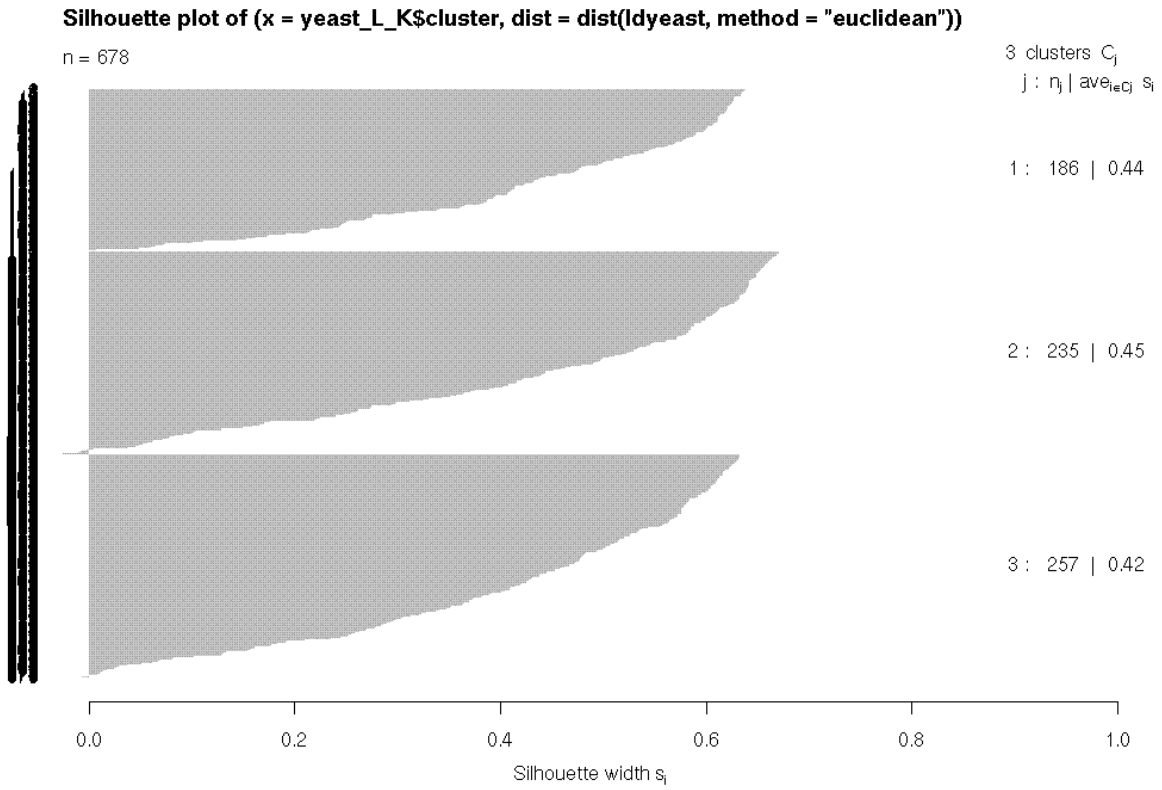


(a)

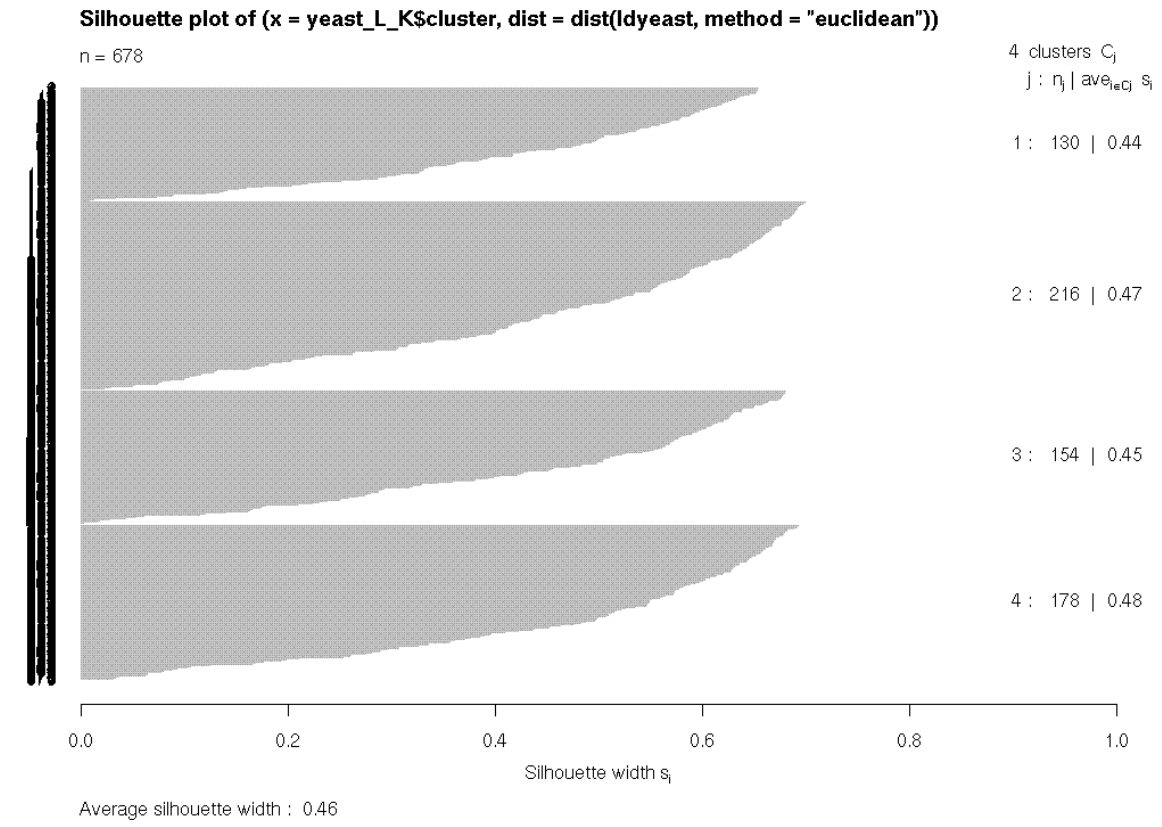


(b)

Fig. 4.9. Spellman et al. data : Clustering using 12 time points) :(a) 5 (b) 6 clusters

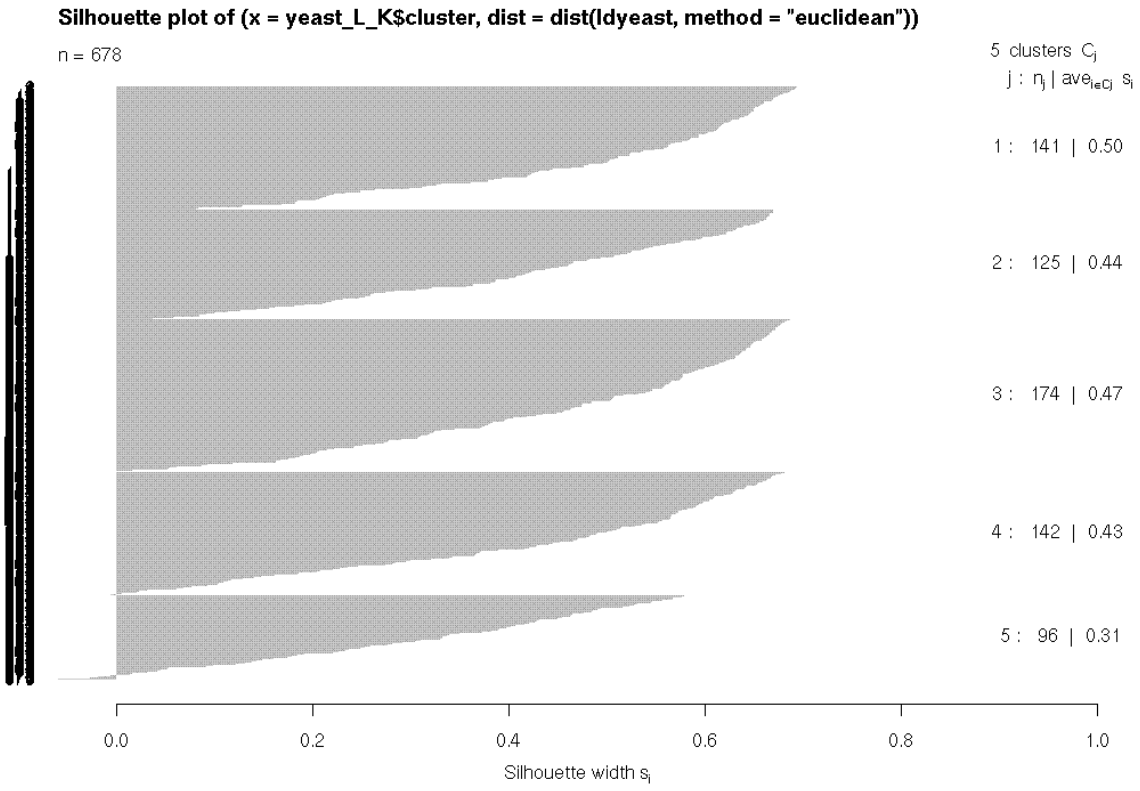


(a)

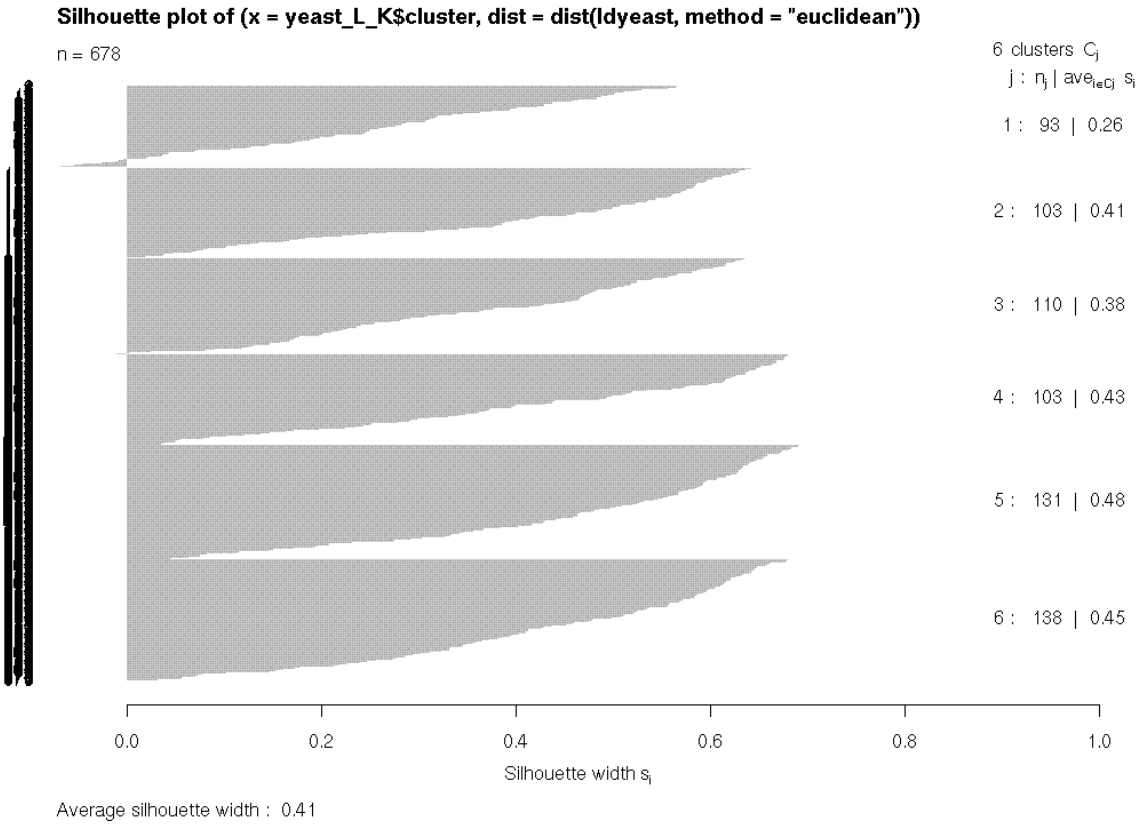


(b)

Fig. 4.10. Spellman et al. data : Clustering in lower dimensions (2 principal components) : Sample silhouette plots for (a) 3 (b) 4 clusters

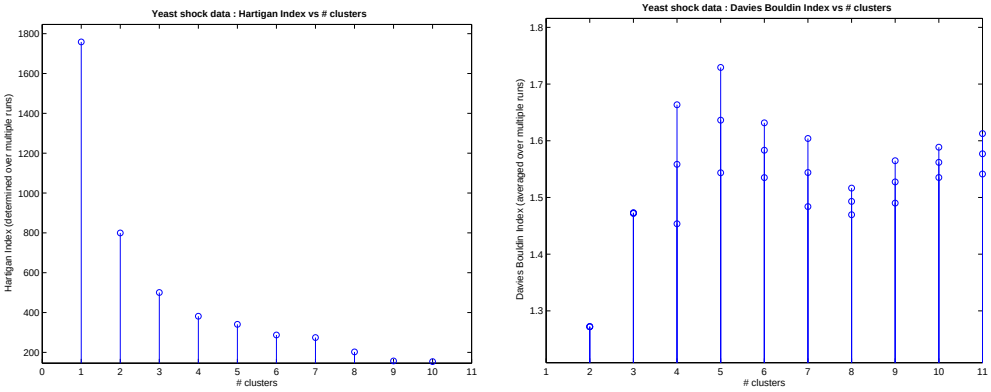


(a)



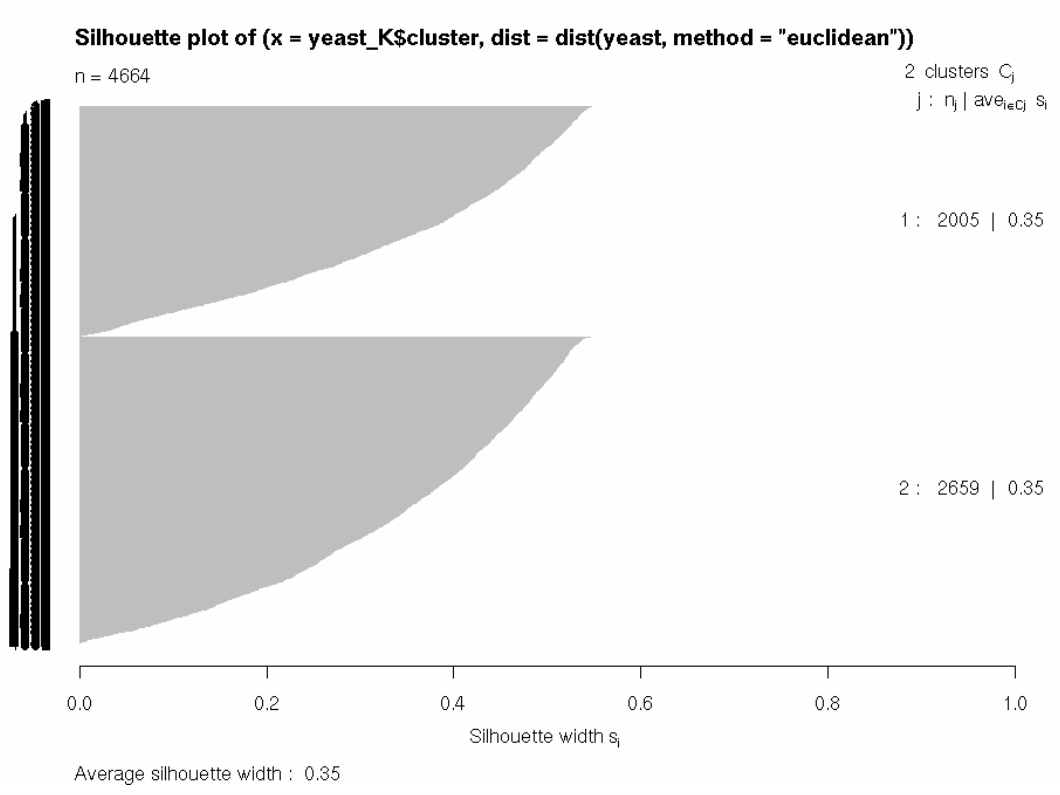
(b)

Fig. 4.11. Spellman et al. data : Clustering in lower dimensions (2 principal components) : Sample silhouette plots for (a) 5 (b) 6 clusters

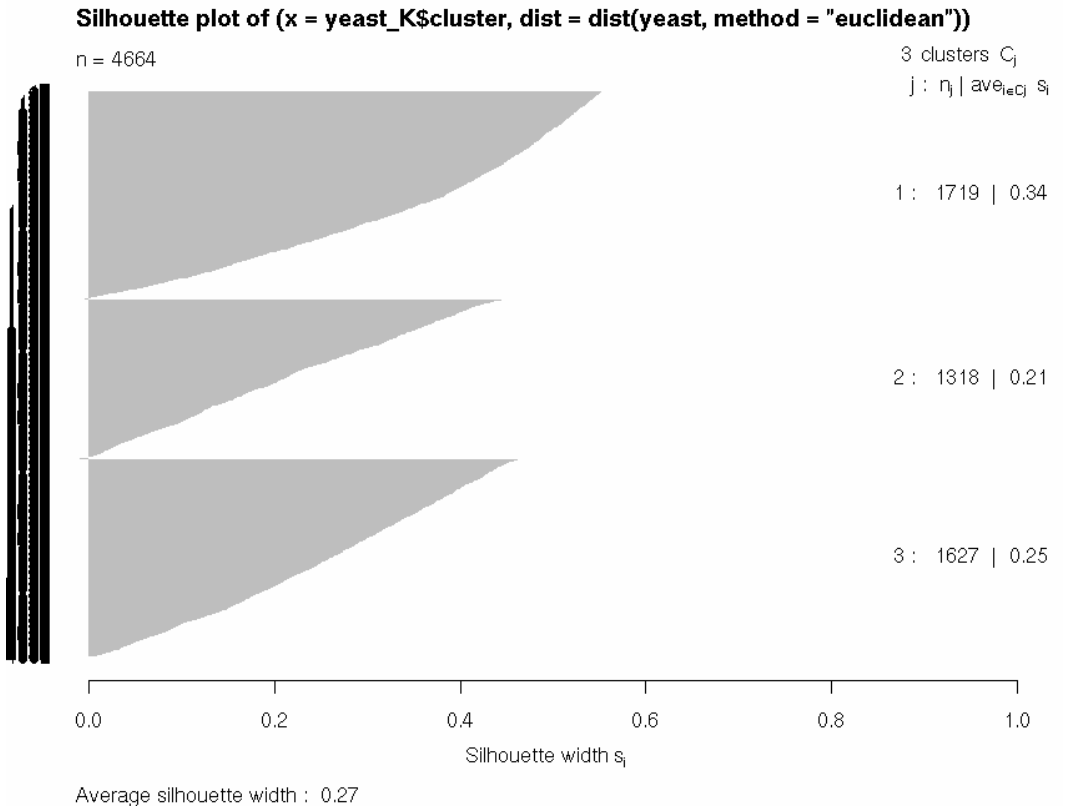


(a) (b)

Fig. 4.12. Yeast shock data (clustering using 8 time points) : Cluster Validity Indices (a) Hartigan Index (b) Davies Bouldin Index

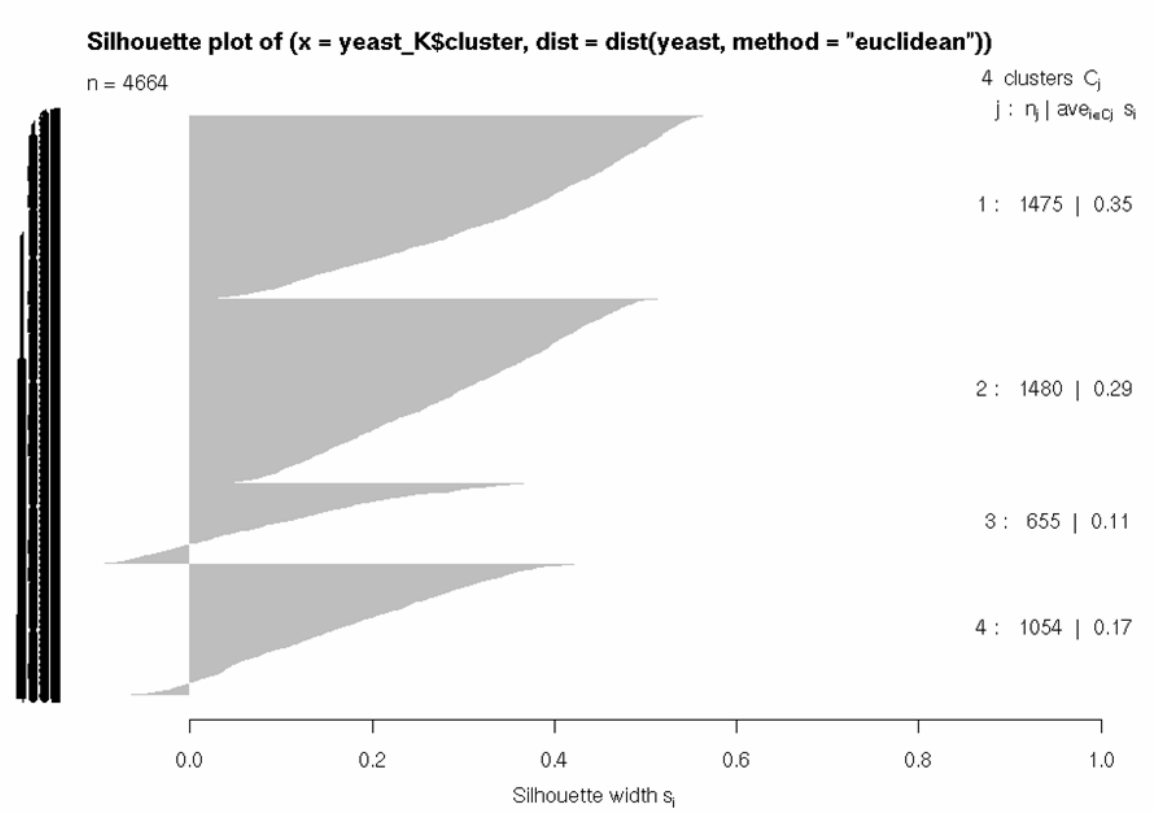


(a)

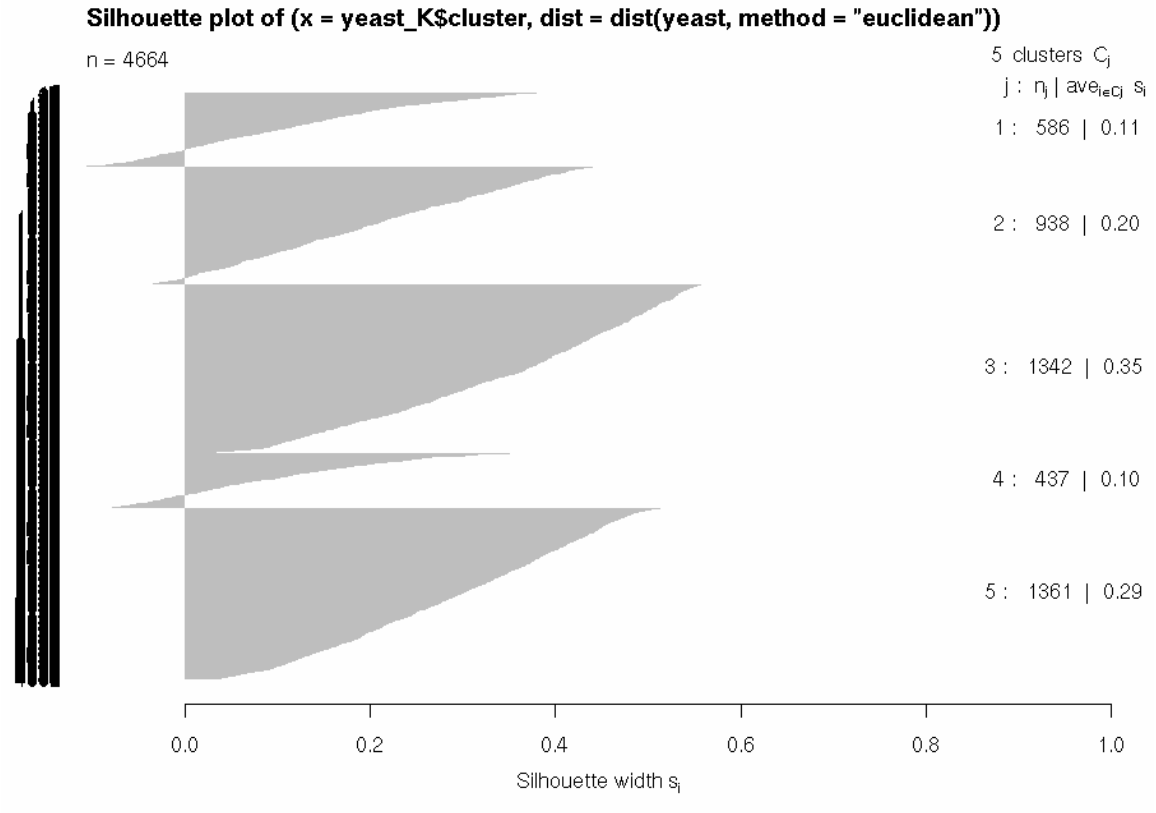


(b)

Fig. 4.13. Yeast shock data : Clustering using 8 time points (in 8 dimensions) : Silhouette plots for (a) 2 (b) 3 clusters



(a)



(b)

Fig. 4.14. Yeast shock data : Clustering using 8 time points (in 8 dimensions) : Silhouette plots for (a) 4 (b) 5 clusters

Chapter 5

External cluster validity indices for genomic data and formulating theoretical problems in cluster validation as flow problems

Summary

In this chapter we focus on the problem of comparing two clusterings in the context of genomic data. As in the previous chapter, we use the term clustering to refer both to the process as well as the result. We use the term *hard clustering* to mean a clustering in which an object (data point) is assigned to one cluster only. In soft clustering, the degree to which an object is associated with a cluster is indicated by a membership grade.

Clustering is often employed as a first step in genomic data analysis in order to identify groups of data samples (often groups of genes) for further analysis. It is often of interest to compare a clustering result with an external ”reference” clustering, for instance comparing a grouping of genes obtained by applying a clustering algorithm on gene expression data against a grouping derived from existing knowledge (based on gene ontology for instance). Although the choice of the reference clustering depends on the task at hand, typically groupings of genes such as those based on gene ontology, are comprised of overlapping clusters. Many of the external cluster validity measures in the literature assume partitions (non-overlapping and exhaustive clusters), hence they may not be suitable in the context described above. We suggest that a Mallows distance based method proposed recently may be more appropriate for cluster validation

of genomic data. We also define a new measure for the distance between two clusterings and discuss when this would be a suitable choice.

Network flow problems are a well studied class of problems with applications in various disciplines constitute a powerful conceptual tool. Many combinatorial problems can be easily formulated as flow problems. The problems of computing the Mallows based distance and the distance measure mentioned above can both be formulated as flow problems. Some related problems can also be formulated as network flow problems, hence we suggest that the flow problem formulation could serve as a useful framework for cluster validation problems.

This chapter is organized as follows. Some of the existing cluster validity indices are briefly reviewed in section 5.1. A brief overview of flow problems is provided in section 5.2. The interested reader is referred to [46, Chapter 26] for a more thorough introduction and [2] for a detailed discussion of flow problems. Results and discussion are presented in section 5.4 followed by conclusions in section 5.5.

Notation

We introduce the notation and definitions used in the rest of the chapter.For the rest of the discussion in this chapter we use the term clustering to refer both to the process as well as the result. A clustering is represented by a membership matrix. Thus if there are N objects grouped into K clusters, the corresponding membership matrix $\mathcal{M}$ is of size $N \times K$. $\mathcal{M}[i, j]$ would represent the degree to which data point i is associated with cluster j. We use $Row(M, i)$ to denote the i^{th} row and $Col(M, j)$ to denote the j^{th} column in matrix M. We use $I(i, J)$ to represent the i^{th} row of an identity matrix of size $J \times J$.

5.1 Brief overview of external cluster validity indices

In this section we briefly discuss some of the external cluster validity indices in the literature and outline some of the common limitations of existing methods. The interested reader may consult one of the many survey papers for a detailed discussion. Meila [41] classifies methods for comparing two clusterings as pair counting, set matching and Variation of information (VI) . Pair counting methods evaluate similarity of two clusterings by examining the number of agreements with respect to how the two clusterings group a pair of data points in the same cluster or separate them into different clusters. The Jaccard Index, Rand Index [48] and the Fowlkes [21] index are a few examples of pair counting methods. Set matching methods seek for matches between clusters of the two clusterings i.e. they look for sets of objects grouped together in the two clusterings. A score is computed in a step-wise manner from these cluster matches. The method proposed in [16] falls into this category. The Variation of Information (VI) [41] is an information theoretic based measure. It measures the amount of information that is lost or gained in changing from one clustering to another.

Despite a large amount of literature on clustering, some of issues in the area of cluster validation, especially for overlapping clusters are not sufficiently addressed. One limitation of many of the external CVIs is that they assume that the clusters are mutually exclusive. This could be a limitation in the context of genomic data, where overlapping (i.e. not mutually exclusive) clusters are typical. In many of the existing external cluster validation methods described above there is no global optimization objective. Also some clusters may not play a role in the matching especially if the number of clusters are different, for example only ”best match” is considered, mismatches are not penalized ([62]). Recently Zhou *et al.* [62] proposed a Mallows

distance based metric, this method has many desirable properties and overcomes many of the shortcomings of previous approaches. This metric may be more suitable for cluster validation in the context of genomic data since it deals with soft clusterings and overlapping clusters. Hence, we pay more attention to this method in this chapter.

Mallows distance [62]

Zhou *et al.* [62] propose a Mallows based distance for comparing two clusterings. This is applicable to soft clusterings. In this method, each cluster in the clusterings is "soft matched" with every cluster in the other clustering. The distance between the clusterings is then the optimal (minimum) weighted sum of each of these matches, where the weights satisfy certain constraints. These weights are obtained from a Linear Program (LP) and hence the optimization over the matching weights is global. This method is briefly described below. The reader is referred to [62] for the details.

Let the number of clusters be J and K in the first and second clusterings respectively. If N is the number of objects, the membership matrices are of sizes $N \times J$ and $N \times K$ respectively. Each cluster is assigned a value reflecting its significance. Let α_j be the significance of cluster j in clustering 1 ($1 \leq j \leq J$) where $\sum_{j=1}^{J} \alpha_j = 1$. Let β_k be the significance of cluster k in clustering 2 ($1 \leq k \leq K$) where $\sum_{k=1}^{K} \beta_k = 1$. A cost matrix C of size $J \times K$ is computed where $C(j,k)$ is the distance between j^{th} cluster of clustering 1 and k^{th} cluster of clustering 2. The Mallows distance between the two clusterings (represented by matrices M_1 and M_2) is given by the Linear Program in (5.1).

$$D(M_1, M_2) = \min_{w_{j,k}} \sum_{j=1}^{J} \sum_{k=1}^{K} w_{j,k} C(j,k) \tag{5.1}$$

subject to

$$w_{j,k} \geq 0 \quad \forall j,k$$

$$\sum_{k=1}^{K} w_{j,k} = \alpha_j$$

$$\sum_{j=1}^{J} w_{j,k} = \beta_k$$

5.2 Brief overview of flow problems

Flow problems find applications in various disciplines. Flow networks can be used to model fluids flowing through pipes, currents in electrical circuits and so on. A flow network can be thought of as directed graph with two special vertices, a **source** and a **sink**. Each edge can be thought of as a conduit for the material which originates from the source and courses through to the sink. In general there could be multiple sources, sinks and more than one commodity flowing through the network. Cormen *et al.* [46, Chapter 26] provide a graph theoretic definition of flow networks. Consider a directed graph $G = (V, E)$ consisting of two distinguished vertices a source (r) and a sink (s). Each edge $(u, v) \in E$ has a non-negative capacity $c(u, v) \geq 0$

A flow is a real-valued function $f : V \times V \rightarrow R$ that satisfies the following three properties.

- **Capacity constraints :**

  For every $u, v \in V$, $f(u, v) \leq c(u, v)$.

- **Skew symmetry :**

  For all $u, v \in V$, $f(u, v) = -f(u, v)$

- **Flow conservation :**

  For all $u \in V - \{r, s\}$,

$$\sum_{v \in V} f(u, v) = 0$$

  This may informally be stated as inflow equals outflow at all the nodes except the source and sink.

Each edge has the following associated values.

- **Cost** : The cost per unit flow on that particular edge
- **Capacity** : The maximum amount of flow through that edge
- **Lower bound** : The minimum amount of flow through that edge. (the default lower bound is 0 due to the non-negativity of flows)

In the formulation of some of our flow problems we make use of the **Integrality Theorem** [46, Chapter 26]. This theorem states that if all the capacities are integers, there exists a maximum flow with integer values.

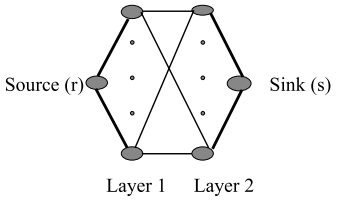
5.3 Description of the flow problems

We use standard min-cost max-flow (see Ahuja *et al.* [2] for more on this) in appropriately generated network. For an arc (i, j) in the network, $c(i, j)$ denotes the cost of sending a unit flow through (i, j). Upper and lower bounds on capacities on the arc are represented by $u(i, j)$ and $l(i, j)$ respectively.

5.3.1 Minimum cost matching distance

DEFINITION 5. *Let M_1 and M_2 be two membership matrices of the sizes $N \times J$ and $N \times K$ respectively. We define the distance between the matrices $D(M_1, M_2)$ as the cost of the min-cost maximum matching in a bipartite graph where there are J and K nodes on the two sides respectively and the cost of edge connecting node i to node j is the distance between the i^{th} column of M_1 and j^{th} column of M_2.*

If $J = K$, this reduces to perfect matching and is known as the Hungarian assignment problem. This situation is shown in Figure 5.1. Let $d(i, j)$ be the distance between i^{th} column of M_1 and j^{th} column of M_2. ($d(i, j)$ can be considered as the cost of converting i^{th} column of M_1 into j^{th} column on M_2.) In general, any distance metric can be used; but we are interested in the case when L_1 norm is used and when M_1 and M_2 are boolean matrices. The optimal (minimum) cost $D(M_1, M_2)$ is the minimum number of bit flips needed for transforming M_1 into M_2, within column permutations. In terms of the original problem, this is the minimum number of reassignments of objects to clusters in order to convert one clustering to another. The Integrality Theorem [46, Chapter 26] guarantees that there exists a maximum flow with integer values which achieves the optimal (minimum) cost. The integer flow solution specifies the optimal perfect matching solution.



If $J \neq K$, we define the distance to be $\min_{\tilde{M}_1} \{D(\tilde{M}_1, M_2)\}$ where $\tilde{M}_1$ is a sub-matrix of M_1 with K columns retained. This situation is shown in Figure 5.2. The corresponding flow network is depicted in Figure 5.3.

Again, let $d(i,j)$ be the distance between i^{th} column of M_1 and j^{th} column of M_2. The computation of the distance measure (min-cost maximum matching) can be represented as a min-cost max-flow problem over network G with:

- Layer 0: r, the source,
- Layer 1: C_1, the set of J clusters in M_1,
- Layer 2: C_2, the set of K clusters in M_2,
- Layer 3: s, the sink.

The costs, capacities with $i \in C_1, j \in C_2$ are:

$$\begin{aligned} (r,i): &\quad c(r,i)=0, &\quad u(r,i)=1 \\ (i,j): &\quad c(i,j)=d(i,j), &\quad u(i,j)=\infty \\ (j,s): &\quad c(j,s)=0, &\quad u(j,s)=1 \end{aligned}$$

5.3.2 Formulating the Mallows distance as a flow problem

The computation of the Mallows distance [62] described earlier may also be formulated as a flow problem as shown in Figure 5.4. The J layer 1 nodes correspond to the J clusters in the first clustering and the K layer 2 nodes correspond to K clusters in the second clustering. The significances α_j ($1 \leq j \leq J$) are the capacities of the edges connecting the source to the J layer 1 nodes. Similarly, the significances β_k ($1 \leq k \leq K$) are the capacities of the edges connecting

the K layer 2 nodes to the sink. The requirement $\sum_{j=1}^{J} \alpha_j = \sum_{k=1}^{K} \beta_k = 1$ ensures flow conservation.

In the special case when $J = K$ and $\alpha_j = 1/K, \quad \beta_j = 1/K, \quad j = 1, \ldots, K$, the solution to the Linear Program yields an optimal one-to-one assignment i.e. it reduces to the Hungarian Assignment Problem.

5.3.3 Computing the lower bound of optimal matching distance between a fixed matrix and another matrix of same size

We now consider the problem of computing the lower bound of distance between a fixed boolean membership matrix and all possible matrices of same size (representing mutually exclusive clusters) under the distance $D(,)$ defined in section 5.3.1. A boolean membership matrix $\mathcal{M}$ of size $N \times K$ is said to be in *Standard-form* if for every $1 \leq i \leq N, 1 \leq j \leq K$, we have $\sum_j \mathcal{M}[i,j] = 1$ and $\sum_i \mathcal{M}[i,j] \geq 1$.

DEFINITION 6. *Let M_1 be a fixed boolean membership matrix of size $N \times K$ and M_2 be any boolean membership matrix of size $N \times K$ in Standard-form define the minimum distance from M_1 as $D_{min}(M_1) = min_{M_2}\{D(M_1, M_2)\}$.*

The brute force approach would be to enumerate all possible matrices M_2 of size $N \times K$ in standard form and then compute the distance $D(M_1, M_2)$ for every M_2 as described in Section 5.3.1. This would be infeasible even for reasonably small values of N and K since the number of such possible matrices is combinatorial. This problem too can however be cast as a flow problem. To compute this lower bound we again construct a flow network G' as shown in Figure 5.5 with:

- Layer 0: r, the source,
- Layer 1: O, the set of N objects,
- Layer 2: C, the set of K clusters,
- Layer 3: s, the sink.

The costs, capacities and lower bounds with $i \in O, j \in C$ are:

$$\begin{array}{lll} (r,i): & c(r,i)=0, & u(r,i)=1, l(r,i)=0 \\ (i,j): & c(i,j)=H(Row(M_1,i),I(j,K)), & u(i,j)=\infty, l(i,j)=0 \\ (j,s): & c(j,s)=0, & u(j,s)=N, l(j,s)=1 \end{array}$$

where $H(\mathbf{u},\mathbf{v})$ is the Hamming distance between vectors $\mathbf{u}$ and $\mathbf{v}$. The cost on edges (i,j) can be seen as the cost of assigning object i to j^{th} cluster. The min-cut, therefore max-flow in G' is N. The lower bound on the edges (j,s) ensures that no cluster is empty. Solving min-cost max-flow on G' give us minimum bit transformations. Again, since all the capacities are integers, the integrality property guarantees the existence of an integer flow solution that achieves the optimal cost.

An important difference between the flow problem discussed in Section 5.3.1 and that shown in Figure 5.5 is that in the problem discussed in the previous section the cost on the edges is the distance between the corresponding *columns* of the respective matrices whereas in the problem discussed in this section (Figure 5.5) the cost on the edges is the (Hamming) distance between the corresponding *rows* of the matrices M_1 and M_2. In the former (Section 5.3.1,

Figure 5.3), the cost on edge (u, v) is the distance between u^{th} cluster in the first clustering and the v^{th} cluster in the second clustering. In the latter the cost on edge (u, v) is the cost of transforming the current cluster assignment of object u (given by u^{th} row of M_1) to assiging object u exclusively to cluster v.

5.4 Results and discussion

In this section we present preliminary results. We first illustrate how the Mallows distance could be used for comparing gene clusters. Next, we compare the minimum cost matching distance with the Mallows distance and suggest possible scenarios where the former may be more suitable.

5.4.1 Comparing gene clusters obtained from clustering with functional groupings from the MIPS database, using the Mallows distance

Many studies (eg. [52] [28]) aim to group genes based on more than one data modality. These studies indicate that including other kinds of information sources such as sequences, ontologies along with gene expression data could lead to better identification of genes with similar functionality.

We used a subset of 678 genes (those with no missing values) from the cdc cell cycle data set generated by Spellman *et al.* [57] as described in Section 4.3.1 of Chapter 4. Multiple sets of 500 genes were obtained by sampling with replacement. The following procedure was repeated for each of these sets, this was done in order to assess the statistical significance of the result.

- Obtain the functional distribution of the above gene set provided by the MIPS CYGD database [1]. Retain only those clusters which have a low p-value. This comprises the *reference grouping*.

- Cluster the genes in the current set based on similarity of gene expression profile only. Compare this clustering with the reference grouping using the Mallows distance and note the value.

- Cluster the genes based on both gene expression as well as motif profiles using the approach proposed by Kasturi *et al.* [30]. This method is briefly described here, the interested reader is referred to [30] for the details. Compare the clustering thus obtained with the reference grouping using the Mallows distance and note the value.

Figure 5.6 plots the score for clustering based on gene expression data only as well as gene expression and sequence data for repeated runs. We normalized the Mallows distance by the number of data points. It was observed that in every run the integrated clusters had a better score (lower distance between the integrated clusters and the MIPS clusters as compared to that between expression clusters and MIPS clusters) as shown in Figure 5.6. This suggests that including multiple data types could indeed produce more functionally relevant clusters.

Description of the SOM based clustering approach proposed by Kasturi *et al.* [30] The method uses a Self-Organizing Map [34] based clustering algorithm to identify clusters of genes simultaneously based on their similarity in multiple datasets (eg. expression as well as motif profiles). Each dataset used is considered a category. Let $C_1, C_2, \ldots, C_m$ denote the m categories. Let $d_1, d_2, \ldots, d_m$ denote the distance measures corresponding to each of the categories respectively. Let the feature vectors for data sample i corresponding to these categories

be $f_1^{(i)}, \ldots, f_m^{(i)}$ respectively. The algorithm uses the augmented feature vector formed by concatenating the individual feature vectors as the input. However each category is treated independently, hence issues of normalization do not arise. The corresponding input for sample i is $< f_1^{(i)} f_1^{(i)} \ldots f_m^{(i)} >$. If N_1, N_2, ..., N_m are the sizes of the feature vectors of the individual categories, the size of the augmented feature vector is $N_1 + N_2 + \ldots + N_m$.

The algorithm is capable of weighting the data categories, where the weights indicate the relative importance of the category (for instance if similarity with respect to motif profiles is more important than similarity with respect to gene expression profile). This would produce clusters with greater similarity of genes within one data source when compared to the other data sources. The algorithm uses an iterative procedure, at each iteration step a category is randomly selected based on the weighting scheme. The chosen category r and its associated distance function d_r are used to train the network of neurons and the weights for the entire input tuple (of dimension $N_1 + N_2 + \ldots + N_m$) are updated using the Kohonen learning rule [34]. The distances are calculated on each segment of the input vector independently using the appropriate distance.

5.4.2 Comparison of Mallows based distance and minimum cost matching distance

For our experiments, we chose a subset of genes, from the cdc cell cycle data by Spellman *et al.* [57]. Those genes with known Transcription Factors according to the SCPD database were chosen, there were about 50 such genes. K-means was used to perform clustering and internal CVIs revealed 3 - 5 clusters in the data. As in the previous experiment, the functional distribution of the above gene set provided by the MIPS CYGD database [1] was chosen as the *reference grouping*. Only those clusters which had a low p-value were retained.

The MIPS database tends to yield a large number of clusters grouped according to functions. For a given set of genes, these individual clusters are often overlapping or even duplicates, often because the database retrieves genes corresponding to a specific function (eg. DNA synthesis and replication) and all its parent functions (eg. DNA processing). Clustering methods producing mutually exclusive clusters yield a smaller number of clusters. Thus a comparison method such as the Mallows distance proposed by Zhou *et al.* [62] where every cluster is "soft-matched" with every other cluster in the other clustering is likely to be sensitive to duplicate clusters. The reference matrix obtained from MIPS consisted of one instance of 2 duplicates of a column. Table 5.1 shows sample results of comparing the MIPS grouping with clusterings using the Mallows and our distances, w. r.t. the duplicate columns. We conducted 200 runs of K-means algorithm with different initializations and for each run, compute our distance $D(,)$ and the Mallows distance. In the table only the results corresponding to the maximum and minimum values are shown. It is seen that our distance is robust to duplicates (the entries in a column are unchanged) and is more likely to find well matched clusters from among the reference clusters. Hence, this measure may be a suitable choice when comparing two clusterings on genomic data.

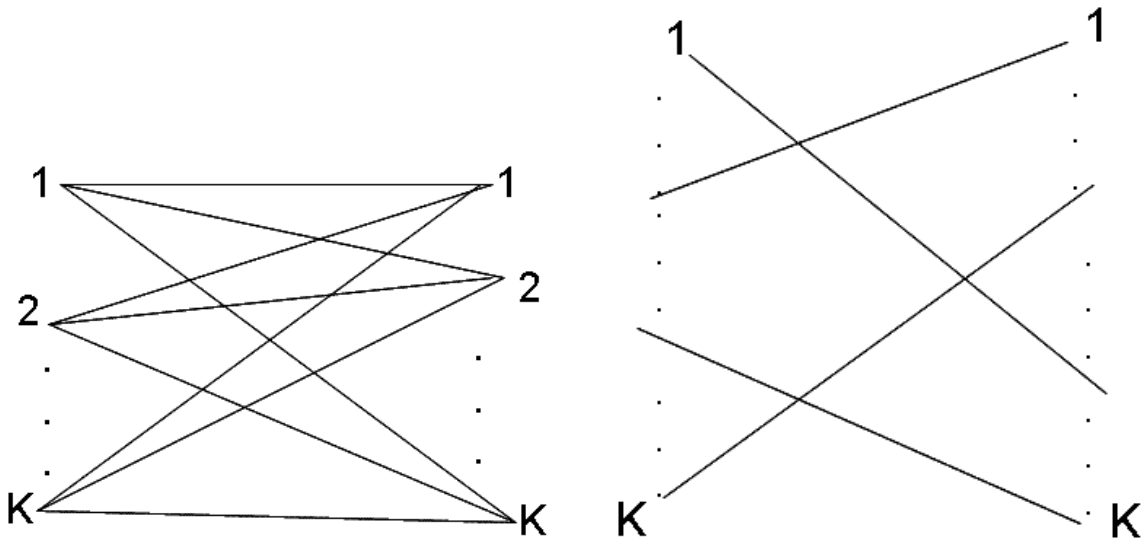
# duplicates	3 clusters				4 clusters				5 clusters			
removed	$D(,)$		Mallows		$D(,)$		Mallows		$D(,)$		Mallows	
	min	max	min	max	min	max	min	max	min	max	min	max
0	36	40	14.04	15.21	34	46	10.47	13.07	34	48	9.08	11.64
1	36	40	14.04	15.20	34	46	10.54	12.96	34	48	9.14	11.59
2	36	40	14.10	15.26	34	46	10.75	12.85	34	48	9.22	11.68

Table 5.1. Comparison of optimal matching cost and Mallows distances with respect to duplicate clusters

5.5 Conclusions

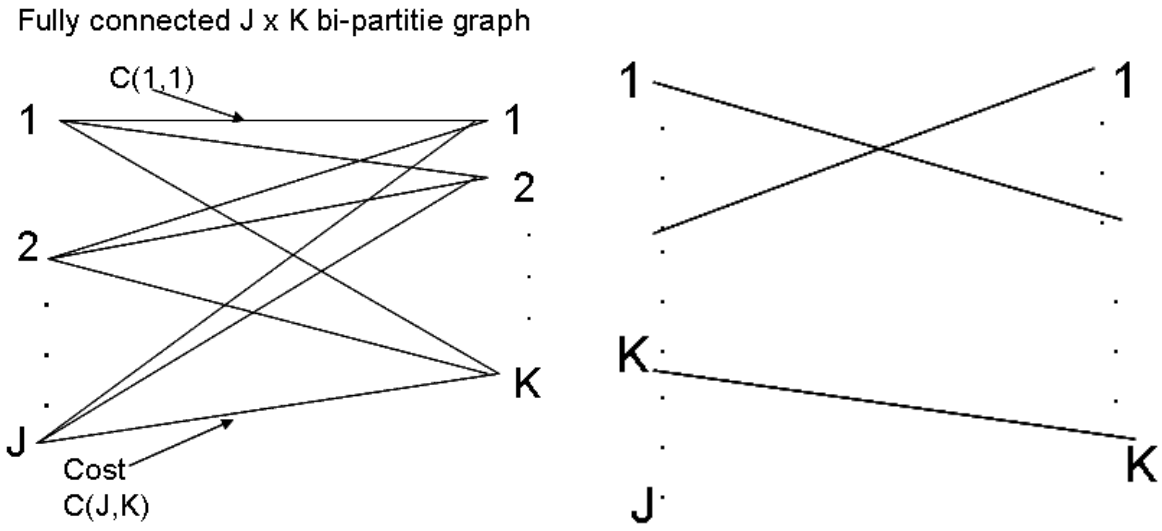
In this chapter we addressed the problem of comparing clusterings in the context of genomic data analysis. We focussed particularly on comparing a clustering obtained by applying a clustering algorithm with a clustering derived from existing biological knowledge (a grouping of genes based on function for instance). Since functional grouping of genes typically consist of overlapping clusters, we suggest that methods which do not assume the clusterings to be partitions are preferable. One example of such a method from the recent literature is the Mallows based distance proposed by Zhou *et al.* [62]. We propose a minimum cost optimal matching distance which is also suitable for overlapping clusters.

Flow problems are a well studied class of problems and constitute a powerful conceptual tool. We formulate theoretical problems in cluster validation which are relevant in the context of genomic data, as flow problems. Both the Mallows based distance and the minimum cost matching distances may be formulated as flow problems. When the number of clusters (columns in the cluster membership matrices) are the same, both these methods reduce to the Hungarian Assignment problem. Recognizing that in genomic data it is typical to have a reference grouping consisting of overlapping clusters, while a number of clustering algorithms produce mutually exclusive clusters, we consider the problem of computing the lower bound of distance between a fixed membership matrix and all possible matrices of same size (representing mutually exclusive clusters) under the optimal matching distance. This could serve as a baseline when evaluating mutually exclusive groupings produced by clustering algorithms. We suggest that the flow problem formulation could serve as a useful framework for cluster validation problems.



(a) (b)

Fig. 5.1. Computing min-cost matching distance when number of clusters are same
(a) $K \times K$ fully connected bi-partite graph
(b)Min cost perfect matching



(a) (b)

Fig. 5.2. Computing min-cost matching distance when number of clusters are different
(a) $J \times K$ fully connected bi-partite graph
(b)Min cost matching

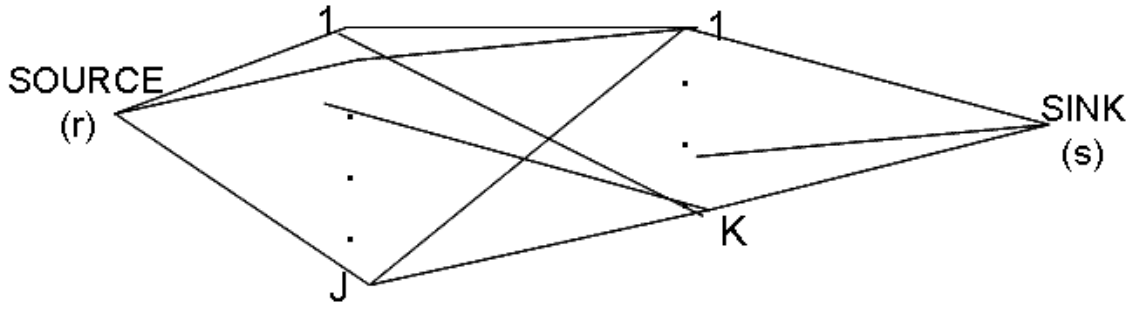


Fig. 5.3. Flow network for computing min-cost matching distance when number of clusters are unequal

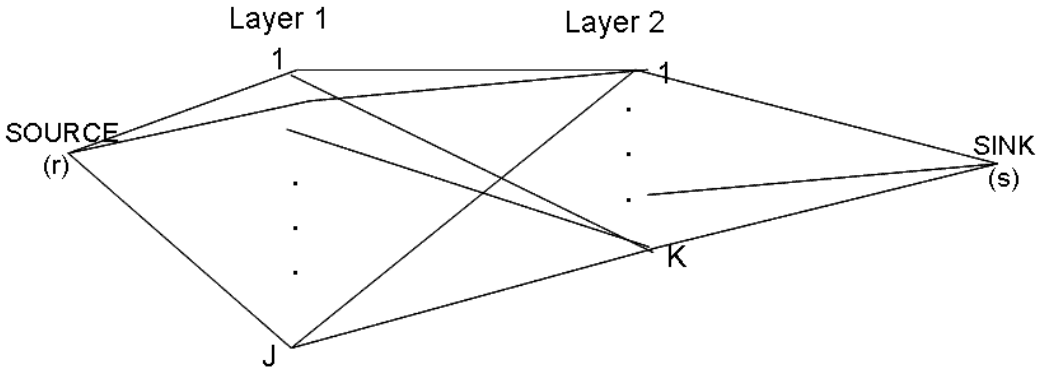


Fig. 5.4. Flow network for computing Mallows distance [62]

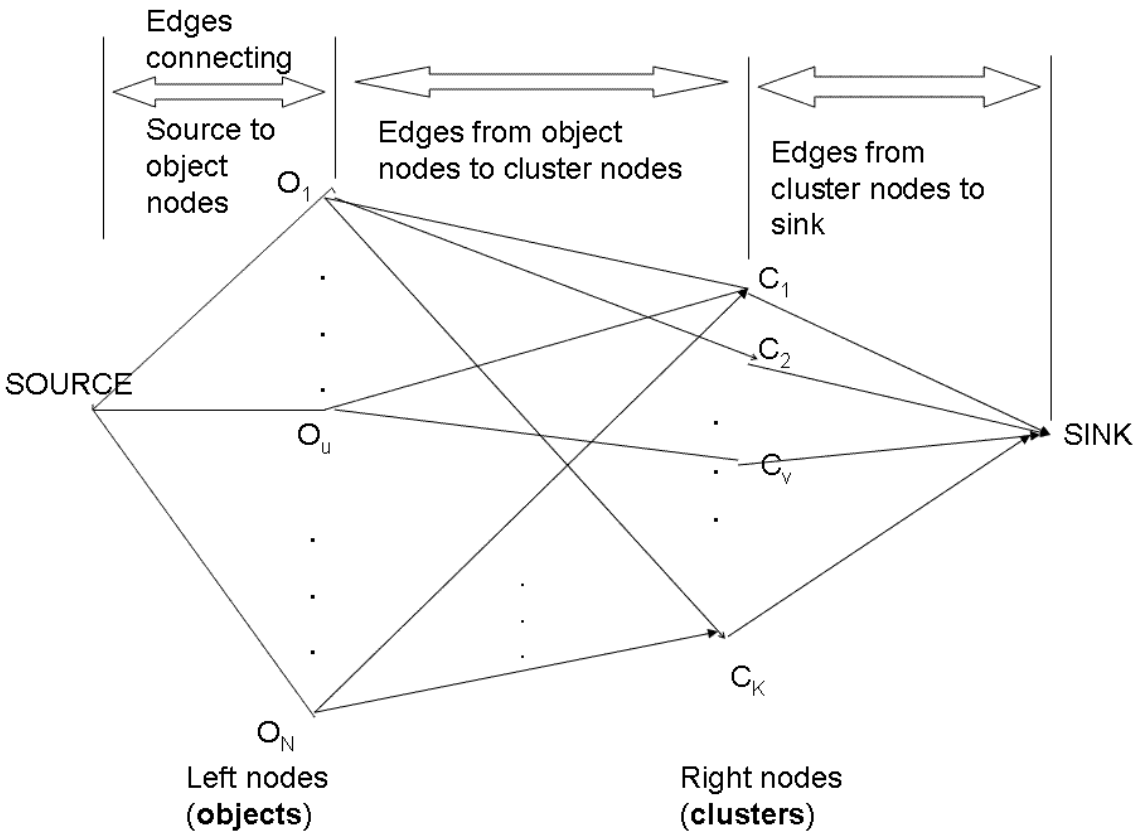


Fig. 5.5. Flow network for computing min-cost matching distance $D_{min}(M_1) = \min_{M_2}\{D(M_1, M_2)\}$

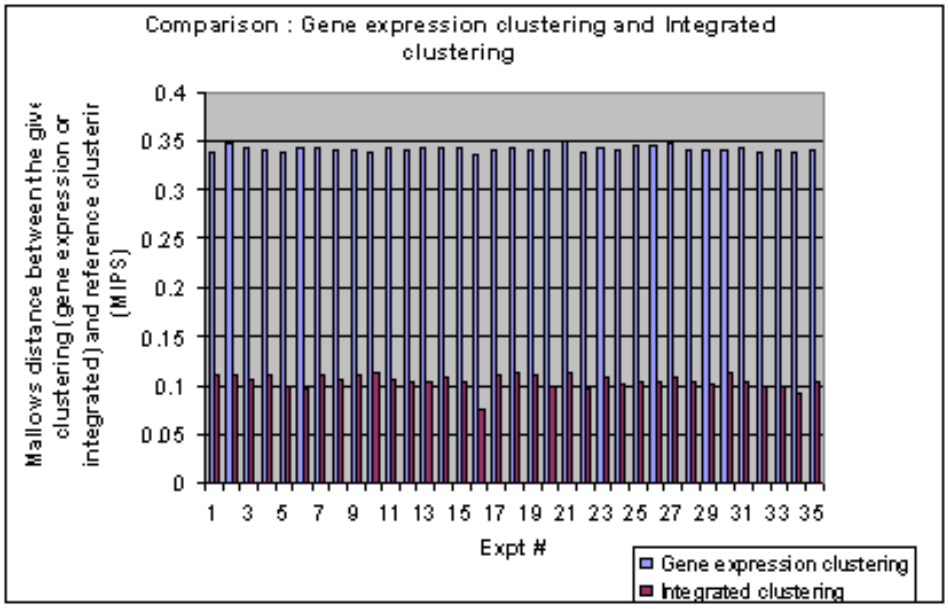


Fig. 5.6. Comparison of integrated clustering and clustering based on gene expression data

References

[1] Mips comprehensive yeast genome database (cgyd) website. http://mips.gsf.de/genre/proj/yeast/.

[2] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1993.

[3] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms:bagging,boosting and variants. *Machine Learning*, 36:105–142, 1999.

[4] A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, 2002.

[5] P. J. Boland. Majority systems and the condorcet jury theorem. *Statistician*, 38:181–189, 1989.

[6] N. Bolshakova and F. Azuaje. Machaon cve: cluster validation for gene expression data. *Bioinformatics*, 19(18):2494–2495, 2003.

[7] Francois Boutin and Mountaz Hascoet. Cluster validity indices for graph partitioning. In *Proc. of the Conference on Information Visualization IV' 2004*, 2004.

[8] L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123–140, 1996.

[9] Jenny Bryan. Problems in gene clustering based on gene expression data. *Journal of Multivariate Analysis*, 90(1):44–66, 2004.

[10] R. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Commun Statistics*, 3:1–27, 1974.

[11] Gopinath Chandroth. *Diagnostic Classifier Ensembles:Enforcing diversity for the reliability in the combination*. PhD thesis, University of Sheffield, November 1999.

[12] R. Cho, M. Campbell, E. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. Wolfsberg, A. Gabrielian, D. Landsman, D. Lockhart, and R. Davis. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2(1):65–73, 1998.

[13] S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P. O. Brown, and I. Herskowitz. The transcriptional program of sporulation in budding yeast. *Science*, 282:699–705, 1998.

[14] P. Cunningham and J. Carney. Diversity versus quality in classification ensembles based on feature selection. Technical Report TCD-CS-2000-02, Department of Computer Science, Trinity College, Dublin, 2000.

[15] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[16] S. Dongen. Performance criteria for graph clustering and markov cluster experiments. Technical Report INSR0012, Centrum voor Wiskundeen Informatica, 2000.

[17] Sandrine Dudoit and Jane Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7):0036.1–0036.21, June 2002.

[18] J. C. Dunn. Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4:95–104, 1974.

[19] D.E. Eckhardt and L.D. Lee. A theoretical basis for the analysis of multiversion software subject of coincident errors. *IEEE Transactions on Software Engineering*, 11(12):1511–1517, 1985.

[20] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95(25):14863–14868, December 1998.

[21] E. B. Fowlkes and C.L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78:553–584, 1983.

[22] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learing and an appl ication to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[23] Audrey P. Gasch, Paul T. Spellman, Camilla M. Kao, Orna Carmel-Harel, Michael B. Eisen, Gisela Storz, David Botstein, and Patrick O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11:4241–4257, 2000.

[24] G. Giacinto and F. Roli. Design of effective neural network ensembles for image classification processes. *Image Vision and Computing Journal*, 19(9/10):699–707, 2001.

[25] L. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

[26] J. A. Hartigan. Statistical theory in clustering. *Journal of Classification*, 2:63–76, 1985.

[27] Tin Kam Ho, Jonathan J. Hull, and Sargur N. Srihari. Decision combination in multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(1):66–75, January 1994.

[28] Ian Holmes and William J. Bruno. Finding regulatory elements using joint likelihoods for sequence and expression profile data. In Knab et al., editor, *Proc Int Conf Intell Syst Mol Biol*, volume 8, pages 202–210, 2000.

[29] Y.S. Huang and C. Y. Suen. A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:90–95, 1995.

[30] J. Kasturi and R. Acharya. Clustering of diverse genomic data using information fusion. In *SAC'04*, 2004.

[31] L. Kaufman and PJ Rousseeuw. *Finding groups in data : An Introduction to Cluster Analysis*. New York : Wiley, 1990.

[32] Josef Kittler, Mohamad Hatef, Robert P. W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, March 1998.

[33] R. Kohavi and D. H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In L. Saitta, editor, *Machine Learning : 13th International Conference*, pages 275–283. Morgan Kaufmann, 1996.

[34] T. Kohonen. *Self-Organizing Maps*. Springer Series in Information Sciences. Springer,Berlin, 1995.

[35] L. I. Kuncheva, C. J. Whitaker, C.A. Shipp, and R.P.W. Duin. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis and Applications*.

[36] L.I. Kuncheva. *Combining Pattern Classifiers*. Wiley Interscience, 2004.

[37] Ludmila I. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):281–286, February 2002.

[38] Ludmila I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationsh ip with the ensemble accuracy. *Machine Learning*, 51:181–207, 2003.

[39] L. Lam and C.Y. Suen. Application of majority voting to pattern recognition: An analysis of its behavior and performance. *IEEE Transactions on Systems, Man and Cybernetics*, 27(5):553–568, 1997.

[40] B. Littlewood and D.R. Miller. Conceptual modeling of coincident failures in multiversion software. *IEEE Transactions on Software Engineering*, 15(12):1596–1614, 1989.

[41] Marina Meila. Comparing clusterings. Department of Statistics, University of Washington, October 2002.

[42] G. W. Milligan and M.C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50:159–179, 1985.

[43] A. Narasimhamurthy. A framework for the analysis of majority voting. In Josef Bigün and Tomas Gustavsson, editors, *SCIA*, volume 2749 of *Lecture Notes in Computer Science*, pages 268–274. Springer, 2003.

[44] Anand Narasimhamurthy. Evaluation of diversity measures for binary classifier ensembles. In Josef Kittler et al. Nikunj C. Oza, Robi Polikar, editor, *Proceedings of 6th Workshop on Multiple Classifier Systems (MCS2005)*, volume 3541 of *Lecture Notes in Computer Science*, pages 267–277, 2005.

[45] Anand Narasimhamurthy. Theoretical bounds of majority voting performance for a binary classification problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1989–1995, 2005.

[46] Thomas H. ormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 2003.

[47] D. Partridge and W.J. Krzanowski. Software diversity:practical statistics for its measurement and exploitation. *Information Software Technology*, 39:707–717, 1997.

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

[49] C.R. Rao. Diversity : Its measurement,decomposition,apportionment and analysis. *Sankya:The Indian Journal of Statistics*, 44(1):1–22, 1982.

[50] P.J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[51] D. Ruta and B. Gabrys. Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting. In *Proc. 2nd Multiple Classifier systems workshop (MCS)*, volume 2096 of *Lecture Notes in Computer Science*, pages 399–408, 2001.

[52] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller. Probabilistic models for gene expression. *Bioinformatics*, pages S243–252, 2001.

[53] E. Segal, R. Yelensky, and D. Koller. Genome-wide discovery of transcriptional modules from dna sequence and gene expression. *Bioinformatics*, 1(1):1–10, 2003.

[54] C.A. Shipp and L.I. Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion*, 3:135–148, 2002.

[55] D. Skalak. The sources of increased accuracy for two proposed boosting algorithms. In *Proc. American Association for Artificial Intelligence AAAI-96, Integrating Multiple Learned Models Workshop*, 1996.

[56] P. Sneath and R. Sokal. *Numerical Taxonomy*. W.H. Freeman & Co., 1973.

[57] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.

[58] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters via the gap statistic. *Journal of Royal Statistical Society*.

[59] K. Tumer and J. Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(34):385–404, 1996.

[60] L. Xu, A. Krzyzak, and C.Y. Suen. Methods of combining multiple classifiers and their application to handwriting recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 22:418–435, 1992.

[61] G. U. Yule. On the association of attributes in statistics. *Philosophy Transactions*, 194:257–319, 1900.

[62] Ding Zhou, Jia Li, and Hongyuan Zha. A new mallows distance based metric for comparing clusterings. In *Proc. International Conference on Machine Learning (ICML)*, Bonn, Germany, August 2005.

Vita

Anand Narasimhamurthy

Anand Narasimhamurthy obtained a Bach. of Engineering degree in Electronics and Communication from Bangalore University, India in 1998. He joined the Ph.D. program in the Department of Computer Science and Engineering, Pennsylvania State University, University Park in August 1999. He obtained an M.Eng. degree in December 2003 and completed his Ph.D. in May 2006. His Ph.D. research encompasses the areas of classifier ensembles and cluster validation. His research interests include the areas of pattern recognition, machine learning and bioinformatics.