INDEPENDENT COMPONENT ANALYSIS FOR NAÏVE BAYES CLASSIFICATION

FAN LIWEI
(*M.Sc., Dalian University of Technology*)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF INDUSTRIAL & SYSTEMS ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2010

ACKNOWLEDGEMENT

I would like to express my utmost gratitude to my supervisor Associate Professor Poh Kim Leng, for his constructive comments and constant support throughout the whole course of my study. I greatly acknowledge Associate Professor Leong Tze Yun for her invaluable comments and suggestions on various aspects of my thesis research and writing. I would also like to thank Associate Professor Ng Szu Hui and Dr. Ng Kien Ming who served on my oral examination committee and provided me many helpful comments on an earlier version of this thesis.

I would like to thank the National University of Singapore for offering a Research Scholarship and the Department of Industrial and Systems Engineering for the use of its facilities, without any of which it would be impossible for me to carry out my thesis research. I am also very grateful to the members of SMAL Laboratory and the members of Bio-medical Decision Engineering group for their friendship and help in the past several years.

Special thanks go to my parents and my sister for their constant encouragement and support during in the past several. Finally, I must say thanks to my husband, Zhou Peng, for his encouragement and pushing throughout the entire period of my study.

TABLE OF CONTENTS

SUMMARY

Independent component analysis (ICA) has received increasing attention as a feature extraction technique for pattern classification. Some recent studies have shown that ICA and its variant called class-conditional ICA (CC-ICA) seem to be suitable for Bayesian classifiers, especially for naïve Bayes classifier. Nevertheless, there are still some limitations that may restrict the use of ICA/CC-ICA as a feature extraction method for naïve Bayes classifier in practice. This thesis focuses on several methodological and application issues in applying ICA to naïve Bayes classification for solving both single-label and multi-label problems.

In this study, we first carry out a comparative study of principal component analysis (PCA), ICA and CC-ICA for naïve Bayes classifier. It is found that CC-ICA is often advantageous over PCA and ICA in improving the performance of naïve Bayes classifier. However, CC-ICA often requires more training data to ensure that there are enough training data for each class. In the case where the sample size is smaller than the number of features, e.g. in microarray data analysis, the direct application of CC-ICA may become infeasible. To address this limitation, we propose a sequential feature extraction approach for naïve Bayes classification of microarray data. This offers researchers or data analysts a novel method for classifying datasets with small sample size but extremely large attribute size.

Despite the usefulness of the sequential feature extraction approach, the number of samples for some classes may be limited to just a few in microarray data analysis. The result is that CC-ICA cannot be used for these classes even if feature

selection has been done on the data. Therefore, we extend CC-ICA and present the partition-conditional independent component analysis (PC-ICA) for naïve Bayes classification of microarray data. As a feature extraction method, PC-ICA essentially represents a compromise between ICA and CC-ICA. It is particularly suitable for datasets which come with only few examples per class.

The research work mentioned above only deals with single-label naïve Bayes classification. Since multi-label classification has received much attention in different application domains, we finally investigate the usefulness of ICA for multi-label naïve Bayes (MLNB) classification and present the ICA-MLNB scheme for solving multi-label classification problems. This research does not only demonstrate the usefulness of ICA in improving MLNB but also enriches the application scope of the ICA feature extraction method.

LIST OF TABLES

LIST OF FIGURES

LIST OF NOTATIONS

ANN	Artificial neural networks
BN	Bayesian network
BSS	Blind source separation
CC-ICA	Class-conditional independent component analysis
ECG	Electrocardiogram
EEG	Electroencephalography
fMRI	Functional magnetic resonance imaging
ICA	Independent component analysis
ICAMM	ICA mixture model
KICA	Kernel independent component analysis
KNN	K-nearest neighborhood
KPCA	Kernel principal component analysis
LDA	Linear discriminant analysis
ML-KNN	Multi-label K-nearest neighborhood
MLNB	Multi-label naïve Bayes
MRMR	Minimum redundancy maximum relevance
NB	Naïve Bayes
PCA	Principal component analysis
PC-ICA	Partition-conditional independent component analysis
TCA	Tree-dependent component analysis
TICA	Topographic independent component analysis
SVM	Support vector machines

CHAPTER 1 INTRODUCTION

Independent component analysis (ICA) is a useful feature extraction technique in pattern classification. This thesis contributes to the development of various ICA-based feature extraction methods or schemes for the naïve Bayes model to classify different types of datasets. In this introductory chapter, we first provide the background and the motivation for this study, which is followed by a brief overview of ICA-based feature extraction methods. After that we outline the scope and objective of this study. Finally, we summarize the content and the structure.

1.1 Background and motivation

Pattern classification, which aims to classify data based on a priori knowledge or statistical information extracted from the patterns, is a fundamental problem in artificial intelligence. Nowadays, pattern classification is a very active area of research that draws the attention of researchers from different disciplines including engineering, computer science, statistics and even social sciences. Since better classification results can provide useful information for decision making, numerous studies have been devoted to improve the performance of pattern classification from different aspects.

Intuitively, better classification results may be obtained from a set of representative features constructed from the knowledge of domain experts. When such expert knowledge is not available, general feature extraction techniques seem to be very useful. They helps to remove redundant or irrelevant information, discover the

underlying structure, facilitate the subsequent analysis, and improve classification performance. In the past several decades, machine learning researchers have developed a number of feature extraction methods, such as, principal component analysis (PCA), multifactor dimensionality reduction, partial least squares regression, and independent component analysis (ICA). Of the various feature extraction methods, independent component analysis (ICA) is recently found to be very useful and effective in helping to extract representative features in pattern classification.

ICA is a relatively new statistical and computational technique for revealing the hidden factors that underlie a set of random variables. Although ICA was initially developed to solve the blind source separation (BSS) problem, previous studies have shown that ICA can serve as an effective feature extraction method for improving the classification performance in both supervised classification (Zhang et al., 1999; Kwak et al., 2001; Cao and Chong, 2002; Herrero et al., 2005; Chuang and Shih, 2006; Widodo et al., 2007; Yu and Chou, 2008) and unsupervised classification (Lee and Batzoglou, 2003; Kapoor et al., 2005; Kwak, 2008). It has also been found that ICA may help to improve the performance of various classifiers, such as support vector machines, artificial neural networks, decisions trees, hidden Markov models, and the naïve Bayes classifier (Sanchez-Poblador et al., 2004; Li et al., 2005; Melissant et al., 2005; Yang et al., 2005).

NB, also called simple Bayesian classifier, is a simple Bayesian network that assumes all features are conditionally independent given the class variable. Since no structure learning is required, it is very easy to construct and implement NB in practice. Despite its simplicity, the naïve Bayes has been found to be competitive with

other more advanced and sophisticated classifiers (Friedman et al., 1997). It is therefore not surprising that naïve Bayes classifier has gained great popularity in solving various classification problems. Nevertheless, the class-conditional independence assumption between features taken by naïve Bayes classifier is often violated in some real-world applications. Since ICA aims to transform the original features into new features that are statistically independent of each other as possible, the ICA transformation is likely to fit well the NB model and its independent assumption (Bressan and Vitria, 2002).

Several earlier studies have been devoted to investigate the applicability of ICA as a feature extraction tool for the naïve Bayes classifier. It was found that ICA and its variants, such as class-conditional ICA (CC-ICA), are often capable of improving the classification performance of the NB model. Nevertheless, some limitations of CC-ICA may restrict the use of CC-ICA as a feature extraction tool to improve the performance of NB classifier in microarray data analysis. In this thesis, we propose several ICA-based feature extraction methods for addressing the limitations in applying ICA to naïve Bayes classification of microarray data. In addition, since most previous studies mainly focused on single-label classification problems, the question of how to adapt the ICA feature extraction method for multi-label classification problems remains to be investigated. Therefore, we also investigate the use of ICA as a feature extraction method for multi-label naïve Bayes classification.

1.2 Overview of ICA-based feature extraction methods

With the development of modern science and technology, large amounts of information can be obtained and recorded for a variety of problems. However, the existence of too much information may often reduce the effectiveness of data analysis. In pattern classification, it implies that the performance of a classifier adopted may worsen when too many features are used to train the classifier. This is due to the fact that some features are redundant for constructing the classifier. Therefore, many feature selection or feature extraction methods have been proposed to minimize the cons of the irrelevant or redundant features. Feature selection methods aim to select the most relevant features, while feature extraction methods attempt to transform features into a new (and may be reduced) set of more representative features.

Several ICA-based methods have been proposed and used for feature extraction in pattern classification. The first one may be referred to as “the direct ICA feature extraction method”, in which ICA is directly used to transform original features into a new set of features for classification use. Since ICA assumes that the variables after the transformation are independent of each other, the features obtained from the direct ICA feature extraction method are as independent with each other as possible. As a result, the new features obtained seem to be more consistent with the assumption of the naïve Bayes classifier compared to the original features. Therefore, the classification performance of the naïve Bayes classifier could be improved using the ICA features (Zhang et al., 1999).

Nevertheless, the strong independence assumption used in the ICA computation may not be appropriate for some real-world datasets. To overcome this

limitation, Hyvarinen et al. (2001a) proposed topographic independent component analysis (TICA) by relaxing the strong independence assumption. TICA uses contrast functions including the higher-order correlations between the components to achieve the relaxation of the strong independence assumption. However, in practice the empirical contrast functions are difficult to construct.

Though the strong independence assumption is inappropriate for some real-world datasets, it may offer the advantages for some specific classifiers such as the NB model. Since the strong independence assumption of ICA makes the new features as independent as possible, the features obtained from ICA may be more consistent with the underlying assumption of naive Bayes classifier. Furthermore, Bressan and Vitria (2002) proposed the CC-ICA feature extraction method that applies ICA within each class, which can help to extract the representative features from the original features within each class. Their empirical studies showed that the CC-ICA feature extraction method may be more suitable than the direct ICA feature extraction method for the NB classifier.

A limitation of the CC-ICA feature extraction method is that it requires more training data than the direct ICA feature extraction method in implementation. Usually, the number of samples should not be less than the number of features within each class for the CC-ICA feature extraction method, while for the direct ICA feature extraction method the number of samples for all the classes is required to be not less than the number of features. However, there may not be enough training data for some real-world applications such as microarray data analysis due to the very high data collection cost. Therefore, it is meaningful to extend CC-ICA and develop new ICA-

based feature extraction method so that it is applicable to the case of small datasets. Since ICA-based feature extraction methods are mainly used for addressing single-label classification problems, it would also be very useful to investigate the usefulness of ICA as a feature extraction method in solving multi-label classification problems.

1.3 Research scope and objectives

The main objective of this thesis is to address several methodological and application issues in applying ICA for feature extraction, which could be helpful to those who expect to use it to improve the performance of the naïve Bayes classifier in solving both single-label and multi-label classification problems. In many cases ICA can extract more useful information than principal component analysis (PCA) for the succeeding classifiers since ICA can make use of high-order statistics information. However, a feature extraction method cannot always perform better than others for all application domains and all classifiers. It is therefore meaningful to compare various feature extraction methods with respect to the classification performance of the succeeding classifier.

Our comparative study found that CC-ICA is often advantageous over PCA and ICA in improving the performance of naïve Bayes classifier. However, the CC-ICA requires more training data to ensure that there are enough training data for each class. In the case where the sample size is much less than the number of features, e.g. in microarray data analysis, the direct implementation of CC-ICA may become infeasible. Therefore, we propose a sequential feature extraction approach for naïve Bayes classification of microarray data. In the sequential feature extraction approach, stepwise regression is first applied for feature selection and CC-ICA is then used for

feature transformation. It is expected that the proposed approach could be adopted by researchers to solve such classification problems with small sample size but extremely large attribute size in different domains including microarray data analysis.

For some microarray datasets, there may be only few samples for some classes so that CC-ICA cannot be applied after feature selection. Therefore, we extend CC-ICA and propose partition-conditional independent component analysis (PC-ICA) for naïve Bayes classification of microarray data. In this research, we applied "minimum redundancy maximum relevance" (MRMR) principle based on mutual information to select informative features and applied PC-ICA for feature transformation for each partition. Compared to ICA and CC-ICA, PC-ICA represents an in-between concept. If each class has enough samples to do ICA, there is no need to combine the samples into partitions and PC-ICA will become CC-ICA. If all the classes are grouped into one partition, CC-ICA will collapse to ICA. PC-ICA could make full use of samples in the partitions including several classes to improve the performance of naïve Bayes classifier. It is expected that PC-ICA could help to solve the multi-class problems even if the number of training examples is small.

For multi-label classification problems, feature extraction is also essential for improving classification performance. Based on the experience of ICA for single-label problems, ICA transformation could make the features more appropriate for multi-label naïve Bayes classification. However, none of the previous studies dealt with the use of ICA as a feature method for multi-label naïve Bayes (MLNB) classifier. Therefore, we propose the ICA-MLNB scheme for solving multi-label classification problems. It is expected that ICA-MLNB could not only expand the

application of ICA in pattern classification but also be adopted by researchers who are interested in applying naïve Bayes to solve multi-label problems.

1.4 Contributions of this thesis

The main contributions of the work presented in this thesis can be summarized from the point of view of methodological and application as follows.

In terms of methodology, we have proposed a new sequential feature extraction method for naïve Bayes classification of microarray data. This method reduces the number of features by the stepwise regression and transforms the features to a small set of independent features. Despite the simplicity of the proposed method, our experimental results showed that it can improve the performance of the classifier significantly. In addition, we proposed PC-ICA for solving multi-class problems. Instead of applying ICA within each class in CC-ICA, PC-ICA uses ICA to do feature extraction within each partition which may consist of several small-size classes. Experimental results on several microarray datasets have shown that PC-ICA usually leads to better performance than ICA for naïve Bayes classification of microarray data.

In terms of application, we first compared the ICA, PCA and CC-ICA feature extraction methods for the NB classifier. It is found that all the three methods keep improving the performance of the naïve Bayes classifier with the increase of the number of attributes. Although CC-ICA has been found to be superior to PCA and ICA in most cases, it may not be suitable for the case where the sample size of each class is not sufficiently large. This is the motivation of the sequential feature extraction method and PC-ICA presented in this thesis. Since none of the previous

studies dealt with the use of ICA for multi-label naïve Bayes classification, we investigate the usefulness of ICA as a feature extraction method for multi-label naïve Bayes classifier and propose the ICA-MLNB scheme for solving multi-label classification problems. Our experimental results demonstrate the effectiveness of the scheme in improving the performance of multi-label naïve Bayes classification.

1.5 Organization of the thesis

This thesis focuses on the study of ICA-based feature extraction methods for the naïve Bayes classifier in solving single and multi -label classification problems. It consists of seven chapters. Figure 1.1 shows the main content of each chapter and the relationships among different chapters.

Chapter 2 reviews the use of ICA as a feature extraction tool in pattern classification. Different ICA feature extraction methods and their applications are summarized and examined. Compared with other feature extraction methods, the superiority of ICA based feature extraction methods lies in their ability of utilizing high-order statistics and their suitability for the non-Gaussian case. Our literature review also found that ICA is particularly suitable for the naïve Bayes classifier but there are still several limitations worth further investigating.

In Chapter 3, we first introduce the naïve Bayes model and three feature extraction methods, namely PCA, ICA and CC-ICA. Then we empirically compare them for the naïve Bayes classifier with regards to the classification performance. Our experimental results have shown that all three methods can improve the performance of the naïve Bayes classifier. In general, CC-ICA outperforms PCA and ICA in terms

of the classification accuracy. However, CC-ICA requires more training data to ensure that there are enough training data for each class.

Chapter 4 presents a sequential feature extraction approach for naïve Bayes classification of microarray data. The proposed feature extraction approach starts from gene selection by stepwise regression, which is a simple but effective dimension reduction technique following the MRMR principle. The data on the genes selected are then transformed by CC-ICA, which makes the new features after transformation become as independent as possible. In Chapter 5, we extend CC-ICA and propose PC-ICA for naïve Bayes classification of microarray data. CC-ICA applies ICA for each class, while PC-ICA uses ICA to do feature extraction within each partition consisting of several small-size classes. As such, it represents a compromise between ICA and CC-ICA. The effectiveness of PC-ICA has been demonstrated by our experimental studies on several microarray datasets.

While Chapters 4 and 5 deal with single-label classification problems, Chapter 6 is mainly concerned with the use of ICA in multi-label naïve Bayes classification problems. In Chapter 6, we apply ICA to multi-label naïve Bayes and propose the ICA-MLNB scheme for multi-label classification. The results obtained from our experimental studies have shown the effectiveness of the ICA-MLNB scheme and also demonstrate the usefulness of ICA as a feature extraction method in solving multi-label classification problems.

Chapter 7 gives the conclusion of this thesis as well as some potential future research topics.

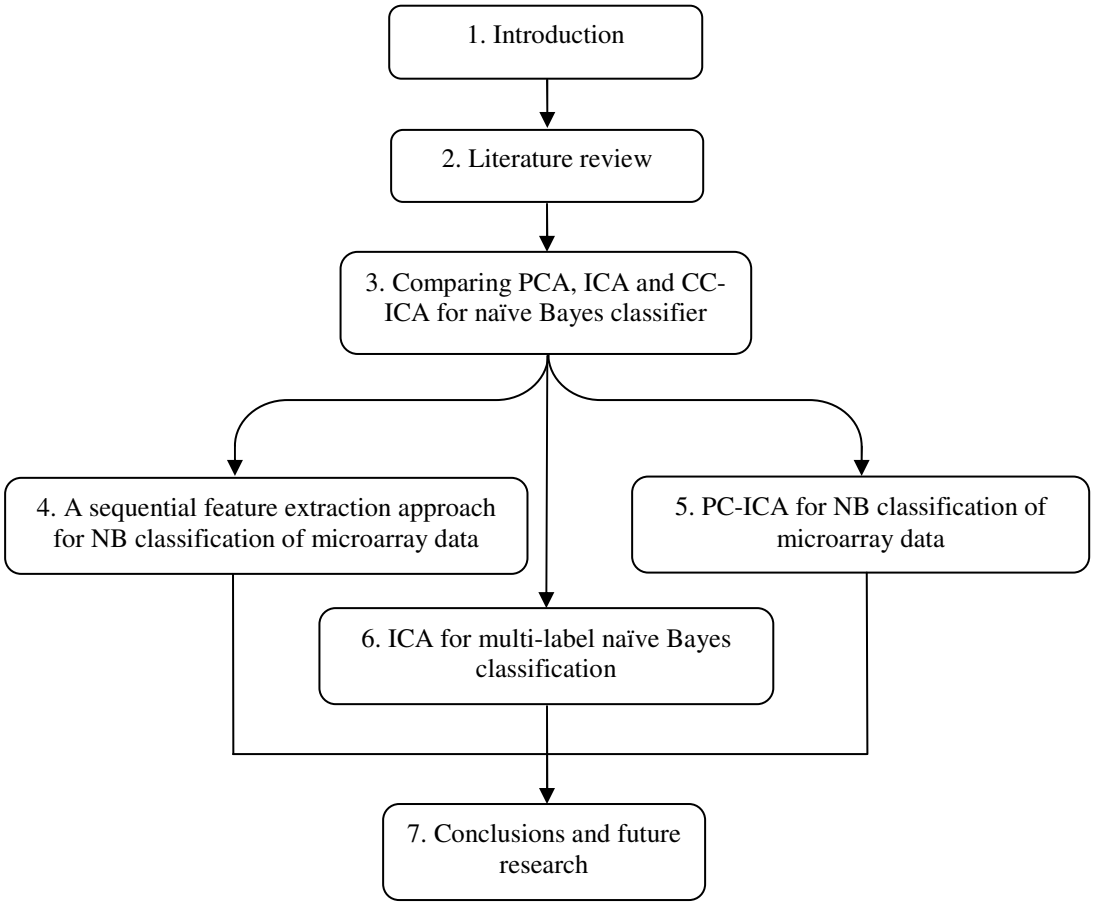


Fig. 1.1 Structure of the thesis

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

Pattern classification problems are usually very complex and cannot be well solved by only one procedure (Jain et al., 2000). For the purpose of reducing computational costs and improving classification performance, certain preprocessing procedure is often adopted to select the most informative features or to appropriately transform the original data into a new set of data. The preprocessing procedure is often termed as feature selection or feature extraction. Previous researchers have proposed a number of feature extraction methods for improving the performance of classification. Among the various feature extraction methods, ICA has received increasing attention due to its usefulness in helping extract representative features for classification.

As mentioned in Chapter 1, ICA is a relatively new statistical technique for finding hidden factors or components to give a novel representation of multivariate data. It was originally proposed by Jutten and Herault (1991) for solving the blind source separation (BSS) problems. In this application, ICA can help to find the underlying independent components, which may provide valuable information for data analysis. As a feature extraction technique, ICA may be viewed as a generalization of PCA. PCA tries to find uncorrelated variables to represent the original multivariate data, whereas ICA attempts to obtain statistically independent variables to represent the original multivariate data, especially in the case of non-Gaussian distribution.

Theoretically, ICA is a computational algorithm to search for a linear transformation that minimizes the statistical dependence between the components of a multivariate variable. Many important theoretical landmarks in ICA, e.g. Common (1994), Bell and Sejnowski (1995), Amari et al. (1996), Cardoso and Laheld (1996), and Hyvarinen and Oja (1997), were established in the 1990s. Since then, ICA has gained more and more popularity in a wide spectrum of areas, e.g. biomedical signal processing, image recognition, fault diagnosis, data mining and financial time series analysis. In most of the previous studies, ICA was taken as an effective preprocessing procedure for further data analysis. It is therefore not surprising that ICA has also received much attention in pattern classification as a feature extraction method.

This chapter provides a review of the most commonly used ICA-based feature extraction methods for pattern classification. The basic ICA model is first briefly introduced in Section 2.2. Section 2.3 presents the direct ICA feature extraction method with more emphases on supervised classification, which is followed by several other ICA-based feature extraction methods presented in Sections 2.4 and 2.5. Section 2.6 summarizes the concluding comments.

2.2 Basic ICA model

ICA was originally developed to deal with BSS problems which are closely related to the classical cocktail-party problem. Assume that there are three microphones used to record time signals in different locations in one room. The amplitudes of the three signals are respectively denoted as $x_1(t), x_2(t)$ and $x_3(t)$, where t is the time index. Further assume that each signal is a weighted sum of three

different source sound signals which are respectively denoted as $s_1(t), s_2(t)$ and $s_3(t)$. The relationship between the three source sound signals and the three microphones' sound signals may be described as

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t) \\ x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t) \\ x_3(t) &= a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t) \end{aligned} \tag{2.1}$$

where a_{ij} ($i, j = 1,2,3$) represent the unknown weights that reflect the distances of the microphones from the sound sources. The problem is to separate the three independent sound sources only based on the three microphones' records.

The simple BSS problem with three sources can be generalized to the case of *n* sources. Suppose that there are *n* observed random variables $x_1, x_2, \cdots, x_n$, which are modeled as the linear combinations of *n* random source variables $s_1, s_2, \cdots, s_n$. Mathematically, it can be expressed as

$$x_i = a_{i1}s_1 + a_{i2}s_2 + \cdots + a_{in}s_n, \quad i = 1,2,\cdots,n \tag{2.2}$$

where a_{ij} ($i, j = 1,2,\cdots,n$) represents the mixing coefficients, and s_i ($i = 1,2,\cdots,n$) are assumed to be mutually statistically independents.

Equation (2.2) can also be represented in the vector-matrix form as follows:

$$x = \mathbf{A}s \tag{2.3}$$

where x is the random column vector with elements $x_1, x_2, \cdots, x_n$, s is the random column vector with elements $s_1, s_2, \cdots, s_n$, and A is the mixing matrix with elements a_{ij}.

In ICA, Eq. (2.3) is often re-written as

$$y = \mathbf{W}x \tag{2.4}$$

where $\mathbf{W} = \mathbf{A}^{-1}$ is the demixing matrix and $y = [y_1, y_2, \cdots, y_n]^T$ denotes the independent components. The task is to estimate the demixing matrix and independent components only based on the mixed observations, which can be done by various ICA algorithms built upon a certain principle.

There are various principles to solve the ICA model, such as maximum likelihood, nongaussianity maximization, and mutual information minimization. In computation, each of the principles generates a specific objective function and its optimization will enable the ICA estimation. Various optimization algorithms may be applied to solve the optimization problems and obtain the independent components.

2.3 Direct ICA feature extraction method

In pattern classification, principal component analysis (PCA) and linear discriminant analysis (LDA) are two popular feature extraction methods. Like PCA and LDA, ICA can also be directly used for feature extraction. Given the variables $x_1, x_2, \cdots, x_n$, the underlying independent variables $s_1, s_2, \cdots, s_m$ $(m \le n)$ and the demixing matrix W can be obtained by different ICA algorithms. Then the

independent variables $s_1, s_2, \cdots, s_m$ $(m \le n)$ obtained can be directly used to train the classifier. Meanwhile, the demixing matrix $\mathbf{W}$ can be directly applied to transform the test data for classification. Since this method involves the direct application of ICA, we here refer to it as "the direct ICA feature extraction method". Figure 2.1 shows the flow chart of the direct ICA feature extraction method for pattern classification.

As shown in Fig. 2.1, to construct an appropriate classifier we usually need to first split the dataset available into training and test datasets. The datasets are preprocessed by certain feature selection procedures. For the training dataset after feature transformation, ICA is used to do the feature extraction and obtain the demixing matrix $\mathbf{W}$, which can then be used to do feature transformation for the test data after feature selection. Meanwhile, the training and test datasets after ICA-based feature extraction can be used to construct an appropriate classifier by learning its parameters and examining its classification performance. In pattern classification, the direct ICA feature extraction method has been widely adopted in both supervised classification and unsupervised classification. In the following, we shall first give a review of some relevant studies divided into supervised and unsupervised classifications, where there are more studies in the supervised classification group. Then we briefly discuss the issue of classifier selection as the direct ICA feature extraction method may be integrated with various classifiers.

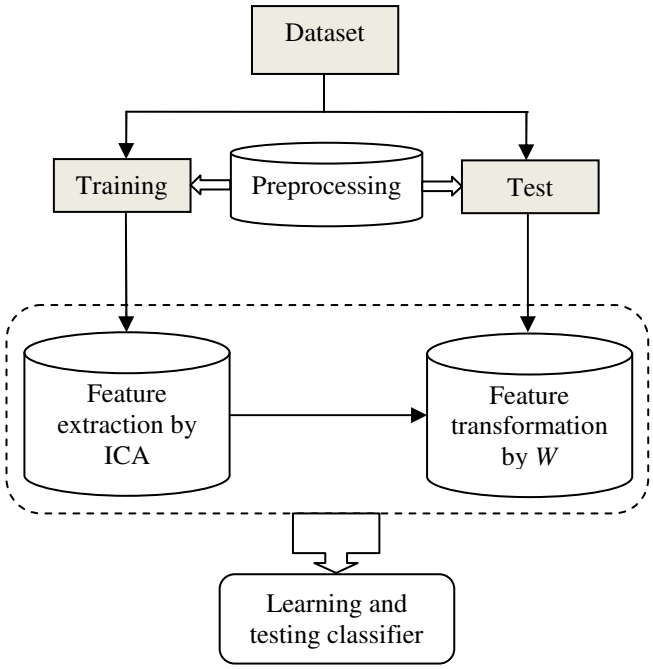


Fig. 2.1. Flow chart of the direct ICA feature extraction method for classification

2.3.1 Supervised classification

Supervised classification refers to the type of classification in which the label for each sample is known in advance. In the training process, a classifier is constructed from the features and labels of sample data, in which the direct ICA feature extraction method plays a major role. In the test process, the label for a new given sample will be predicted by the classifier obtained. Application areas of the supervised classification based on the direct ICA feature extraction method include face recognition, signal analysis, image analysis, text categorization, etc.

(1) Face recognition

Face recognition is a major application area in which the direct ICA feature extraction method has gained in popularity. In this application, the earliest study could

be attributed to Bartlett and Sejnowski (1997) who proposed an ICA representation of face images and compared it with the PCA representation of the same face images. Their study showed that ICA provides a better representation than PCA because in the latter only the second-order statistics are decorrelated. Guan and Szu (1999) compared the direct ICA and PCA feature extraction methods for the nearest neighbor classifier for face recognition. Their study found that ICA outperforms PCA when one training image per person is used. It indicates that the direct ICA feature extraction method may be a better alternative when only few training samples are available. Also using the nearest neighbor classifier, Donato et al. (1999) showed that ICA representation performed as well as the Gabor representation and better than PCA representation, which are popular representation methods in classifying facial actions.

Kim et al. (2004) proposed an ICA based face recognition scheme, which was found to be robust to the illumination and pose variations. An interesting finding by Kim et al. (2004) is that in the residual face space ICA provides a more efficient encoding in terms of redundancy reduction than PCA.

In face recognition, the algorithms based only on the visual spectrum are not robust enough to be used in uncontrolled environments. Motivated by this question, Chen et al. (2007) proposed to fuse information from visual spectrum and infrared imagery to achieve better results. Their scheme also employs ICA as a feature extraction method for the support vector machine (SVM) classifier. Their experimental results show that the scheme improves recognition performance substantially.

Based on an application of the direct ICA feature extraction method to Yale Face Databases and AT&T Face Databases, Kwak et al. (2002) found that ICA transformation can make new features as independent with each other as possible. Similar to earlier studies, the study by Kwak et al. (2002) also showed that ICA outperforms PCA and LDA as feature extraction method for face recognition. Subsequently, Kwak and Choi (2003) further extended the work by Kwak et al. (2002) by developing a stability condition for the earlier study. The two earlier studies mentioned above focused on the two-class face recognition problems. More recently, Kwak (2008) extended the use of the direct ICA feature extraction method to the case of multi-class face recognition using the nearest neighborhood classifier. The experimental results for several face databases demonstrated the usefulness of the direct ICA feature extraction method in solving multi-class face recognition problems.

(2) Signal analysis

Signal analysis is also a major application area where the direct ICA feature extraction method has been widely used. Applications of the direct ICA feature extraction method to signal analysis include data analysis of functional magnetic resonance imaging (fMRI), electroencephalography (EEG), and electrocardiogram (ECG). Previous studies have shown that the direct ICA feature extraction method can help to extract task-related components and reduce the noise of signals effectively (Stone, 2004).

Laubach et al. (1999) compared PCA and ICA for quantifying neuronal ensemble interactions, and found that ICA performs better than PCA in terms of the classification performance. The study by Hoya et al. (2003) attempted to classify the

EEG signals of letter imagery tasks by combining ICA and probabilistic neural network. It was found that the inclusion of ICA in the classifier led to an improvement of classification accuracy rate by around 17-30%. Melissant et al. (2005) studied the EEG measurements for detecting Alzheimer's disease, and found that the classification results for the group with severe Alzheimer's disease using ICA are comparable to the best classification results in the literature. In addition, the direct ICA feature extraction method has also been applied to the discrimination of mental tasks for EEG-based brain computer interface systems. It was found that ICA integrated with the SVM classifier may produce good classification performance, which could be attributed to the fact that the temporal information from a window of data is effectively extracted by ICA.

The direct ICA feature extraction method has also been applied to heartbeat classification. Herrero et al. (2005) used ICA and machining pursuits to do feature extraction for heartbeat classification. Their conclusion is that ICA could improve the system's ability of discriminating various beat signals, which is particularly useful in clinical use. More recently, Yu and Chou (2008) proposed to integrate ICA and neural networks for ECG beat classification. Their experimental results showed that the scheme of integrating ICA and neural networks is of great potential in the computer-aided diagnosis of heart diseases based on ECG signals.

(3) Image analysis

Image analysis usually requires effective feature extraction through various feature extraction methods such as ICA. Hoyer and Hyvarinen (2000) investigated the use of ICA in decomposing natural color and stereo images. They found that the

features extracted by ICA could be directly used for pattern recognition of color or stereo data. Karvonen and Simila (2001) also found that the ICA representation of data is useful to improve the classification performance in sea ice Synthetic aperture radar (SAR) image analysis. Fortuna et al. (2002) showed that ICA performs better than PCA as a feature extraction method in object recognition under varying illumination.

Leo and Distante (2003) proposed a comparative study of wavelet and ICA for automatic ball recognition using the back propagation neural network. Borgne et al. (2004) applied ICA to extract features from natural images, and use the new features for a K-nearest neighborhood (KNN) classification paradigm. Their experimental results demonstrated the effectiveness of the direct ICA feature extraction method in classifying natural images. Based on a large set of consumer photographs, the Fourier-transformed images, Boutell and Luo (2005) applied the direct ICA feature extraction method to derive their sparse representations for classification. The empirical analysis results showed the superiority of ICA over PCA as a feature extraction technique.

In addition to the traditional ICA model, other types of ICA models have also been directly used for feature extraction in image analysis. For instance, Cheng et al. (2004) showed the effectiveness of kernel independent component analysis (KICA) for texture feature extraction. The study by Luo and Boutell (2005) used overcomplete ICA for the heuristic and support vector machine classification of Fourier-transformed images and demonstrated its effectiveness as a feature extraction method.

(4) UCI machine learning repository

Some researchers have also applied the direct ICA feature extraction method to analyze the data from the UCI machine learning repository. Kwak et al. (2001) added class information to the Wisconsin Breast Cancer Diagnosis and Chess End-Game datasets, which plays an important role in extracting useful features for classification. Experimental results showed that the features extracted by ICA are more useful than the original features in classification.

Using the nine continuous datasets from the UCI machine learning repository, Prasad et al. (2004) evaluated the integration of the direct ICA feature extraction method with naïve Bayes, instance based learning and decision trees. Their experimental results showed that naïve Bayes classifier outperforms other classifiers for five of the nine datasets. For the remaining four datasets, naïve Bayes classifier is comparable with other classifiers. It could be attributed to the fact that the naïve Bayes classifier is known to be optimal when attributes are independent with each other given the class. Based on another nine datasets from the UCI machine learning repository, Sanchez-Poblador et al. (2004) examined the applicability of ICA as a feature extraction technique for decision trees and multilayer perceptrons. It was found that for some datasets the direct ICA feature extraction would benefit the classification, while for others the benefit was minor. The conclusion was that the use of ICA as a preprocessing technique may improve the classification performance when the feature space has a certain structure.

(5) Microarray data analysis

Accurate classification of microarray data is very important for successful diagnosis and treatment of diseases such as cancer. Recently, some researchers have also applied the direct ICA feature extraction method to help improve the classification performance of microarray data analysis. For instance, Zheng et al. (2006) combined ICA with the sequential floating forward technique to do feature extraction for classifying the DNA microarray data. Their study showed the effectiveness of the direct ICA feature extraction method in classifying microarray data. More recently, Liu et al. (2009a,b) developed a genetic algorithm/ICA based ensemble learning system to help improve the performance of microarray data classification. Their experimental results further demonstrated the usefulness of the direct ICA feature extraction method in microarray data analysis.

(6) Miscellaneous

In addition to the application areas described above, the direct ICA feature extraction method has also been used to help solve the classification problems in other application areas. Here we shall only give two examples on the use of ICA in text categorization and fault diagnosis.

Text categorization is based on statistical representations of documents that usually consist of a huge dimension. It is necessary to find an effective dimension reduction for a better representation of word histograms. In this application context, Kolenda et al. (2002) applied the direct ICA feature extraction method and found that the ICA representation is better than PCA representation in explaining the group

structure. The study by Widodo et al. (2007) integrated ICA and SVM for intelligent faults diagnosis of induction motors, which showed the advantage of ICA over PCA as a feature extraction technique.

2.3.2 Unsupervised classification

In contrast to supervised classification, unsupervised classification does not require user to input sample classes in performing classification. It uses certain techniques to determine which features are related with each other and which samples can be grouped into a class. In classification process, the user can specify the desired number of output classes. The applicability of the direct ICA feature extraction method in unsupervised classification has also been widely explored. Lee et al. (2000) proposed the ICA mixture model (ICAMM) for unsupervised classification of non-Gaussian classes. Its classification performance was found to be comparable to or advantageous over those obtained by AutoClass that uses a Gaussian mixture model.

The ICAMM has been used for unsupervised image classification, segmentation, and enhancement (Lee and Lewicki, 2002). Several other researchers, including Hashimoto (2002) and Shah et al. (2002, 2003, 2004), also applied the ICAMM to solve other image classification problems using different algorithms. These earlier studies showed that in image analysis the unsupervised classification based on ICAMM could produce higher accuracy than the *K*-means algorithm, which illustrates the benefits of employing higher order statistics in classification.

In Bae et al. (2000), the ICAMM has also been applied for blind signal separation in teleconferencing. The authors found that ICAMM could learn well the

unmixing matrices given the number of classes. However, if no optimal number of classes were given, ICAMM would likely result in a local optimum in most cases. Therefore, Oliveira and Romero (2004) proposed the Enhanced ICAMM to modify the learning algorithm based on a gradient optimization technique. This new model improves the performance of the original ICAMM to some degree. In future, other estimation principles and algorithms are expected to be explored in order to further improve the classification performance of ICAMM.

Unsupervised classification has also been used in microarray data analysis. An example is the study by Lee and Batzoglou (2003), which applied linear and nonlinear ICA to project microarray data into statistically independent components that correspond to putative biological processes. Then the genes can be grouped into clusters based on the independent components obtained. It has been found that ICA outperformed methods such as PCA, K-means clustering and the Plaid model, in constructing functionally coherent clusters on microarray datasets. Szu (2002) proposed a spectral ICA-based unsupervised classification algorithm for space-variant imaging for breast cancer detections, which may offer an unbiased, more sensitive, accurate, and generally more effective way to track the development of breast cancer. Suri (2003) also compared ICA and PCA for detecting coregulated gene groups in microarray data, and found that ICA may be more useful than PCA in finding coregulated gene groups.

2.3.3 Comparisons between various feature extraction methods and classifiers

In pattern classification, there are many other feature extraction methods for use in addition to ICA. Some researchers have therefore conducted studies on comparing the direct ICA feature extraction method with other feature extraction methods such as PCA. For example, Cao and Chong (2002) compared PCA, Kernel PCA (KPCA) and ICA for SVM classification. They found that SVM integrated with PCA, KPCA or ICA performs better than that without any feature extraction methods in terms of classification accuracy. Furthermore, the KPCA and ICA feature extraction methods seem to be more suitable than PCA for the SVM classifier. Deniz et al. (2003) conducted a comparison of classification performance between PCA and ICA for SVM in face recognition. Their experiment results showed that PCA and ICA are comparable, which may be due to the fact that the SVM classifier is insensitive to the representation space.

As the training time for ICA was more than that for PCA, Deniz et al. (2003) suggested the use of PCA feature extraction method if the SVM classifier is adopted. Fortuna and Capson (2004) also compared the PCA and ICA feature extraction methods for face recognition based on SVM. Different from the study by Deniz et al. (2003), Fortuna and Capson (2004) drew the conclusion that ICA outperformed PCA in its generalization ability by improving the margin and reducing the number of support vectors. Yang et al. (2005) used the SAR image data to compare PCA and ICA feature extraction methods for KNN and SVM classifiers. Their conclusion is that PCA and ICA are comparable with each other.

Since the direct ICA feature extraction method may be integrated with various classifiers, it is meaningful to compare the performance of various classifiers with the direct ICA feature extraction method. Jain and Huang (2004a) integrated ICA and LDA for gender classification of face recognition. Their study showed a significant improvement in gender classification accuracy rate after the direct ICA feature extraction method is used. Furthermore, Jain & Huang (2004b) applied ICA representation of facial images to nearest neighbor classifier, LDA and SVM for gender identification. The experimental results showed that SVM with ICA may have better classification performance than the other two. Kocsor and Toth (2004) compared the performance of artificial neural networks (ANN), SVM and Gaussian mixture modeling (GMM) with feature extraction methods such as PCA, ICA, LDA and springy discriminant analysis (SDA) for phoneme classification. Their experimental results showed that SVM integrated with ICA has better classification performance than other schemes.

Gilmore et al. (2004) applied ICA for image feature extraction and compared the performance of vector quantization, neural network and Fisher classifier. Although the performance of all the three classifiers has been improved by ICA, the Fisher classifier seems to have the best classification performance among the three classifiers. Prasad et al. (2004) tested the performance of naïve Bayes, C4.5 and Seeded K-means integrated with ICA through the classification of Emphysema in High Resolution Computer Tomography (HRCT) images. It is found that naïve Bayes in the ICA space achieved the best classification performance. This is not surprising as the independence assumption between attributes in ICA space is consistent with the underlying assumption of naïve Bayes.

Based on the previous studies such as those described above, we may draw a conclusion that the direct ICA feature extraction method often performs better than other methods such as PCA in improving classification performance. Although the SVM classifier integrated with ICA was found to achieve better classification performance in many cases, none of the classifiers always dominates others. In some cases, some simple classifiers are competitive with more complicated ones. In practice, the choice between various classifiers should be made with factors such as "ease of use" and "accuracy" in mind.

2.4 Class-conditional ICA feature extraction method

The class-conditional ICA (CC-ICA), proposed by Bressan and Vitria (2001, 2002), is a preprocessing procedure for naïve Bayes classifier. Its idea is to extract the representative features from the original features within each class in the training data. At the same time, a demixing matrix $\mathbf{W}_i$ for each class can be estimated. Given a test instance, the representative features can be transformed by the corresponding demixing matrix for each class. The instance is then classified as the class with the highest posterior probability according to the naïve Bayes classifier. The process can be described as Fig. 2.2.

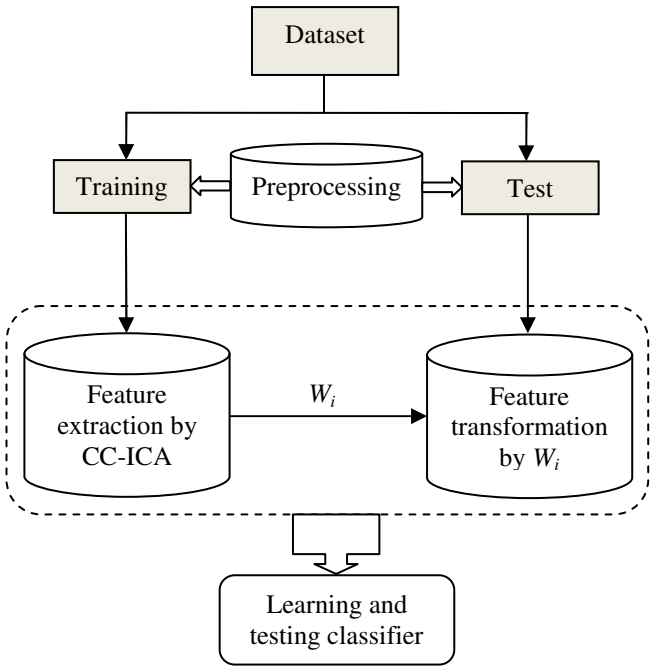


Fig. 2.2. Flow chart of the CC-ICA feature extraction method for classification

The idea similar to that of CC-ICA has also been adopted by several earlier studies. Govindan et al. (1998) proposed applying ICA to ECG classification during atrial fibrillation. Four ICA networks were trained and each of them was constructed for one class of data. Then the feature vectors generated from the training data for each class were used to train a multiple layer perceptron. It was found that the use of ICA resulted in a significant reduction in the correlation. More recently, Kotani and Ozawa (2005) applied the ICA feature extraction method to the two cases, namely hand-written digits in the MNIST database and acoustic diagnosis for a compressor. During the process ICA is performed within each category. The experimental results showed that doing ICA within each category can extract more useful features for classification. Also, the components from ICA seem to be better than the components from PCA in terms of the recognition accuracy.

From the methodological point of view, the CC-ICA feature extraction method seems to be more reasonable than PCA and ICA for naïve Bayes classifier (Bressan and Vitria, 2002; Vitria et al., 2007). An underlying assumption of naïve Bayes classifier is that the features are independent with each other given the class label, while CC-ICA makes each feature as independent as possible for each class. It has been found that naïve Bayes classifier integrated with CC-ICA often outperforms naïve Bayes classifier with PCA/ICA (Fan and Poh, 2007). However, in some cases such as microarray data analysis where the sample size for each class is very small, the direct use of CC-ICA may not be feasible.

2.5 Methods for relaxing the strong independence assumption

A limitation of the ordinary ICA is its strong independence assumption, which is difficult to be satisfied by real-world data. To capture the dependence between the components, Hyvarinen et al. (2001a) proposed topographic independent component analysis (TICA) to find the higher-order correlation for the components by the correlation of energies. The correlation of energies is defined as

$$\operatorname{cov}\left(s_i^2, s_j^2\right) = E\{s_i^2 s_j^2\} - E\{s_i^2\}E\{s_j^2\} \neq 0 \tag{2.5}$$

if s_i and s_j are close in the topography. In the TICA model all s_i are independent with each other given their variances, which weaken the assumption of ICA.

Bach and Jordan (2002) proposed the tree-dependent component analysis (TCA), which is essentially a generalization of ICA by using tree-structured graphical model to weaken the independence assumption in ICA. In TCA, the topology of the

tree T is not fixed in advance. The linear transform matrix $\mathbf{W}$ can be found by minimizing the following contrast function with respect to $\mathbf{W}$ and T :

$$J(x, \mathbf{W}, \mathbf{T}) = I(s_1, \cdots, s_m) - \sum_{(u,v) \in T} I(s_u, s_v) \tag{2.6}$$

wher I represents the mutual information function. Equation (2.6) is a theoretical contrast function for TCA. To apply it to real cases, Bach and Jordan (2002) proposed three empirical contrast functions. When one of the two variables $\mathbf{W}$ and T is fixed, the minimization of contrast functions can be solved with respect to another variable. The model can find the tree-structured dependencies among multiple time series (Bach and Jordan, 2003a).

In their another study, Bach and Jordan (2003b) extended TCA by allowing the tree to be a forest, which can help to find "clusters" of components. It will let components be dependent within a cluster and independent between clusters. More recently, Kim and Choi (2006) applied TCA to gene clustering. Empirical comparisons of TCA with PCA and ICA show that the TCA-based clustering is more useful for grouping genes into biologically relevant clusters and for finding the underlying biological processes.

TICA and TCA have been compared by Meyer-Base et al. (2005) for the statistical analysis of fMRI data. It was found that both of them are able to identify signal components with high correlation to the fMRI stimulus and cluster the dependent components. Nevertheless, the complexity of TICA and TCA restricted their applications in practice.

2.6 Concluding comments

In this chapter, we provide a review of the use of ICA as a feature extraction technique in pattern classification. Different ICA-based feature extraction methods together with their applications are briefly summarized and assessed. Compared with other feature extraction methods for classification, the superiority of ICA lies in its ability in utilizing high-order statistics and its suitability for the non-Gaussian case. As a result, it has received increasing attention in different application areas.

Within the family of ICA-based feature extraction methods, it has been found that the direct ICA feature extraction received much attention because of its simplicity and effectiveness. Among the bulk of its applications, most previous studies are relevant to supervised classification. Although the direct ICA feature extraction method was adopted in many previous studies, the CC-ICA feature extraction method seems to have some theoretical strength when naïve Bayes classifier is used. It is therefore meaningful to carry out a comparative study among the three feature extraction methods, such as PCA, the direct ICA feature extraction method and the CC-ICA feature extraction method, for naïve Bayes classifier, which is the objective of Chapter 3.

Despite the strength of CC-ICA as a feature extraction method for naïve Bayes classifier, in some cases CC-ICA may not be directly applied because the dataset often has a small number of samples but a huge number of attributes. Provided that in the dataset one or more classes include very few samples, the implementation of CC-ICA may even become infeasible. It is therefore worthwhile to further investigate the issues relevant to the use of CC-ICA for naïve Bayes classification of small datasets

such as microarray data analysis. In addition, previous studies using naïve Bayes integrated with ICA or CC-ICA feature extraction method mainly dealt with the single-label classification problems. However, in real-world applications, multi-label classification has also been an important topic. As such, it is meaningful to investigate the use of ICA as a feature extraction method for multi-label naïve Bayes classification. Chapters 4 to 6 of this thesis aim to explore these issues.

CHAPTER 3 COMPARING PCA, ICA AND CC-ICA FOR NAÏVE BAYES

3.1 Introduction

Naïve Bayes classifier is a simple but effective Bayesian classifier built upon the strong assumption that different features are independent with each other (Langley et al., 1992). Classification is done by selecting the highest posterior of classification variable given a set of features. Despite its simplicity, it is competitive with other more sophisticated classifiers such as decision trees (Friedman et al., 1997). In addition, since it does not require structure learning, it is easier to construct and implement. Owing to these advantages, the naïve Bayes classifier has gained great popularity in solving different classification problems, e.g. Friedman et al. (1997). Nevertheless, a major limitation of the naïve Bayes classifier is that the real-world data may not satisfy the independence assumption among features. Domingos and Pazzani (1997) showed that naïve Bayes classifier still performed well even when there exists strong dependence among different features. However, it may not be optimal if the independence assumption is violated. In real-world applications, the prediction accuracy of naïve Bayes classifier could be highly sensitive to the correlated features.

Many approaches have been proposed to improve the performance of the naïve Bayes classifier. In general, these approaches can be divided into two groups (Fan and Poh, 2008). One attempts to relax the independence assumption of naïve

Bayes classifier, e.g. the methods described in Section 2.5. The other attempts to use certain preprocessing procedure, e.g. the direct ICA and CC-ICA feature extraction methods, to make the features as independent as possible. In the second line of research, Gupta (2004) found that PCA is very useful to improve the classification accuracy and reduce the computational complexity. Prasad (2004) applied the direct ICA feature extraction method and found that the performance of naïve Bayes classifier integrated with ICA performs better than C4.5 and IB1 integrated with ICA. Bressan and Vitria (2002) and Vitria et al. (2007) proposed the CC-ICA method to do feature extraction for the naïve Bayes classifier, and found that CC-ICA based naïve Bayes classifier outperforms the pure naïve Bayes classifier.

From the methodological point of view, the CC-ICA feature extraction method seems to be more suitable than PCA and ICA for naïve Bayes classifier (Bressan and Vitria, 2002). However, in some cases, particularly when the sample size for each class is very small, the application of CC-ICA may become infeasible. In addition, the difference between PCA and ICA used for the naïve Bayes classifier needs to be further investigated. It is therefore necessary to compare the alternative feature extraction methods for naïve Bayes classifier under different scenarios.

In this chapter, we first give a brief introduction to the naïve Bayes classifier and the PCA, ICA and CC-ICA. We have also described how to integrate them with the naïve Bayes classifier. Then we empirically compare PCA, ICA and CC-ICA as feature extraction methods for naïve Bayes classifier and present the results obtained.

3.2 Naïve Bayes classifier

3.2.1 Basic model

The naïve Bayes classifier, also called simple Bayesian classifier, is a classifier built upon the Bayes' theorem. It is essentially a simple Bayesian Network (BN) and particularly suitable for the case when the dimensionality of the inputs is high (Langley et al., 1992). Figure 3.1 shows the structure of the naïve Bayes classifier as a special case of BN.

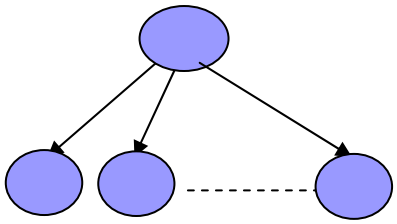


Fig. 3.1. Structure of naïve Bayes classifier

Assume that a set of samples $x_1, x_2, \cdots, x_K$ is given with their associated class labels $c_{x_1}, c_{x2}, \cdots, c_{x_K}$, where $c_{x_k} \in \Omega = \{c_1, c_2, \cdots, c_L\}$. Further assume that the samples have *n* features denoted as $z_1, z_2, \cdots, z_n$. The task is to use the samples to learn a naïve Bayes model that will predict the label c_x for any future sample x.

A general BN classifier, which uses the Bayes rule to compute the posterior of classification variable c based on the feature variables $z_1, z_2, \cdots, z_n$, can be described as follows:

$$p(c|z_1, z_2, \cdots, z_n) = \frac{p(z_1, z_2, \cdots, z_n|c)p(c)}{p(z_1, z_2, \cdots, z_n)} \tag{3.1}$$

In application, it is not practical to estimate the joint conditional probability $p(z_1, z_2, \cdots, z_n|c)$ in Eq. (3.1). A common practice is to simplify it as the naïve Bayes classifier by imposing two assumptions on Eq. (3.1). The first is the so-called class-conditional independence assumption, i.e. all features $z_1, z_2, \cdots, z_n$ are independent with each other given the classification variable c. Mathematically, it can be written as

$$p(z_1, z_2 \cdots, z_n|c) = p(z_1|c)\, p(z_2|c) \cdots p(z_n|c) \tag{3.2}$$

which means that the joint conditional probability is the product of all the marginal conditional probabilities. The second assumption is that all features $z_1, z_2, \cdots, z_n$ are directly dependent on the classification variable c. If the two assumptions are imposed on the general BN classifier, we can obtain the naïve Bayes classifier as follows:

$$p(c|z_1, z_2, \cdots, z_n) = \frac{p(c)\prod_{i=1}^{n} p(z_i|c)}{p(z_1, z_2, \cdots, z_n)} \tag{3.3}$$

In classification, the conditional probability of z_i given c (i.e. $p(z_i|c)$) and the prior of c (i.e. $p(c)$) can be obtained from the model learning process based on the given training dataset. In addition, since $p(z_1, z_2, \cdots, z_n)$ is common for a certain sample, it can be ignored in the classification process. As a result, we can derive the following model:

$$c = \arg\max_{c\in\Omega} p(c)\prod_{i=1}^{n} p(z_i|c) \tag{3.4}$$

which can be used to predict the class of each sample. In application, Eq. (3.4) is often replaced by its logarithmic form as follows:

$$c = \arg\max_{c\in\Omega}\{\log(p(c)) + \sum_{i=1}^{n}\log(p(z_i \mid c))\} \tag{3.5}$$

3.2.2 Dealing with numerical features for naïve Bayes

In pattern classification, continuous or numerical features are often involved. To use the naïve Bayes for classification, we often need to first model the density function of each continuous feature/variable. Many methods have been proposed and employed to model the density function of a continuous variable. According to Perez et al. (2009), these methods can be grouped into the following four categories:

a. Discretize the continuous variable and estimate its probability distribution .
b. Directly estimate the density function in a parametric manner based on certain distributional assumptions such as Gaussian distribution.
c. Directly estimate the density function in a non-parametric manner using the techniques such as kernel density estimator.
d. Directly estimate the density in a semi-parametric manner using models such as finite mixture model.

In the literature, a popular practice is to discretize the continuous variables and estimate their probability distributions, i.e. using approach (a). Many discretization methods have been developed and used for the naïve Bayes classifier. Yang and

Webb (2002) carried out a comparative study of nine discretization methods and found that the lazy discretization, nondisjoint discretization and weighted proportional k-interval discretization methods can help the naïve Bayes classifier achieve better classification performance. An advantage of approach (a) is its simplicity and ease of implementation. However, it may often result in the loss of information in the process of discretization (Perez et al., 2009).

The second approach, i.e. approach (b), attempts to directly estimate the density functions of the continuous variables using a parametric way. In most studies, the Gaussian function will be used to approximate the densities of many real-world data. However, the real-world data may not always follow the Gaussian distribution well. As such, researchers developed the semi-parametric and even non-parametric density estimation approach, i.e. approach (d) and (c), for use. Of the various non-parametric density estimation methods, kernel density estimation is the most popular one, which may provide a better approximation to complex distributions than the Gaussian parametric estimation approach. Therefore, in this study we choose the kernel density estimation for the learning of the naïve Bayes classifier. Another reason for choosing the kernel density estimation method is due to the inappropriateness of the Gaussian parametric density estimation method, as the new components obtained from the ICA-based feature extraction are non-Gaussian.

Mathematically, the kernel based *n*-dimensional estimator can be expressed as follows:

$$f\left(x;\mathbf{H}\right)=m^{-1}\sum_{i=1}^{m}K_{H}\left(x-x^{(i)}\right) \tag{3.6}$$

where $\mathbf{H}$ is a $n \times n$ bandwidth or smoothing matrix, $x = (x_1, x_2, \cdots, x_n)$ is a n-dimensional instantiation of X, m is the number of samples from which the estimator is learned, i is the index of a case in the training set, and $K_H(\cdot)$ is the kernel function used. A kernel density estimator is characterized by means of the kernel density K selected and the bandwidth matrix $\mathbf{H}$.

3.3 PCA, ICA and CC-ICA feature extraction methods

The strong class-conditional independence assumption underlying the naïve Bayes classifier is often not able to be satisfied by real-world data. Three popular feature extraction methods, namely PCA, ICA and CC-ICA, are often used to transform the original data so that in the transformed space the data may satisfy the assumption to some extent. Given a training dataset with features $x_1, x_2, \cdots, x_n$, PCA attempts to transform the original data into a new uncorrelated dataset (Haykin, 1999), while ICA/CC-ICA attempts to transform them into a new independent dataset with features $y_1, y_2, \cdots, y_n$ (Hyvärinen et al., 2001b).

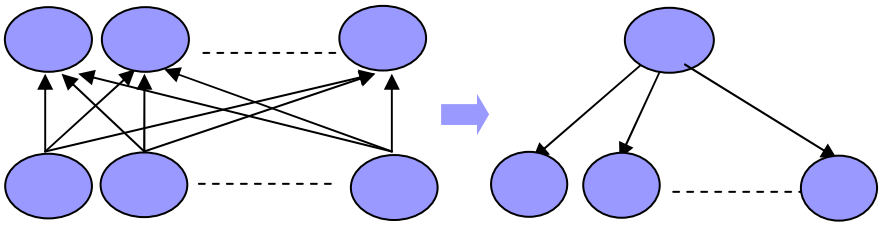


Fig. 3.2. Graphical illustration of PCA and ICA for naïve Bayes classifier

Figure 3.2 shows a graphical illustration of PCA and ICA used for the naïve Bayes classification. The left part of Fig. 3.2 provides a graphical representation of PCA and ICA, which is essentially a neural network. The graphical representation of

the naïve Bayes classifier, i.e. the right part of Fig. 3.2, is essentially a Bayesian network. The combination of PCA/ICA with naïve Bayes classifier links the neural network to the Bayesian network in a sequential way. The CC-ICA feature extraction method, proposed by Bressan and Vitria (2002), can be considered as an extension to the ICA feature extraction method. It is built upon the idea that ICA is used to make the new features as independent as possible within each class. In this way, the new features obtained from CC-ICA seem to be more reasonable than those from the PCA and ICA to satisfy the independence assumption of the naïve Bayes classifier. In the followings, we shall describe some technical features of the PCA, ICA and CC-ICA as well as their main differences.

3.3.1 Uncorrelatedness, independence and class-conditional independence

The differences among PCA, ICA and CC-ICA mainly come from the different concepts they are based on. PCA is based on the concept of uncorrelatedness, while ICA and CC-ICA are respectively based on the concepts of independence and class-conditional independence.

In statistics, two random variables, e.g. z_i and z_j, are said to be uncorrelated if their covariance is zero. Mathematically, the uncorrelatedness condition can be written as

$$E\{(z_i - \bar{z}_i)(z_j - \bar{z}_j)\} = 0 \tag{3.7}$$

where $\bar{z}_i$ and $\bar{z}_j$ are respectively the expected values of z_i and z_j. Eq. (3.7) is also equivalent to

$$E\{z_i z_j\} = E\{z_i\}E\{z_j\} = \bar{z}_i \bar{z}_j \tag{3.8}$$

In the case of multiple random variables, uncorrelatedness means that each pair of them are uncorrelated with each other.

Statistical independence is defined in terms of distribution functions or probability densities. Two random variables are independent with each other if knowing the value of one variable does not give any information on the value of the other. Mathematically, z_i and z_j are said to be independent if and only if

$$p(z_i, z_j) = p(z_i)p(z_j) \tag{3.9}$$

where $p(\cdot)$ denotes the density function of a random variable or the joint density function of a set of random variables. In the case of multivariate random variable $z = (z_1, z_2, \cdots, z_n)$, independence implies that

$$p(z) = p(z_1)p(z_2)\cdots p(z_n) \tag{3.10}$$

It should be pointed out that uncorrelatedness and independence have similarities while they are essentially different from each other. If two variables are independent with each other, they must be uncorrelated with each other. However, uncorrelatedness does not imply independence. Therefore, uncorrelatedness is a weaker form of independence.

Conditional independence is just a natural extension to the concept of independence through incorporating the conditional operator, i.e.

$$p(z_i, z_j \mid c) = p(z_i \mid c)p(z_j \mid c) \tag{3.11}$$

An equivalent form of Eq. (3.11) is

$$p(z_i \mid z_j, c) = p(z_i \mid c) \tag{3.12}$$

3.3.2 Principal component analysis

PCA is one of the most commonly used statistical techniques in data analysis. It deals with the transformation of a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. Usually, the first principal component accounts for as much of the variability in the data as possible. Each succeeding component accounts for as much of the remaining variability as possible. As a feature extraction technique, PCA can reduce the redundancy of the original features through extracting a smaller set of uncorrelated features from them.

Technically, a principal component can be defined as a linear combination of optimally-weighted observed features. Consider a data matrix $\mathbf{X} = (x_{ij})_{m \times n}$ where each column is considered as a feature variable with zero empirical mean. The PCA data transformation can be formulated as:

$$\mathbf{Y}^T = \mathbf{X}^T \mathbf{W} = \mathbf{V}\mathbf{\Sigma} \tag{3.13}$$

where $\mathbf{W}\sum\mathbf{V}^{\mathbf{T}}$ is the singular value decomposition of the data matrix $\mathbf{X}$. As such, finding the principal components is equivalent to finding the singular value decomposition of $\mathbf{X}$.

In practice, principal components can be derived by various algorithms such as covariance maximization, mean square error minimization and other on-line algorithms. The advantage of on-line algorithms is that the eigenvector estimates change in an incremental method without computing the covariance matrix at all. For the examples of PCA implementation, please refer to Smith (2002).

3.2.3 Independent component analysis

ICA is a relatively new statistical and computational technique for data analysis. It was initially proposed for solving the blind source separation problem, i.e. separating a multivariate signal into its additive subcomponents with the assumption of the mutual statistical independence of the non-Gaussian source signals. In feature extraction, ICA can extract a smaller set of approximately independent features with less redundancy from a set of original features.

The basic ICA model for feature transformation can be written as

$$\mathbf{Y}^T = \mathbf{W} \cdot \mathbf{X}^T \tag{3.14}$$

where $\mathbf{W}$ is a n by n de-mixing matrix, $\mathbf{X}$ is a k by n mixed matrix, and $\mathbf{Y}$ is a K by n source matrix. Every column of $\mathbf{Y}$ represents one "independent component" and all the columns consist of the new features for classification purpose. The main purpose of ICA is to estimate $\mathbf{W}$ and $\mathbf{Y}$.

There are several principles to solve the ICA model, such as maximum likelihood method, nongaussianity maximization, and mutual information minimization (Hyvärinen et al., 2001b). Each principle will generate a specific objective function and its optimization will enable the ICA estimation. Various algorithms may be used to solve the optimization problems, among which the fixed-point algorithm is a popular one.

In ICA, whitening is often performed by PCA before estimating the independent components. Whiteness means that the new variables (after PCA transformation) not only have zero-mean and unity-variance but also are uncorrelated with each other. The first step of whitening is to estimate the mean vector of the data matrix and to transform the original variables into a set of new variables with zero means. Then we can make them uncorrelated and have unit variance.

Once the process of data whitening is finished, we can apply the fix-point algorithm to estimate the transformation matrix and independent components for use. Here we only introduce the fixed-point algorithm based on the principle of mutual information minimization, which is used in the research work presented in this thesis. Other details on the fixed-point algorithm can be found in Hyvärinen et al. (2001b).

Suppose that the differential entropy H of a random vector $y = [y_1, y_2, \cdots, y_n]^T$ with density $f(\cdot)$ is defined as follows:

$$H(y) = -\int f(y) \log f(y) dy \tag{3.15}$$

Based on the differential entropy, we can define the negentropy J as

$$J(y) = H(y_{Gauss}) - H(y) \tag{3.16}$$

where y_{Gauss} is a Gaussian random vector with the same covariance matrix.

Mutual information, a measure of the dependence between random variables, is defined as follows:

$$I(y_1, y_2, \cdots, y_n) = J(y) - \sum_i J(y_i) \tag{3.17}$$

It can be shown that the mutual information measure is always nonnegative. It will be equal to zero if and only if the variables are statistically independent with each other. In addition, $J(y)$ does not depend on the de-mixing matrix $\mathbf{W}$.

Our task is to find the de-mixing matrix W that minimizes the mutual information. Since in Eq. (3.17) $J(y)$ can be considered a constant term, ICA estimation by minimizing mutual information is equivalent to maximizing the sum of negentropies of the independent components, i.e. $\sum_i J(y_i)$.

In computation, the negentropy of the independent component y_i can be approximately expressed as

$$J_G(y_i) \approx c[E\{G(y_i)\} - E\{G(\upsilon)\}]^2 \tag{3.18}$$

where G is practically any non-quadratic function, c is an irrelevant constant, and υ is a Gaussian variable with zero mean and unit variance. It should be pointed out that one may obtain more robust estimators if choosing G wisely. The study by Hyvärinen et al. (2001b) has provided several good candidate functions for G.

To find one independent component, we substitute $y_i = w_i^T x$ into Eq. (3.18) and obtain the following optimization problem:

$$\begin{aligned} &\text{maximize} \quad \sum_{i=1}^{n} J_G(w_i) = \left[E\{G(w_i^T x)\} - E\{G(v)\}\right]^2 \\ &\text{subject to} \quad E\left\{(w_i^T x)^2\right\} = 1 \qquad i = 1, \cdots, n \end{aligned} \tag{3.19}$$

One independent component can be estimated by solving the optimization problem through the simple FastICA algorithm. The basic form of the FastICA algorithm (Hyvärinen et al., 2001b) is described as follows:

Step 1. Centre the data to make its mean zero.

Step 2. Whiten the data to give new x (for convenience, we still use x to represent the whitened data).

Step 3. Choose an initial vector w of unit form.

Step 4. Let $w \leftarrow E\{xg(w^T x)\} - E\{g'(w^T x)\}w$, where g is the derivative of G.

Step 5. Let $w \leftarrow w / \|w\|$.

Step 6. If not converged, go back to Step 4.

The above algorithm only helps to estimate one independent component. In order to find more independent components, we need to run one-unit FastICA algorithm many times. However, after every iteration, the vectors obtained need to be decorrelated or orthogonalized. As discussed in Hyvärinen et al. (2001b), there are several methods that can be used to achieve decorrelation or orthogonalization, which

will not be described in this thesis. The details on the variants of FastICA algorithm for estimating more independent components can be found in Hyvärinen et al. (2001b). For the numerical and application examples of ICA implementation, please refer to Hyvärinen et al. (2001b) and Stone (2004).

3.2.4 Class-conditional independent component analysis

CC-ICA, proposed by Bressan and Vitria (2001), is built upon the idea that ICA is performed within each class so that one projection matrix will be obtained for each class. The new features obtained from CC-ICA may satisfy the class-conditional independence assumption of the naïve Bayes classifier better. In application, the usefulness of CC-ICA as a feature extraction technique for the naïve Bayes classifier has been empirically assessed by Bressan and Vitria (2001, 2002), and Vitria et al. (2007).

In implementation, the CC-ICA models are established and solved from the training set for each class. Assume that x^k, y^k and $\mathbf{W}^k$ are respectively the original features, the independent components and the de-mixing matrix for class k. The basic CC-ICA model can be written as

$$y^k = \mathbf{W}^k x^k, \quad k = 1,2,\cdots,K \tag{3.20}$$

Using the FastICA algorithm to solve the CC-ICA models, we can obtain the projection matrix $\mathbf{W}^k$ and the independent components $y_n^k (n = 1,2,\cdots,N)$ for class k. Then we can use the class-conditional independent components to establish the class-conditional naïve Bayes classifier for use.

Theoretically, $\mathbf{W}^k = \mathbf{B}^k(\mathbf{D}^k)^{-1/2}\mathbf{E}^k$ where $\mathbf{E}^k$ is the eigenvector matrix from PCA, $\mathbf{D}^k$ is the diagonal matrix with the corresponding eigenvalues λ_n^k $(n = 1,2,\cdots,N)$, and $\mathbf{B}^k$ is the ICA projection matrix for the whitened data for class k. Assume that the class-conditional representation of the original data provides independent components, the class-conditional probability in transformed space can be expressed as

$$p(x \mid c_k) = \alpha^k p(y^k \mid c_k) = \alpha^k \prod_{n=1}^{N} p(y_n^k \mid c_k) \tag{3.21}$$

where $\alpha^k = \left|\det((\mathbf{D}^k)^{-1/2})\right| = \prod_n 1/\sqrt{\lambda_n^k}$. Accordingly, the naïve Bayes classifier based on log-likelihoods can be reformulated as

$$c^* = \arg\max_{k\in\Omega} \sum_{n=1}^{N} \log p_k(y_n^k) + \log(\alpha^k) \tag{3.22}$$

The class-conditional marginal densities $p_k\left(y_n^k\right)$ can be estimated using various density estimation techniques such as the nonparametric kernel method described in Section 3.2.2. Despite the theoretical reasonableness of CC-ICA, its application may be restricted by the fact that the ICA learning usually requires a large number of samples, particularly for high-dimensional data. If the sample size is not large enough, the class-conditional representation obtained may not be trustable.

3.3 Empirical comparison results

A comparative study is carried out to empirically evaluate PCA, ICA and CC-ICA for naïve Bayes classifier. Three popular datasets are collected from the UCI

machine learning repository for our study. Since ICA is only applicable to continuous data, the features of the three datasets selected are all of continuous type. Table 3.1 shows the main characteristics of the three datasets used. Since the Yeast dataset has two features with many zero values and the sample size for six classes is not large enough for implementing CC-ICA, we also reduce the Yeast dataset to a smaller dataset, i.e. Yeast_1 as displayed in Table 3.1, by removing the two features and the samples for the six classes for our study use.

Table 3.1

UCI datasets with their specific characteristics

Dataset	Number of features	Number of classes	Number of instances	Remarks
Pima	8	2	768	To classify if a patient has Diabetes
Vehicle	18	4	946	To classify a given silhouette as one of four types of vehicle by 2D images
Yeast (Yeast_1)	8 (6)	10 (4)	1484 (1300)	To classify a given gene data as one of four types of yeast

These datasets are classified by pure naïve Bayes classifier (NB), the NB classifier integrated with the PCA feature extraction method (PCA+NB), the NB classifier integrated with the ICA feature extraction method (ICA+NB) and the NB classifier integrated with the CC-ICA feature extraction method (CC-ICA+NB), respectively. The FastICA algorithm is used to do ICA and CC-ICA estimations. Since a major assumption of ICA is that the distributions of the underlying independent components are non-Gaussian, it is not appropriate to use the parametric method to estimate their density functions. We therefore adopt the popular non-parametric kernel density estimation technique for use.

For each dataset, nine tenths of the data are randomly selected as the training data and the remaining one tenth of the data act as the testing data. Such a procedure is carried out for ten times for each classifier. We then use the classification results based on testing data to compare the performance of the four classifiers. Table 3.2 shows the means and the standard deviations of the accuracy rate under each scenario and the corresponding *p*-values (in brackets) for testing the difference between the naïve Bayes classifiers with certain feature extraction method and the pure naïve Bayes classifier.

Table 3.2

Experiment results of the UCI datasets

Dataset	Naïve Bayes	PCA+NB	ICA+NB	CC-ICA+NB
Pima	0.61±0.0354	0.63±0.0700 (0.4306)	0.66±0.0505 (0.0195)	0.68±0.0279 (0.0001)
Vehicle	0.62±0.0482	0.79±0.0424 (0.0000)	0.79±0.0295 (0.0000)	0.85±0.0396 (0.0000)
Yeast_1	0.56±0.0335	0.57±0.0422 (0.5646)	0.58±0.0310 (0.1828)	0.58±0.0346 (0.2056)
Yeast	0.31±0.027	0.528±0.0495 (0.0000)	0.53±0.0354 (0.0000)	-

Table 3.2 shows that all the feature extraction methods can improve the performance of naïve Bayes classifier to a certain degree. It is likely due to the fact that these feature extraction methods could weaken the dependence among different features. For the Pima and Vehicle datasets, the performance of the naïve Bayes classifier has been significantly improved by the use of feature extraction methods.

For Yeast_1datset, the performance has not been significantly improved. The possible reason is that some information may be deleted when some features are deledted.

It can be found from Table 3.2 that the performance of CC-ICA+NB is better than that of PCA+NB or ICA+NB. It indicates that CC-ICA could be the most appropriate feature extraction method for naïve Bayes classifier. The reason is that CC-ICA performs ICA for each class, which seems to be more reasonable for satisfying the class-conditional independence assumption of the naïve Bayes classifier. However, a limitation of the CC-ICA feature extraction method is that it cannot be implemented when the sample size in some classes is not large enough to do ICA, e.g. the Yeast dataset. In such cases, ICA+NB and PCA+NB are recommended since they still perform better than the pure naïve Bayes classifier.

Interestingly, Table 3.2 also shows that the discrepancy between ICA+NB and PCA+NB is not large. This may be an indication that PCA and ICA are competitive in improving the performance of naïve Bayes classifier. It results from their close relationship that ICA could be treated as a generalization of PCA. PCA tries to find uncorrelated variables, whereas ICA attempts to obtain statistically independent variables to represent the original multivariate data. Therefore, the dependence among features might be weakened at a similar level.

In order to investigate the relationship between the number of features and the performance of the classifiers, we reduce the number of features in the Vehicle dataset step by step and carry out the same experiments as described above. Fig. 3.3 shows the relationship between the average classification accuracy rate and the number of features.

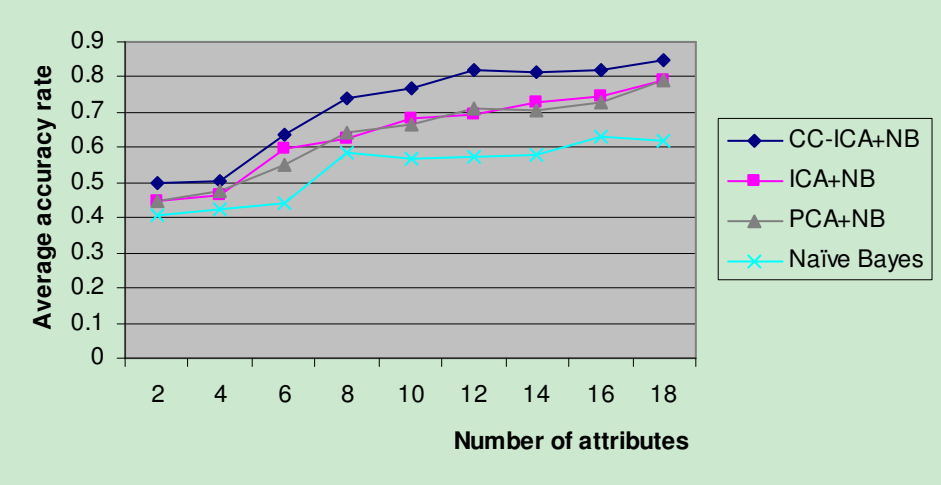


Fig. 3.3. Relationship between average accuracy rate and the number of features

It can be observed from Fig. 3.3 that all the three feature extraction methods are always effective in improving the performance of the naïve Bayes classifier. Compared with other classifiers, the performance of CC-ICA+NB seems to be the most promising. In most cases, ICA+NB and PCA+NB are competitive with each other. In the case of pure naïve Bayes classifier, its performance has almost no changes when the number of features becomes large (>7). However, with the increase of the number of features, the three feature extraction methods keep improving the performance of naïve Bayes classifier. One possible reason is that the dependence among features is enhanced when the number of features increases. For the pure naïve Bayes classifier, the information offered by the new features may be counteracted by the dependence enhanced. But for other classifiers, the feature extraction methods may extract more information while weakening the dependence. As a result, the feature extraction methods remain effective with the increase of the number of features.

It should be pointed out that our comparative study is only with regards to naïve Bayes classifier. Although it is meaningful to carry out a more comprehensive study by comparing PCA, ICA and CC-ICA for various classifiers, the main focus of our study is to assess the usefulness of ICA-based feature extraction methods for naïve Bayes classifier. As such, our empirical studies presented in this chapter as well as subsequent chapters do not include other types of classifiers.

3.4 Conclusion

In this chapter, we give an introduction to naïve Bayes classifier and PCA, ICA and CC-ICA feature extraction methods. Then we empirically compare the three alternative feature extraction methods for naïve Bayes classifier. Our experimental results have shown that all the three methods can improve the classification performance of naïve Bayes. When the size of features becomes larger, they could substantially improve the performance of the naïve Bayes classifier. In most cases, CC-ICA+NB outperforms PCA+NB and ICA+NB in terms of classification accuracy. However, CC-ICA requires more samples to ensure that there are enough training data for each class. When the sample size is much less than the number of the features, e.g. in the case of microarray data analysis, the implementation of CC-ICA may become infeasible. To overcome this limitation, in the next Chapter, we propose a CC-ICA based sequential feature extraction approach for naïve Bayes classification of microarray data.

CHAPTER 4 A SEQUENTIAL FEATURE EXTRACTION APPROACH FOR NAÏVE BAYES CLASSIFICATION OF MICROARRAY DATA

4.1 Introduction

As mentioned in Chapter 3, naïve Bayes classifier is a simple Bayesian network classifier built upon the strong assumption that different attributes are independent with each other given the class (Friedman et al., 1997; Gurwicz and Lerner, 2005). Despite its simplicity, naïve Bayes classifier has been found to be surprisingly effective compared with other more sophisticated classifiers (Hall, 2007). It is therefore not surprising that naïve Bayes classifier has gained popularity in solving various classification problems including microarray data analysis, e.g. Sandberg et al. (2001) and Kelemen et al. (2003).

Nevertheless, there exist two major limitations that may severely affect the successful application of naïve Bayes classifier to microarray data analysis. The first is the class-conditional independence assumption embedded in the classifier itself, which is hardly satisfied by the microarray data. This limitation could be, at least theoretically, overcome by the CC-ICA technique proposed by Bressan and Vitria (2002). The experimental results of our comparative study presented in Chapter 3 have shown that CC-ICA could effectively improve the performance of naïve Bayes classifier in some application domains.

Another limitation comes from the intrinsic characteristics of microarray dataset, which usually consists of thousands of genes with only tens of samples due to the expensive experiment. The extremely high dimensionality of microarray data may greatly increase the computational costs of naïve Bayes classifier. In addition, since the sample size is far smaller than the gene size, the use of CC-ICA can hardly enhance the independence among genes as well as improve the performance of naïve Bayes classifier. When the sample size in some classes is not large enough to do ICA, the implementation of CC-ICA even becomes infeasible (Fan and Poh, 2007). It is therefore necessary to do feature selection to reduce the dimensionality of genes before applying CC-ICA for naïve Bayes classification of microarray data.

In this chapter, we propose a CC-ICA based sequential feature extraction approach for naïve Bayes classification of microarray data. Section 4.2 gives a brief introduction to microarray data analysis. In Section 4.3, we present the sequential feature extraction approach for naïve Bayes classifier, which includes feature selection by stepwise regression and feature transformation by CC-ICA. Section 4.4 presents the experimental results on five commonly used microarray datasets, which show that the proposed approach can not only improve the average classification accuracy rates but also reduce the variation of classification performance. Section 4.5 concludes this chapter.

4.2 Microarray data analysis

Recent advancements in DNA microarray technology have enabled people to monitor and measure the expression levels of hundreds of thousands of genes

simultaneously, which allowed a great deal of microarray data to be generated. Technically, microarray data can be collected with the help of various technologies, such as the spotted cDNA and GeneChips. Spotted cDNA microarrays are microchips with more than ten thousands of spots that correspond to a unique gene per condition. GeneChips are silicon chips for measuring the expression levels of thousands of genes simultaneously.

Gene measurements of microarray data may provide insights into biological processes, which are very helpful to cancer prediction and diagnosis. As such, researchers have applied mathematical models and computational tools to capture the underlying characteristics of microarray dataset. Broadly speaking, the approaches for microarray data analysis can be classified into two groups, namely supervised and unsupervised approaches. Unsupervised approaches are mainly used for discovering novel biological mechanisms and revealing genetic regulatory networks.

Supervised approaches mainly deal with the identification of gene expression patterns specific to each class, and the class prediction of new samples. Different methods, from simple statistical techniques such as linear regression to complex machine learning algorithms such as support vector machines, have been employed to select informative genes and do classification of microarray data. Examples of such studies include Guyon et al. (2002), Huang and Pan (2003), Kim and Cho (2004, 2006), Chen (2006), Zheng et al. (2006) and Park et al. (2007). As a simple but useful classifier, the applicability of naïve Bayes in microarray data analysis has also been explored in many previous studies including Sandberg et al. (2001) and Kelemen et al. (2003). However, since microarray data usually has a small sample size but a huge

number of genes, feature extraction of microarray data must be done before the naïve Bayes classification of microarray data. In the next section, we shall introduce a CC-ICA based feature extraction approach for this prupose. Before that, we will describe the mathematical symbols of microarray data that will be used later.

Assume that there are K samples and M genes (usually $K << M$), and the expression level of gene i for sample k is x_{ki} . Let $x_k = (x_{k1}, x_{k2}, \cdots, x_{kM})$ and $\mathbf{X} = (x_{ki})_{K \times M}$ respectively denote the gene expression profile of sample k and the summarized microarray data matrix. Let g_i $(i = 1,2,\cdots,M)$ denote the variable representing gene i. Further assume that the class label of sample k is c_k where $c_k \in \Omega = \{1,2,\cdots,L\}$. Let c and $\mathbf{C} = (c_1, c_2, \cdots, c_K)^T$ respectively denote the class variable and the column vector of class labels for the K samples. The purpose is to train a naïve Bayes classifier based on $\mathbf{X}$ and $\mathbf{C}$, which may be used to accurately classify a given test sample with unknown class labels.

4.3 Sequential feature extraction approach

One specific characteristic of microarray data is that its feature (gene) size is far larger than sample size, which is known as “the curse of dimensionality problem”. It is therefore necessary to do feature selection on the original dataset. Effective feature selection can reduce the complexity in computation, increase the classification accuracy and enhance the generalization property of classifiers (Ding and Peng, 2005).

A number of methods have been developed and applied to do feature selection. A relatively comprehensive overview on alternative feature selection methods can be

found in Guyon and Elisseeff (2003). Feature selection algorithms typically fall into two categories: feature ranking and subset selection. Feature ranking evaluates the features by a metric and eliminates all features that do not achieve an adequate score. Subset selection searches the set of possible features for the optimal subset. In general, subset selection methods also can be divided into two big categories, namely filtering approach and wrapper approach. In microarray data analysis, filtering approach seems to be more popular. Although many filtering methods focus on the rankings of individual genes in terms of their relevance with class variable, recent studies have shown that the methods following the "minimum redundancy - maximum relevance" principle may select more representative genes (Ding and Peng, 2005; Park et al., 2007). Stepwise regression is just a simple statistical technique that follows the "minimum redundancy - maximum relevance" principle for the feature selection of the microarray data (Park et al., 2007).

In addition, ICA could transform the features as independent as possible to make them suitable for the assumption of naïve Bayes classifier. Especially for multi-class datasets, CC-ICA could transform the features for each class to make them more suitable for the assumption of naïve Bayes classifier. As such, our sequential feature extraction approach consists of two steps: stepwise regression-based feature selection and CC-ICA based feature transformation.

4.3.1 Stepwise regression-based feature selection

Stepwise regression is an automatic statistical procedure for selecting the representative predictive variables, e.g., genes in microarray data, to build good

regression models. It iteratively constructs a sequence of regression models by adding or removing variables at each step. In implementation, stepwise regression consists of three methods, namely forward selection, backward elimination, and forward-backward mix. At each step, forward selection adds the most statistically significant variable and backward selection deletes the least significant variable provided that the *p*-values for the two variables are respectively less than p_{in} and larger than p_{out}, where p_{in} and p_{out} are the probabilities of Type I error related to entering and deleting a variable.

Conceptually, stepwise regression also follows the "minimum redundancy - maximum relevance" principle as adopted by several recently proposed feature selection methods. Meanwhile, it is simple and easy to implement but has still good performance (Park et al., 2007). This feature is consistent with the "simple but competitive with some more complicated classifiers" feature of the naïve Bayes classifier. Therefore, we propose the use of stepwise regression rather than other methods for gene selection in this chapter, which could keep the simplicity of the naïve Bayes classifier. The procedures for performing forward selection and backward elimination are given below.

Forward Selection:

Step 1. Build regression models with only one predictor, and choose the one with the most statistically significant gene.

Step 2. Compute the *p*-values for all remaining predictors and choose the gene *j* with the smallest *p*-value.

Step 3. If the *p*-value for gene *j* is less than p_{in}, include the gene in the regression model. Go back to Step 2.

Step 4. Else stop and select the model with all the entered genes.

Backward Elimination:

Step 1. Build a regression model with all the M genes.

Step 2. Compute the p-values for all the predictors and choose the gene j with the largest p-value.

Step 3. If the p-value for gene j is greater than p_{out}, remove the gene from the regression model. Go back to Step 2.

Step 4. Else stop and select the model containing all the genes that were not eliminated.

The forward-backward mix procedure is a combination of forward selection and backward elimination. It starts with forward selection by adding a predictor to the model, which is followed by an examination of the predictors that were included previously to check if any predictor needs to be eliminated. During the process, p_{in} and p_{out} are still taken as criteria for examining whether a predictor should be included or removed. The procedure continues until no genes can be added or removed from the model.

In terms of the determination of p_{in} and p_{out}, a rule of thumb is to let them be small enough so that the number of genes selected is less than the number of samples (in order to do CC-ICA effectively). Without loss of generality, we assume that only the first N ($N<K$) genes are retained after stepwise regression-based feature selection. The microarray data matrix after feature selection is denoted by $\mathbf{Y}$ where $\mathbf{Y} = (g_1, g_2, \cdots, g_N) = (x_{ki})_{K\times N}$.

4.3.2 CC-ICA based feature transformation

Compared to PCA that attempts to transform these variables into a set of uncorrelated variables, ICA attempts to transform them into new variables that are mutually independent or as independent as possible with each other. It is therefore a more powerful technique that has been widely applied for feature transformation in different application areas such as time series forecasting, image processing, and microarray data analysis.

Given the microarray data matrix $\mathbf{Y}$, the basic ICA model for feature transformation can be written as

$$\mathbf{Z}^T = \mathbf{W} \cdot \mathbf{Y}^T \tag{4.1}$$

where $\mathbf{W}$ is a N by N de-mixing matrix and $\mathbf{Z}$ is a K by N source matrix. Every column of $\mathbf{Z}$ represents one "independent component" and all the columns consist of the new features for classification purpose. The task is to estimate $\mathbf{W}$ and $\mathbf{Z}$. As mentioned in Chapter 3, there are many principles and algorithms for performing the task. We here adopt the FastICA algorithm, which has been widely accepted as a computationally highly efficient method, to estimate $\mathbf{W}$ and $\mathbf{Z}$.

CC-ICA is built upon the idea that ICA is done within each class so that one mixing matrix can be obtained for each class (Vitria et al., 2007). In this way, the new attributes after transformation may satisfy the class-conditional independence assumption of the naïve Bayes classifier well. If we split the microarray data matrix

$\mathbf{Y}$ into a set of sub-matrices $\mathbf{Y}_l$ $(l = 1,2,\cdots,L)$ according to the class label, the set of models for doing CC-ICA can be written as

$$\mathbf{Z}_l^T = \mathbf{W}_l \cdot \mathbf{Y}_l^T \tag{4.2}$$

where $\mathbf{W}_l$ is a N by N mixing matrix and $\mathbf{Z}_l$ is a K_l by N source matrix for class l. Similarly, we can still use FastICA algorithm to estimate $\mathbf{W}_l$ and $\mathbf{Z}_l$ for each class.

4.4 Naïve Bayes classification of microarray data

We shall use the data after feature extraction, i.e. $\mathbf{Z}_l (l = 1,2,\cdots,L)$, to build a naïve Bayes classifier, which is used to classify a new test sample with gene values $z_1^t, z_2^t, \cdots, z_N^t$ (after ICA or CC-ICA based feature transformation). In general, Bayesian network classifier computes the posterior probability that the test sample belongs to class c by using the Bayes rule as follows:

$$p(c \left| z_1^t, z_2^t, \cdots, z_N^t \right.) = \frac{p(z_1^t, z_2^t, \cdots, z_N^t \left| c \right.) p(c)}{p(z_1^t, z_2^t, \cdots, z_n^t)} \tag{4.3}$$

By assuming that the class-conditional independence among genes in the ICA space is approximately satisfied, we obtain the following naïve Bayes classifier:

$$p(c \left| z_1^t, z_2^t, \cdots, z_N^t \right.) = \frac{p(c) \prod_{i=1}^{N} p(z_i^t \left| c \right.)}{p(z_1^t, z_2^t, \cdots, z_N^t)} \tag{4.4}$$

Since $p(z_1^t, z_2^t, \cdots, z_N^t)$ is a common factor for the testing sample, it can be ignored in classification process. In addition, since the gene values are of continuous type, we can use the probability density value $f(z_i^t|c)$ to replace the probability value $p(z_i^t|c)$. The class-conditional probability density $f(\cdot|c)$ for each gene can be estimated using the nonparametric kernel density estimation method (Perez et al., 2009). Meanwhile, the prior $p(c)$ can be obtained from the learning process. Finally, the following naïve Bayes classification model is derived:

$$c^* = \arg\max_{c\in\Omega}\{\log(p(c)) + \sum_{i=1}^{N}\log(f(z_i^t|c))\} \tag{4.5}$$

4.5 Experimental results

We evaluate the performance of the sequential feature extraction approach for naïve Bayes classifier based on five well-known gene expression datasets, namely Leukemia-ALLAML, Leukemia-MLL, Colon Tumor, Lung Cancer I and Lung Cancer II. Table 4.1 shows the five datasets with their characteristics. In addition to feature selection integrated with CC-ICA plus naïve Bayes classifier (FS+CCICA+NB), we also implement three other classification rules, namely naïve Bayes classifier (NB), feature selection plus naïve Bayes classifier (FS+NB), and feature selection integrated with ICA plus naïve Bayes classifier (FS+ICA+NB) on the five datasets. Here feature selection is performed through the stepwise regression approach. Since the proposed sequential feature extraction approach aims to address the issues arising from naïve Bayes classification of microarray data, its integrations

with other popular classifiers such as support vector machines are not considered in our experimental study.

Table 4.1

Summary of five microarray datasets

Dataset	Leukemia-ALLAML	Leukemia-MLL	Colon Tumor	Lung Cancer I	Lung Cancer II
Data source	Golub et al. (1999)	Armstrong et al. (2002)	Alon et al. (1999)	Bhattacharjee et al. (2001)	Gordon et al. (2002)
Number of attributes	7129	12528	2000	12600	12533
Number of classes	2	3	2	5	2
Number of instances	62	72	62	203	181

In our experiments, both leave-one-out and hold-out classification accuracy rates are used to give a relatively comprehensive comparison on the performances of alternative classification rules. Every dataset is partitioned into two parts, i.e. training and test datasets. The training dataset is used to do feature selection, carry out ICA/CC-ICA computation and train classifiers. The test dataset is used to evaluate the performances of alternative classifiers. The whole procedure for our experimental study, which includes model learning and testing, is described as follows:

Step 1. Split the data into training data $\mathbf{X} = (x_{ki})_{K\times M}$ and test data $\mathbf{X}' = (x'_{si})_{S\times M}$

Step 2. For training data $\mathbf{X} = (x_{ki})_{K\times M}$, do

a. Determine the initial values of p_{in} and p_{out} for stepwise regression.

b. Do feature selection by stepwise regression till the number of features is less than the number of samples (by modifying p_{in} and p_{out}). The new dataset is denoted as $\mathbf{Y} = (g_1, g_2, \cdots, g_N) = (x_{ki})_{K \times N}$.

c. Do feature extraction by CC-ICA (or ICA) from the transformation matrix W and obtain the new dataset $\mathbf{Z} = (z_1, z_2, \cdots, z_N)$.

d. Learning the naïve Bayes classifier *C* from $\mathbf{Z} = (z_1, z_2, \cdots, z_N)$.

Step 3. For the test data $\mathbf{X}' = (x'_{si})_{S \times M}$

a. Select the new features corresponding to the same features selected from the training data. The new dataset is denoted as $\mathbf{Y}' = (g'_1, g'_2, \cdots, g'_N) = (x'_{ki})_{K \times N}$.

b. Transform the new dataset $\mathbf{Y}'$ by same transformation matrix $\mathbf{W}$ to a new dataset $\mathbf{Z}'$.

c. Classify the new dataset $\mathbf{Z}'$ by the naïve Bayes classifier *C*.

For leave-one-out experiments, the pure naïve Bayes classifier was not included due to its extremely time-consuming computations. The classification accuracy rates for the other three classifiers are displayed in Table 4.2. Each sample was used to leave out once for measuring the accuracy rate. It can be seen from Table 4.2 that both FS+CCICA+NB and FS+ICA+NB perform better than FS+NB in microarray data analysis, which demonstrates the effectiveness of the proposed approach. As for the comparison between the former two classification rules, FS+CCICA+NB performs obviously better than FS+ICA+NB in terms of classification accuracy.

Since leave-one-out classification accuracy rates cannot provide the information on the variation of classification performance, we applied holdout classification accuracy rates to further evaluate the performances of alternative classification rules. In our experiment, four fifth of the samples are randomly selected as the training data and the remaining one fifth of the samples are taken as the test data. Such a procedure is repeated ten times for each classification rule on the four datasets exclusive of the Lung Cancer I dataset, which is due to the fact that some classes in the training data for this dataset have not enough samples to implement CC-ICA.

Table 4.2

Classification accuracy rates (%) of three classification rules on five datasets

Dataset	Leukemia-ALLAML	Leukemia-MLL	Colon Tumor	Lung Cancer I	Lung Cancer II
FS+NB	66.7	43.1	74.2	78.8	87.8
FS+ICA+NB	88.9	77.8	80.6	80.8	92.8
FS+CCICA+NB	95.8	83.3	82.3	82.3	98.3

Figures 4.1 to 4.4 show the boxplots of the holdout classification accuracy rates for the four classification rules on the four datasets. It can be found that FS+CCICA+NB and FS+ICA+NB have better classification performances than FS+NB or NB, which is consistent with the leave-one-out classification results. Feature selection by stepwise regression could improve the naïve Bayes classification accuracy rates, whereas the degree of performance improvement depends on the dataset. For instance, as shown in Fig. 4.1 and Fig. 4.2, the discrepancy between NB and FS+NB is not obvious for the first two datasets. However, for the last two

datasets the classification performance of FS+NB is significantly better than that of NB. Although feature selection may not always be effective, its integration with ICA/CC-ICA transformation has been found to certainly improve the classification performance.

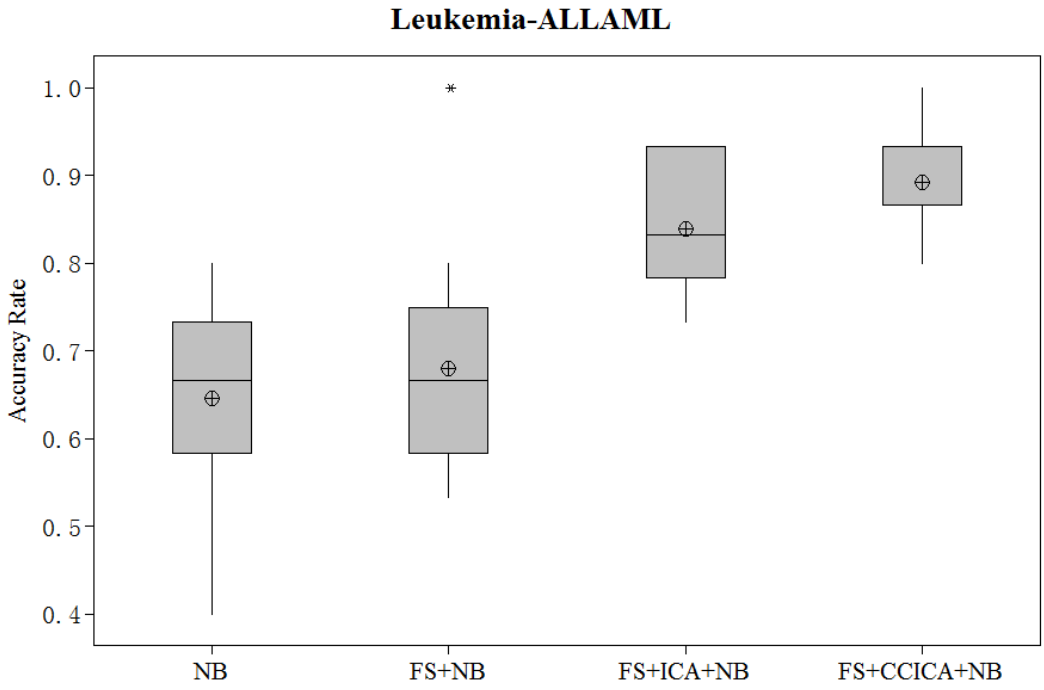


Fig. 4.1. Boxplots of the holdout classification accuracy rates for Leukemia-ALLAML

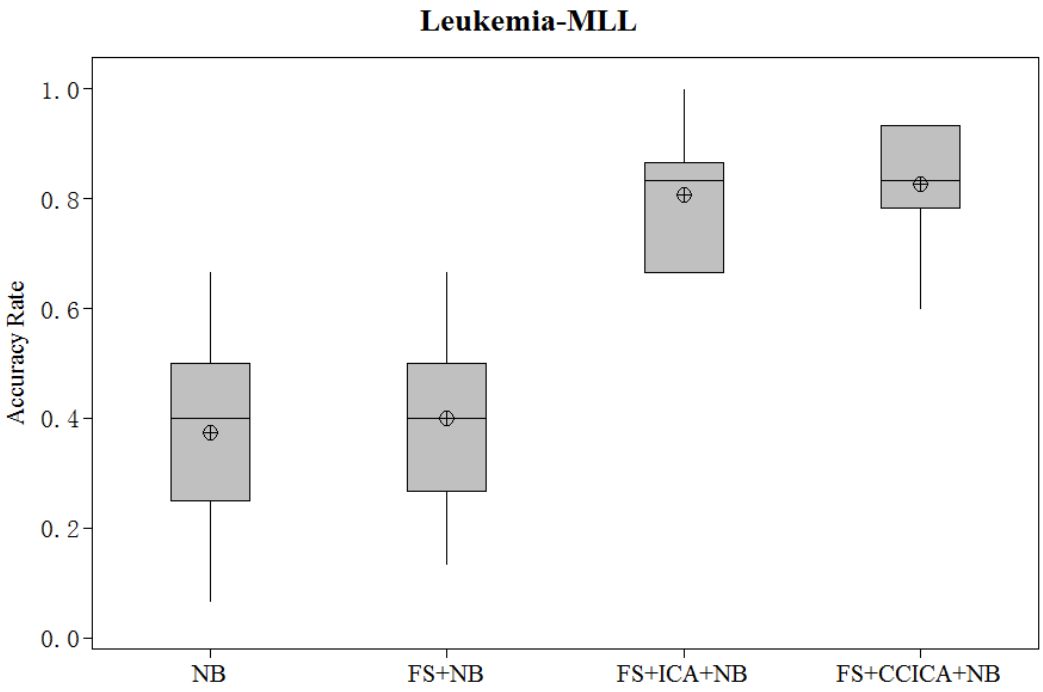


Fig. 4.2. Boxplots of the holdout classification accuracy rates for Leukemia-MLL

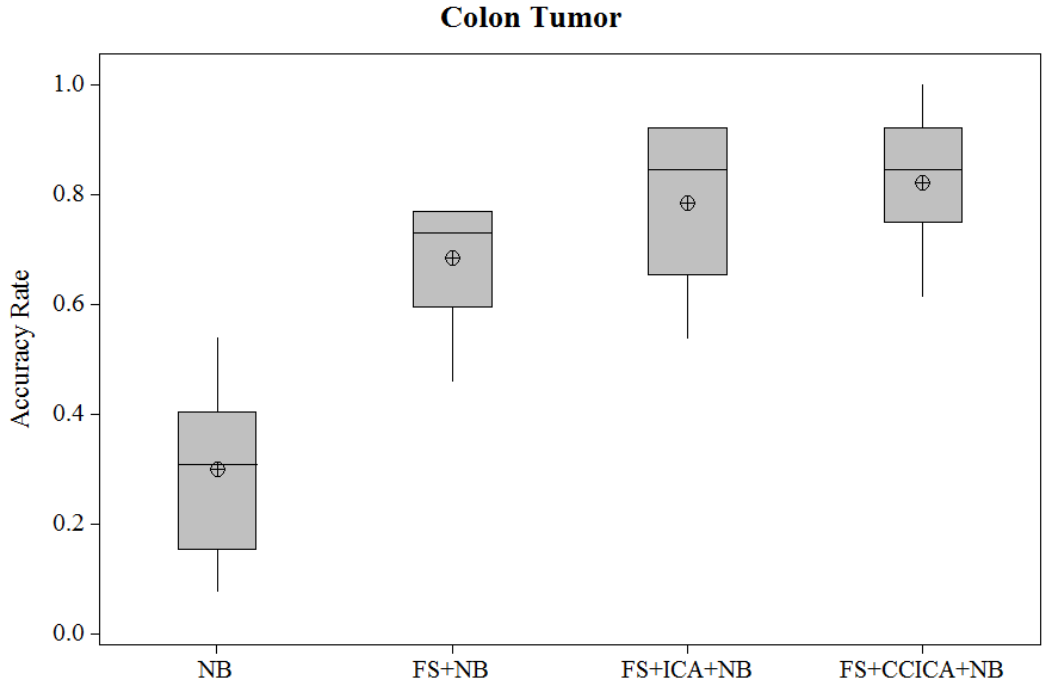


Fig. 4.3. Boxplots of the holdout classification accuracy rates for Colon Tumor

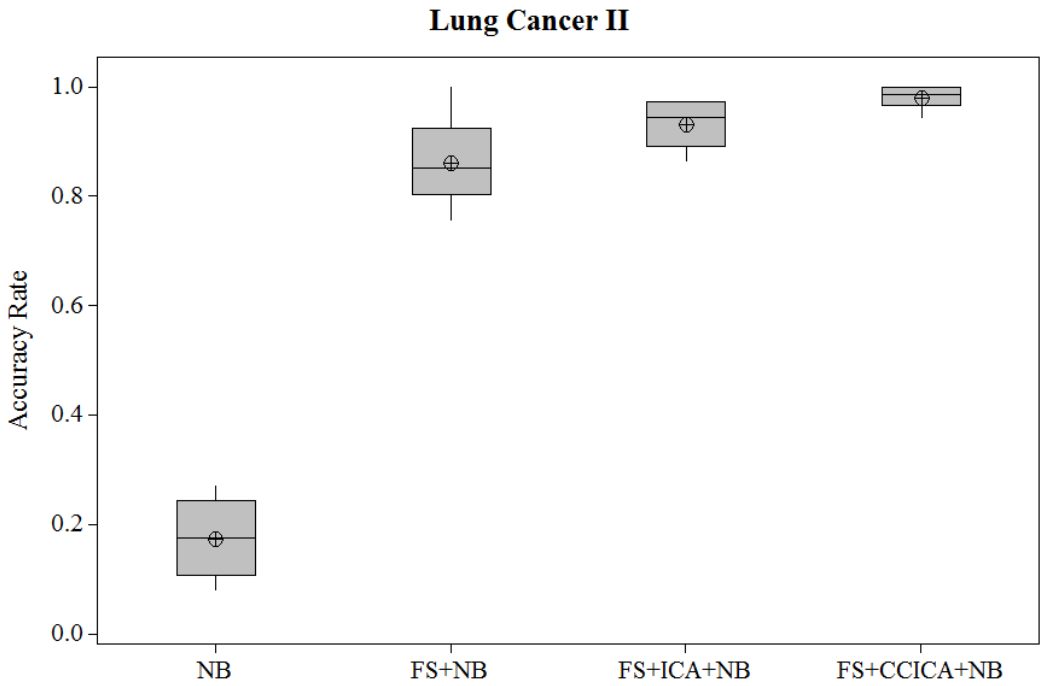


Fig. 4.4. Boxplots of the holdout classification accuracy rates for Lung Cancer II

It can be seen from Figs. 4.1 to 4.4 that the FS+CCICA+NB is generally superior to the FS+ICA+NB in the sense that the former is more stable than the latter in terms of classification performance. The possible reason is that feature transformation by CC-ICA seems to be more reasonable for the data to satisfy the class-conditional independence assumption underlying naïve Bayes classifier. However, a limitation of FS+CCICA+NB is that it may not be implemented when the sample size in some classes is too small to do ICA for each class. In such cases, FS+ICA+NB is recommended for use since it still performs better than NB and FS+NB.

4.6 Conclusion

In this chapter, we present a sequential feature extraction approach for naïve Bayes classification of microarray data. The feature extraction approach proposed starts from gene selection by stepwise regression, which is a simple but effective dimension reduction technique following the "minimum redundancy - maximum relevance" principle. The data on the genes selected are then transformed by CC-ICA, which makes the new features after transformation become as independent as possible. Our experimental results on five microarray datasets demonstrate the effectiveness of the sequential feature extraction approach in improving the classification performance of naïve Bayes classifier in microarray data analysis.

Our experimental study has also shown that CC-ICA seems to be very useful in improving the performance of naïve Bayes classifier by increasing its classification accuracy rate and reducing its standard deviation in microarray data analysis. However, it also shows that when the sample size for some classes is not large enough (e.g. the Lung Cancer I dataset), the implementation of CC-ICA becomes infeasible. To address this limitation, we present a partition-conditional ICA approach for naïve Bayes classification of microarray data, which will be described in the next chapter.

CHAPTER 5 PARTITION-CONDITIONAL ICA FOR BAYES CLASSIFICATION OF MICROARRAY DATA

5.1 Introduction

In the last chapter, we present a sequential feature extraction approach by combining stepwise regression and CC-ICA for naïve Bayes classification of microarray data. Despite the usefulness of the sequential approach in improving the performance of naïve Bayes classifier, the application of CC-ICA may be restricted when the sample sizes for some classes are too small. For instance, in microarray data analysis with multiple classes, a certain class may have only several samples. As a result, it becomes infeasible to do ICA estimation for the class.

To make use of the strengths of ICA and CC-ICA, in this chapter we propose partition-conditional independent component analysis (PC-ICA) for naïve Bayes classification of microarray data (Fan et al., 2010). Conceptually, PC-ICA attempts to implement ICA within each partition that may consist of several classes. A feature of PC-ICA is that the ICA and CC-ICA feature extraction methods can be considered as special cases of PC-ICA. As such, PC-ICA may represent an in-between concept compared to ICA and CC-ICA. Its usefulness in application is demonstrated by our experiments on two microarray datasets presented in this chapter.

In the following, we first introduce an alternative feature selection method based on mutual information, which also follows the "minimum redundancy maximum relevance" (MRMR) principle as done by stepwise regression. Then we

present PC-ICA as a feature extraction technique for naïve Bayes classification of microarray data. Finally, we present our experimental results on two microarray datasets, which demonstrate the effectiveness of PC-ICA.

5.2 Feature selection based on mutual information

Microarray data usually has a small number of samples but a huge large number of genes (features). This phenomenon is known as the "curse of dimensionality" problem in pattern classification. Therefore, feature selection is usually indispensable in microarray data analysis. Effective feature selection can help to reduce the complexity in computation, increase the classification accuracy and enhance the generalization property of classifiers (Ding and Peng, 2005).

A number of feature selection methods have been developed and employed in earlier studies. The study by Guyon and Elisseeff (2003) provides a relatively comprehensive review of various feature selection methods. As mentioned in last chapter, feature selection methods can be roughly divided into two categories, namely filtering approach and wrapper approach. In the line of filtering approach, most methods attempt to select the individual genes with the highest relevance to the class variable. Despite the usefulness of the "maximum relevance" criterion, the redundancy among selected features based on "maximum relevance" criterion may lead to poor classification performance (Jain et al., 2000). Therefore, some researchers have suggested to use the MRMR criterion to do feature selection (Peng et al., 2005). In microarray data analysis, several recent studies have shown that feature selection following the MRMR principle may likely help select more informative genes (Ding and Peng, 2005; Park et al., 2007; Fan et al., 2009). The stepwise

regression procedure used in Chapter 4 is also a feature selection method that follows the MRMR principle.

Despite the usefulness of stepwise regression based feature selection, its limitation in microarray data analysis is that the class variables are often of categorical type. In this chapter, we follow Peng et al. (2005) and choose mutual information as the measure of redundancy and relevance. In addition to its suitableness for categorical class variables, the mutual information measure is capable of quantifying the dependence between two features without assuming their distributions.

Let $c \in \Omega = \{1,2,\cdots,L\}$ and $g_1, g_2, \cdots, g_M$ respectively denote classification variable and the M features. Let $x_k = (x_{k1}, x_{k2}, \cdots, x_{kM})$ and $\mathbf{X} = (x_{ki})_{K\times M}$ respectively denote the gene expression profile of sample k and the summarized microarray data matrix. Assume that the joint probability distribution of two features g_i and g_j is $p(g_i, g_j)$ and their respective marginal probability distribution are $p(g_i)$ and $p(g_j)$. The mutual information of the two features can be defined as

$$I(g_i, g_j) = \sum_{l,m} p(x_{li}, x_{mj}) \log \frac{p(x_{li}, x_{mj})}{p(x_{li})\, p(x_{mj})} \tag{5.1}$$

The mutual information $I(g_i, g_j)$ can be used to quantify the similarity between genes g_i and g_j. The idea of minimum redundancy is to select the genes that have maximal dissimilarities. Mathematically, the minimum redundancy condition can be formulated as

$$\min W_I = \frac{1}{|S|^2} \sum_{i,j \in S} I(g_i, g_j) \tag{5.2}$$

where S denotes the subset of features with minimum redundancy and $|S|$ is the number of features in S.

The maximum relevance criterion attempts to capture the genes which have the maximal relevance with class variable c. Similar to Eq. (5.2), the maximum relevance condition can be written as

$$\max V_I = \frac{1}{|S|} \sum_{i \in S} I(c, g_i) \tag{5.3}$$

The MRMR feature selection based on mutual information attempts to optimize Eqs. (5.2) and (5.3) simultaneously. As Ding and Peng (2005) suggested, this can be done by aggregating the two criterion functions into a single criterion function. If the "minimum redundancy" and "maximum relevance" conditions are assumed to be equally important, we can define the single MRMR criterion functions as follows:

$$\max(V_I - W_I) \tag{5.4}$$

$$\max(V_I / W_I) \tag{5.5}$$

In Ding and Peng (2005), the criterion used in Eq. (5.4) is termed as mutual information difference criterion, while that in Eq. (5.5) is termed as mutual information quotient criterion. Then we can apply the heuristic algorithm given by

Ding and Peng (2005) to solve the MRMR optimization problem, which is briefly described below.

Step 1. Select the first gene based on Eq. (5.3), and define m=1.

Step 2. Currently, m genes have been selected. If m is equal to the number of genes required, stop. Otherwise, go to Step 3.

Step 3. Add an additional feature from the set of $\Omega_S = \Omega - S$ (i.e. all genes except those already selected) by solving the following MRMR conditions

$$\max_{i \in \Omega_S} \left\{ I(c, g_i) - \frac{1}{|S|} \sum_{j \in S} I(g_i, g_j) \right\} \tag{5.6}$$

or $$\max_{i \in \Omega_S} \left\{ I(c, g_i) \Big/ \left[\frac{1}{|S|} \sum_{j \in S} I(g_i, g_j) \right] \right\} \tag{5.7}$$

Step 4. Let m=m+1 and go to Step 2.

Note that the MRMR feature selection based on mutual information requires the estimation of mutual information. Although the mutual information estimation for discrete variables is straightforward, it is difficult to calculate the mutual information between continuous genes. A commonly adopted practice is to discretize the continuous genes first and then estimate the mutual information from Eq. (5.1) (Peng et al. 2005). In our experimental study, we also adopt this method for use.

5.3 PC-ICA for naïve Bayes classifier

The MRMR feature selection based on mutual information is capable of reducing the dimensionality of genes, which decreases the computational cost of using naïve Bayes to classify microarray data. However, since the conditional independence

assumption is hardly satisfied by the set of microarray data selected, the performance of naïve Bayes classifier may not be satisfactory. Previous studies including our work presented in Chapters 3 and 4 have found that CC-ICA may be an effective feature extraction method for improving naïve Bayes classifier in microarray data analysis. However, when some classes have only a small number of samples, the application of CC-ICA may become infeasible. Therefore, we extend CC-ICA and present the following PC-ICA method for use. Before introducing PC-ICA, we first give a brief review of the general ideas behind ICA and CC-ICA, which could be useful to highlight the difference between them and PC-ICA.

5.3.1 General overview of ICA

ICA attempts to transform the variables into new ones that are mutually independent or as independent as possible with each other. It is therefore a more powerful technique for feature extraction than PCA. In application, ICA has been widely applied to solve various classification problems, e.g. microarray data analysis (Zheng et al., 2006; Liu et al., 2009b) and ECG beat classification (Yu and Chou, 2008, 2009).

For ease of presentation, we assume that the genes chosen from the MRMR feature selection method based on mutual information are $g_1, g_2, \cdots, g_N$. The microarray data matrix after feature selection is denoted by $\mathbf{Y} = (g_1, g_2, \cdots, g_N) = (x_{ki})_{K \times N}$. Given the N features, the idea of ICA is to use a certain projection matrix $\mathbf{W}$ to transform the original features into a new set of features. Mathematically, the basic ICA model can be formulated as

$$\mathbf{Z}^T = \mathbf{W} \cdot \mathbf{Y}^T \tag{5.8}$$

where $\mathbf{W}$ is a $N \times N$ projection matrix and $\mathbf{Z}$ is a K by N source matrix. If the new features are mutually independent and at most one new feature is normally distributed, $\mathbf{W}$ is completely determined (Bressan and Vitria, 2003).

The next task is to estimate $\mathbf{W}$ and $\mathbf{Z}$ which can be used to train a classifier and do classification. It can be done by optimizing objective functions such as maximizing likelihood and negentropy or minimizing mutual information. To efficiently solve the optimization problems and derive the independent components, Hyvarinen and Oja (1997) developed a robust algorithm termed as FastICA algorithm, which has been briefly described in Chapter 3.

Despite the popularity of ICA in feature extraction, the features obtained from ICA model may hardly satisfy the class-conditional independence assumption of various features that are taken by the naïve Bayes classifier. To make the features as independent as possible within each class, Bressan and Vitria (2003) proposed CC-ICA method for naïve Bayes classifier, which is described in the next section.

5.3.2 General overview of CC-ICA

CC-ICA is built upon the idea that ICA is performed within each class so that one projection matrix will be obtained for each class. The new features obtained through CC-ICA feature transformation may satisfy the class-conditional independence assumption of the naïve Bayes classifier better. In application, the effectiveness of CC-ICA in improving the naïve Bayes classifier has been demonstrated by Vitria et al. (2007) and Fan et al. (2009).

Assume that $\mathbf{Y}_k$, $\mathbf{Z}_k$ and $\mathbf{W}_k$ are respectively the original microarray data, the independent components and the projection matrix for class k. The basic CC-ICA model can be written as

$$\mathbf{Z}_k^T = \mathbf{W}_k \mathbf{Y}_k^T, \quad k = 1,2,\cdots,L \tag{5.9}$$

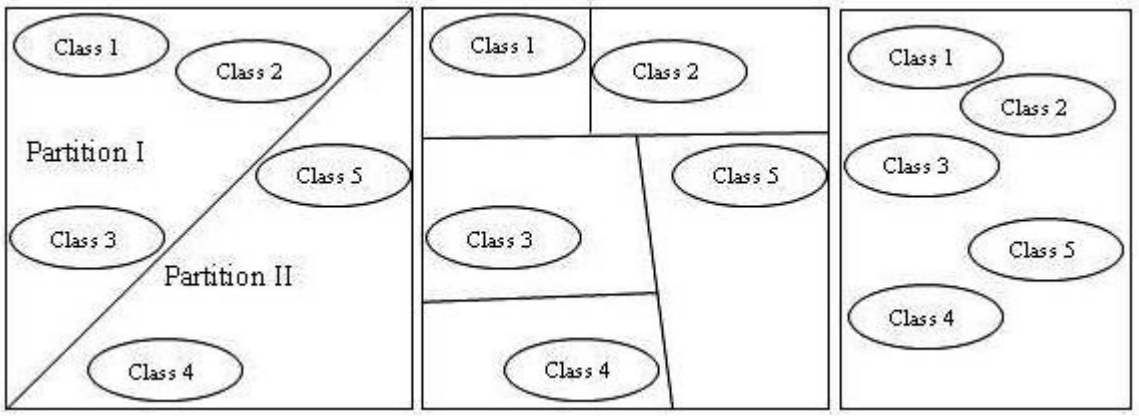
By applying the FastICA algorithm to solve the CC-ICA models, we can obtain the projection matrix $\mathbf{W}_k$ and the independent component matrix $\mathbf{Z}_k (n = 1,2,\cdots,N)$ for class k. Then we can use them to train a naïve Bayes classifier and do classification. More detailed discussions on CC-ICA can be found in Bressan and Vitria (2003) and Chapter 3 of this thesis.

5.3.3 Partition-conditional ICA

PC-ICA is an extension to CC-ICA for dealing with the case when CC-ICA cannot be employed due to the small sample sizes for some classes. The main idea of PC-ICA is to split the small-size samples into different partitions in an appropriate manner so that ICA can be done within each partition. Compared to ICA and CC-ICA, PC-ICA represents an in-between concept. If each class has enough samples to do ICA, there is no need to split the samples into partitions and PC-ICA will become CC-ICA. If all the classes are finally grouped into one partition, CC-ICA will collapse to ICA.

Figure 5.1 graphically illustrates the differences between PC-ICA, CC-ICA and ICA. In PC-ICA, the samples for classes 1, 2 and 3 are grouped into Partition I and the remaining into Partition II. ICA is carried out for each of the two partitions.

In CC-ICA, ICA is performed for each class based on their samples. In contrast, ICA estimates the independent components by using all the sample data for all the classes.



(a) PC-ICA **(b) CC-ICA** **(c) ICA**

Fig. 5.1. Graphical illustration of the difference among PC-ICA, CC-ICA and ICA

Technically, we assume that the K classes are grouped into R ($R \le K$) partitions. Let Y^r denote the microarray data for partition r and y^r denote the vector of gene variables. The PC-ICA model can be formulated as

$$z^r = \mathbf{W}^r y^r, \quad r = 1,2,\cdots,R \tag{5.10}$$

where z^r and $\mathbf{W}^r$ are respectively the independent components and the projection matrix for partition r. Similar to ICA and CC-ICA, the FastICA algorithm can be applied to solve Eq. (5.10).

Since PCA is often taken as the preprocessing stage for ICA , $\mathbf{W}^r = \mathbf{B}^r (\mathbf{D}^r)^{-1/2} \mathbf{E}^r$ where $\mathbf{E}^r$ is the eigenvector matrix from PCA, $\mathbf{D}^r$ is the diagonal matrix with the corresponding eigenvalues λ_r^n $(n = 1,2,\cdots,N)$, and $\mathbf{B}^r$ is the

ICA projection matrix for the whitened data for partition r. Assume that the partition-conditional representation provides independent components, the class-conditional probability in transformed space can be expressed as

$$p(x^r \mid c_k) = \alpha^r p(z^r \mid c_k) = \alpha^r \prod_{n=1}^{N} p(z_n^r \mid c_k) \tag{5.11}$$

where $\alpha^r = \left|\det((\mathbf{D}^r)^{-1/2})\right| = \prod_n 1/\sqrt{\lambda_n^r}$. Accordingly, the naïve Bayes classifier based on log-likelihoods and PC-ICA representation of original data is

$$c^* = \arg\max_k \sum_{n=1}^{N} \log p(z_n^r \mid c_k) + \log(\alpha^r) \tag{5.12}$$

5.4 Methods for grouping classes into partitions

Up to now the model for using PC-ICA in naïve Bayes classifier has been established. In practice, we still need to split different classes into partitions where each partition has enough samples to implement ICA. A general principle is to set the classes with enough sample sizes as base partitions and then allocate other classes to the base partitions. Since microarray data often have a small number of classes, we may do the allocation by "trial and error" method. If there is only one class with smaller sample size which is the case of our experimental study, we can allocate the class to each of the base partitions. For every scenario, we train a naïve Bayes classifier with PC-ICA and test its performance. Comparing all the possible scenarios, we may be able to determine the "best" partition way. If there are more classes with smaller sample sizes, we can do the partition processes one by one and select the "best" partition way. In the case that there are a number of classes with smaller

sample sizes which rarely occur, we may allocate them to the base partitions randomly and choose the partition way with "best" classification performance for use.

In addition to the "trial-and-error" method, the partition process can be done by a formal procedure such as hierarchical clustering if there are many classes. There are various hierarchical clustering methods that are based on different ways of defining distance (or similarity) between clusters. The study by Kerr et al. (2008) provides an excellent review of alternative techniques for clustering microarray data. Hastie et al. (2009) recently gives a detailed introduction to alternative hierarchical clustering methods. Here we only briefly introduce several commonly used hierarchical clustering methods.

Conceptually, hierarchical clustering is a cluster analysis technique for constructing hierarchical representation of clusters in which the clusters at each level of the hierarchy are merged from a lower level. In general, hierarchical clustering can be performed through agglomerative (bottom-up) approach and divisive (top-down) approach (Hastie et al., 2009). For the purpose of doing PC-ICA, we only need to merge different classes into partitions. As such, the agglomerative approach seems to be more appropriate. The base for doing hierarchical clustering is to choose a distance measure for quantifying the dissimilarity between two samples. There are various distance measures available for use, e.g. Euclidean distance, Manhattan distance and Chebyshev distance. It is possible that different distance measures may lead to different partition strategies. Our suggestion is therefore to try the several simple but commonly used distance measures and make a comparison between then with regards

to the final classification results, which may provide more insights for the partition approach.

Assume that the distance between two samples s_i and s_j is denoted as $d(s_i, s_j)$. Our proposed hierarchical clustering algorithm for grouping classes into partitions is described below.

Step 1. Identify the classes for which ICA cannot be implemented and let C_1 denote the set of classes, i.e. $C_1 = \{C_{11}, \cdots, C_{1L_1}\}$. Let C_{2i} $(i = 1, \cdots, L_2)$ denote the classes for which ICA can be performed, which are treated as the basic partitions for doing ICA.

Step 2. Compute the distance between an element of C_1, e.g. C_{11}, and each element of C_2.

Step 3. Merge the element into its nearest partition to produce a new partition.

Step 4. Repeat Steps 2 and 3 until all the elements of C_1 have been merged into partitions.

In step 2, a distance measure between two groups of samples is needed. A suggestion is to use the nearest neighbor single linkage algorithm given by

$$D(C_{1i}, C_{2j}) = \min\{d(x, y\}: x \in C_{1i}, y \in C_{2j}\} \tag{5.13}$$

In the algorithm, the distance between groups is simply defined as the distance between the closest pair of objectives. In application, the usefulness and effectiveness of the algorithm has been verified by Brida et al. (2009).

5.5 Experimental results

We evaluate the performance of PC-ICA for naïve Bayes classification of microarray data based on two gene expression datasets, namely Leukemia-MLL and Lung Cancer I. The sources of the two datasets are Armstrong et al. (2002) and Bhattacharjee et al. (2001), respectively. Table 1 shows the main features of the two datasets in which the brackets give the numbers of instances for each.

Table 5.1
Summary of two microarray datasets

Dataset	Leukemia-MLL	Lung Cancer I
Number of genes	12528	12600
Number of classes	3	5
Number of instances	72 (24/20/28)	203 (139/21/20/6/17)

The hold-out classification accuracy rates are used to compare the performance of ICA and PC-ICA for naïve Bayes classification of microarray data. The case of CC-ICA is not included since it cannot be implemented when a class in the training data has not enough samples. Every dataset is split into two parts, i.e. training and test datasets. In our experiments, four fifth of the samples are randomly selected as the training data and the remaining one fifth of the samples are taken as the test data. The training dataset is used to do MRMR feature selection based on mutual information, carry out ICA/PC-ICA computation and train classifiers. In feature selection, we consider different cases where the number of genes chosen ranges from five to fifteen. The test dataset is used to compare the performances of

ICA and PC-ICA. Such a procedure is repeated ten times for ICA and PC-ICA on the two datasets.

Figure 5.2 shows the comparative boxplots of the Bayesian classification accuracy rates of Leukemia-MLL dataset for ICA and PC-ICA when the numbers of genes selected are 5, 7, 9, 11, 13 and 15. It can be seen from Fig. 5.2 that in most cases PC-ICA will lead to better classification performance than ICA in terms of mean, median and variances of classification rates. In the case of N=15, although the average classification accuracy rate for ICA is slightly higher than that for PC-ICA, the standard deviation of classification rates for PC-ICA is smaller than that for ICA.

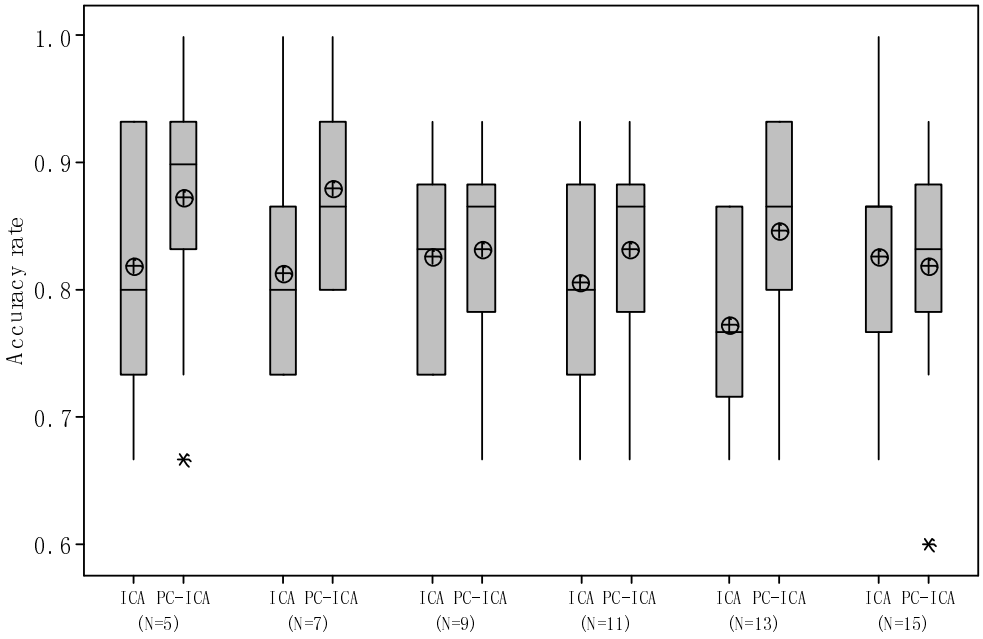


Fig. 5.2. Boxplots of classification accuracy rates for ICA and PC-ICA based on Leukemia-MLL dataset when the number of genes selected (N) is changeable

Figure 5.3 shows the comparative boxplots of the Bayesian classification accuracy rates for ICA and PC-ICA based on the Lung Cancer I dataset. We can find

that PC-ICA is generally superior to ICA in the sense that the former has a better average classification performance, which is consistent with the conclusion drawn from the experiments based on Leukemia-MLL dataset. A possible reason is that that the class-conditional independence of the features extracted by PC-ICA might be stronger than that by ICA. Therefore, the new features extracted from PC-ICA are more suitable for the naïve Bayes classifier compared to those from ICA.

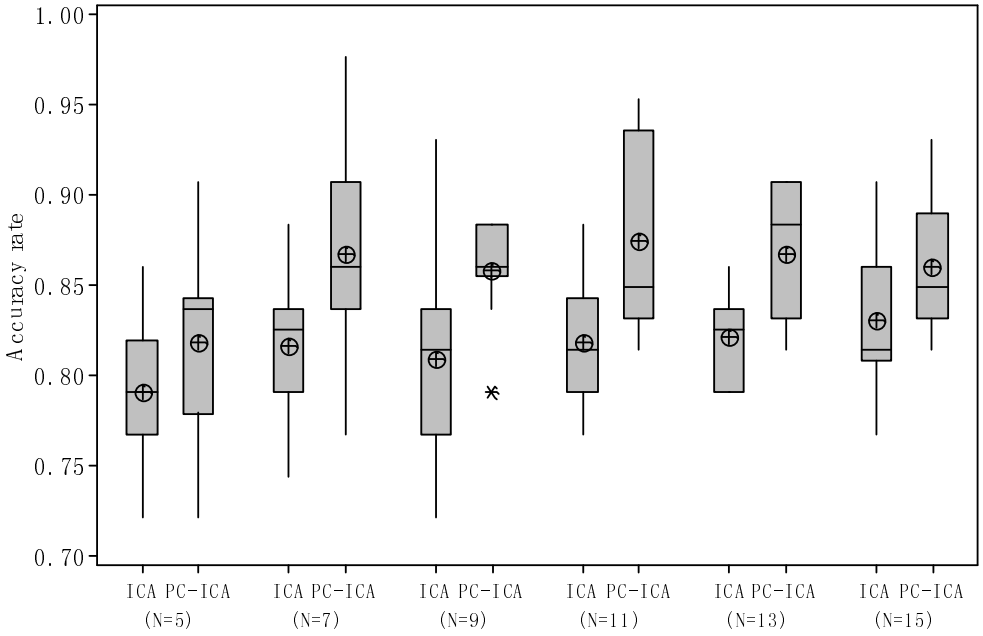


Fig. 5.3. Boxplots of classification accuracy rates for ICA and PC-ICA based on Lung Cancer I dataset when the number of genes selected (*N*) is changeable

5.6 Conclusion

Accurate classification of microarray data is very important for medical decision making. Past studies have shown that CC-ICA is effective in improving the performance of naïve Bayes classifier. In microarray data analysis, it is possible that the sample size for some classes is not large enough to perform ICA within each class.

In such a circumstance, the application of CC-ICA becomes infeasible. In this chapter, we extend CC-ICA and proposed PC-ICA for naïve Bayes classification of microarray data. A key feature of PC-ICA is that it uses ICA to do feature extraction within each partition consisting of several small-size classes. ICA and CC-ICA can be considered as two extreme cases of PC-ICA in feature extraction. Experimental results on two microarray datasets have shown that PC-ICA usually has better performance than ICA in naïve Bayes classification of microarray data. Further research may be carried out to extend this study by using more datasets and comparing it with other feature extraction techniques.

CHAPTER 6 ICA FOR MULTI-LABEL NAÏVE BAYES CLASSIFICATION

6.1 Introduction

Previous chapters deal with only single-label classification problems, in which each sample is associated with a single label from a set of disjoint classes. If the set of disjoint classes includes two elements, the single-label classification problem is referred to as a binary classification problem. If the set includes more than two elements, the single-label classification problem is called a multi-class classification problem. However, in some classification problems, a sample may simultaneously be associated with multiple labels. For instance, in text categorization a newspaper may belong to several pre-defined categories such as *Society* and *Movies*. This type of problems is often called multi-label classification problems. Although multi-label classification was originally motivated by text categorization and medical diagnosis, it has received increasing attention in other domains of pattern classification, such as music categorization, scene classification, and protein function classification (Tsoumakas and Katakis, 2007).

Various approaches have been proposed in the literature for dealing with multi-label classification problems. For instance, Chen et al. (2003) designed a decision tree classifier for solving multi-value and multi-label classification problems. Later, Chou and Hsu (2005) extended it and proposed the so-called multi-valued and multi-labeled decision tree to improve the classification accuracy of the original

decision tree classifier. Boutell et al. (2004) compared several possible approaches to training and testing classifier and developed new metrics for evaluating the performance of multi-label classifiers. Zhang and Zhou (2007) extended the traditional K-nearest neighbor (KNN) and proposed a multi-label KNN (ML-KNN) algorithm for solving multi-label classification problems, which has been found to be competitive with some more sophisticated algorithms. More recently, Zhang and Wang (2009) developed an algorithm based on two-layer radial basis function neural networks for solving multi-label classification problems. Their experiments on two real-world multi-label classification tasks demonstrate the effectiveness of the algorithm. Cheng and Hullermeier (2009) unified instance-based learning and logistic regression and proposed a more general approach for multi-label classification, which overcomes some limitations of existing instance based multi-label classification methods.

In the case of naïve Bayes classifier, Zhang et al. (2009) recently extended it and proposed a classifier called multi-label naïve Bayes (MLNB) to handle multi-label classification problems. A two-stage filter-wrapper feature selection strategy, consisting of the usage of PCA and genetic algorithm, is incorporated into the MLNB in order to improve its classification performance. As discussed in Zhang et al. (2009), this study could be the first one in which feature selection is incorporated into the multi-label learning algorithm. Their experimental results have shown that feature selection is capable of improving the performance of MLNB significantly.

The work by Zhang et al. (2009) has laid a good foundation for further research on the use of naïve Bayes in solving multi-label classification problems.

Previous chapters have demonstrated the usefulness of ICA in improving the classification performance of single-label naïve Bayes. The purpose of this chapter is to explore the usefulness of ICA as a feature extraction method in MLNB, which could not only expand the application scope of ICA but also improve the classification performance of MLNB. We first describe multi-label classification problems in a general manner, which are followed by an overview of the methods used for multi-label classification. We then propose ICA-based MLNB (or ICA-MLNB) scheme for solving multi-label classification problems. Finally, experimental studies on two multi-label datasets are presented, which shows that ICA is an effective feature extraction method for improving the performance of the MLNB classifier.

6.2 Multi-label classification problem

We use a simple example to illustrate the concept of multi-label classification problem. Assume that there are five documents and the first document can be simultaneously classified into the classes of computer science, mathematics and application. The second document belongs to the classes of biology and the third document can be assigned to classes of mathematics, physics and theory. The fourth document belongs to the class of application, and the fifth can be classified as physics or application. This problem is a typical multi-label text classification problem, which is shown in Table 6.1. It consists of five samples and six classes (or labels) including computer science, mathematics, physics, biology, theory and application. Each sample document belongs to one label or more than one label simultaneously. The task is to

learn a classifier from the five documents with good generalization capability for predicting the labels of a new document.

Table 6.1

A simple multi-label classification problem

Sample	Computer science (C)	Mathematics (M)	Physics (P)	Biology (B)	Theory (T)	Application (A)
1						
2						
3						
4						
5						

Mathematically, suppose that $\mathbf{X}$ denotes the input space and let $\Omega = \{\lambda_1, \lambda_2, \cdots, \lambda_s\}$ denote a finite set of class labels. Further assume that each sample $x \in \mathbf{X}$ is associated with a subset of labels $L_x \in 2^{\Omega}$. In multi-label classification, L_x is often referred to as the set of relevant labels while its complement $\Omega \setminus L_x$ is called the set of irrelevant labels. Given a set of training data $D = \{(x_i, L_i) | i = 1,2,\cdots,M\}$ where x_i is the input vector of sample i and $L_i \subseteq \Omega$ is the set of labels associated with x_i, the task of the multi-label classification problem is to train a function $f : \mathbf{X} \to 2^{\Omega}$ so that f predicts the label sets well for each unseen sample. Alternatively, the learning system can be represented by a real-valued function g such that $g : \mathbf{X} \times \Omega \to \mathbf{R}$. Given a sample x_i and its associated label set L_i, a good classifier will produce a larger function values for labels in the label set than those not in the label set. That is to say, if $l \in L_i$ and $l' \notin L_i$, we have $g(x_i, l) > g(x_i, l')$.

Once a multi-label classifier is learned, we need to evaluate its performance before putting it into application. Usually, the performance evaluation of a multi-label classifier is more complicated than traditional single-label classifier. In the literature, a number of criteria and metrics have been developed for evaluating the performance of multi-label classifier. Here we shall introduce several commonly used measures (Zhang et al., 2009). Given a set of testing samples x_i ($i = 1, \cdots, I$), let $f(x_i) \subseteq \Omega$ denote the multi-label prediction and L_i denote the real set of labels for x_i. The most commonly used evaluation metric is termed as *Hamming loss*, which is defined as follows.

$$\text{HamLoss}(f) = \frac{1}{I}\sum_{i=1}^{I}\frac{1}{|\Omega|}\left|f(x_i)\Delta L_i\right| \tag{6.1}$$

where Δ is the symmetric difference between two sets (corresponding to the XOR operation in Boolean logic), and $|\cdot|$ represents the cardinality of a set (i.e. the number of its elements). *Hamming loss* is actually a measure of the percentage of labels whose relevance is incorrectly predicted.

Suppose that the real-valued scoring function $g(x_i, l)$ has been defined. It can be transformed into a ranking function $rank_g(x_i, l)$, which maps the outputs of $g(x_i, l)$ for any l such that if $g(x_i, l_1) > g(x_i, l_2)$ then $rank_g(x_i, l_1) > rank_g(x_i, l_2)$. We can also define several other metrics used for evaluating the performance of multi-label classifiers. The measure of "*one error*", which attempts to compute how many times the top-ranked label is not relevant, is expressed as follows

$$\text{OneError}(g) = \frac{1}{I}\sum_{i=1}^{I}\psi(x_i) \tag{6.2}$$

where $\psi(x_i)$ is defined as

$$\psi(x_i) = \begin{cases} 1 & \text{if } \arg\max_{\lambda\in\Omega} g(x_i,\lambda) \notin L_i \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

In addition to *one error*, there are some other metrics that have been reported and used in the literature. Coverage, i.e. Eq. (6.4), is defined as the average distance to cover all the relevant labels assigned for the testing samples.

$$\text{Coverage}(g) = \frac{1}{I}\sum_{i=1}^{I}\max_{\lambda\in\Omega} rank_g(x_i,\lambda) - 1 \tag{6.4}$$

Ranking loss refers to the average fraction of label pairs that are not correctly ordered for the sample. Mathematically, it can be expressed as

$$\text{RankLoss}(g) = \frac{1}{I}\sum_{i=1}^{I}\frac{1}{\left|L_i\right|\left|\overline{L_i}\right|}\left|\{(l_i,\bar{l}_i)\middle| g(x_i,l_i) < g(x_i,\bar{l}_i),\right.$$

$$\left. l_i \in L_i \text{ and } \bar{l}_i \in \overline{L_i}\}\right| \tag{6.5}$$

where $\overline{L_i}$ is the complementary set of L_i.

Average precision measures the average fraction of relevant labels ranked above a particular relevant label. It is given by

$$\text{AvePrec}(g) = \frac{1}{I}\sum_{i=1}^{I}\frac{1}{\left|L_i\right|}\times$$

$$\sum_{l \in L_i} \frac{\left| \{ l' \left| rank_f(x_i, l') \le rank_f(x_i, l), l' \in L_i \} \right|}{rank_f(x_i, l)} \tag{6.6}$$

Among the five metrics mentioned above, the first four are of "cost" type. That is to say, a smaller value means a better classification performance. However, *average precision* is a benefit type of measure. A larger *average precision* value implies a better classification performance. If AverPrec(g) is equal to 1, it means that the classifier has perfect classification performance.

6.3 Multi-label classification methods

Researchers have proposed various methods for solving multi-label classification problems. The study by Tsoumakas and Katakis (2007) provides a general overview of the multi-label classification methods. Recently, de Carvalho and Freitas (2009) provide a more comprehensive introduction to various methods for multi-label classification. Broadly speaking, the existing methods used for multi-label classification can be classified into two main categories, namely algorithm independent approach and algorithm adaptation approach. Algorithm adaption approach tries to adapt some well-established single-label classification algorithms to solve multi-label classification problems. In contrast, the algorithm independent approach is to transform the original multi-label classification problem into a set of single-label problems. Then, any learning algorithm used in solving single-label classification problems can be directly applied to multi-label classification.

As the MLNB classifier proposed in Zhang et al. (2009) is an algorithm independent approach, we shall only briefly describe the algorithm independent

approach in this chapter. Details on the algorithm adaption approach can be found in Tsoumakas and Katakis (2007) and de Carvalho and Freitas (2009). According to de Carvalho and Freitas (2009), there are two kinds of problem transformation methods in algorithm independent approach. One is based on labels and the other is based on samples, which are respectively called label-based transformation and sample-based transformation.

6.3.1 Label-based transformation

Label-based transformation for multi-label problems has some similarities with the one-against-all approach for multi-class problems (Hsu and Lin, 2002). The purpose of one-against-all approach is to use binary classifiers to solve a classification problem with more than two classes. For label-based transformation, the original multi-label problem can be transformed to a set of single-label problems and each label is associated with a binary classification problem. Then a binary classifier can be trained for each of the binary classification problems.

The process of label-based transformation can be illustrated by the simple multi-label classification problem given in Table 6.1. Since there are six classes or labels, the original problem can be divided into six binary classification problems that are associated with the six classes or labels. Table 6.2 shows the resulting six binary classification problems transformed from the original multi-label classification problem.

Table 6.2

Six binary classification problems obtained from label-based transformation

Sample	Problem 1	Problem 2	Problem 3	Problem 4	Problem 5	Problem 6
1	C+	M+	P-	B-	T-	A+
2	C-	M-	P-	B+	T-	A-
3	C-	M+	P+	B-	T+	A-
4	C-	M-	P-	B-	T-	A+
5	C-	M-	P+	B-	T-	A+

Note: "+" and "-" respectively denote that the sample is positive or negative for the current class.

Technically, the label-based transformation attempts to train a separate binary classifier h_i for each label $\lambda_i \in \Omega$, i.e.

$$h_i(x) = \begin{cases} 1 & \text{if label } \lambda_i \text{ is relevant to } x \\ 0 & \text{otherwise} \end{cases} \tag{6.7}$$

From Eq. (6.7), we can derive the following multi-label classifier for predicting the labels of x

$$L_x = \bigcup_{\lambda_i \in \Omega} \{\lambda_i | h_i(x) = 1\} \tag{6.8}$$

The main advantage of the label-based transformation approach lies with its simplicity for use. In addition, all the methods for single-label classification can be directly taken for use. However, since it treats every label independent with each other, it has the disadvantage that the correlations and interdependencies between various labels are not considered in classification. Nevertheless, previous studies, e.g. Zhang et al. (2009), have shown the effectiveness of label-based transformation approach in solving multi-label classification problems.

6.3.2 Sample-based transformation

The use of sample-based transformation approach helps to convert the original multi-label problem into one or more single-label problems through redefining the set of labels associated with each sample. Compared to label-based transformation that only generates binary classification problems, sample-based transformation may produce binary or multi-class classification problems.

Table 6.3
Single-label problem through eliminating samples with more than one label

Sample	Class
2	B
4	A

There exist several sample-based transform methods that can convert a multi-label problem into traditional single-label problem. The most straightforward, also the least effective, sample-based transformation method is to eliminate the samples with more than one label. Table 6.3 shows the resulting problem from the use of this method for the previous simple example. It can be seen that three samples are eliminated, which essentially changes the current problem to another simpler problem. An obvious drawback of this method is that it leads to the loss of information. This method is suitable for the datasets that have few multi-label samples.

In addition, one may choose to keep only one label for multi-label samples instead of eliminating the samples. This method is to convert the multi-label samples into single-label samples by simplification. It can be done through randomly selecting one label or using a certain criterion to select for the multi-label samples (see Table

6.4). Since selecting one label may oversimplify the problem, we can also decompose the multi-label problem into a set of single label problems in appropriate manner. The decomposition can keep the original information while the number of classifiers may become very large, which is equal to the product of the number of labels for each sample. For previous example, 18 single-label classifiers need to be trained if a decomposition is carried out.

Table 6.4

Single-label problem through selecting one label for multi-label samples

Sample	Class
1	M
2	B
3	P
4	A
5	A

The original multi-label problems can also be converted into a single-label problem by considering all the possible label sets as new classes. The creation of new classes can largely increase the number of classes, which may cause some classes to be with very few samples. Table 6.5 shows the single-label problem derived from the method. It can be easily seen that the number of classes has increased. Compared to previous sample-based transformation methods, this method will not result in the loss of information but requires the learning algorithms to be capable of dealing with small-sample datasets.

Table 6.5

Single-label problem through creating new classes for multi-label samples

Sample	Class
1	MixClass1
2	B
3	MixClass2
4	A
5	MixClass3

6.4 ICA-based multi-label naïve Bayes

6.4.1 Basic multi-label naïve Bayes

The recent study by Zhang et al. (2009) provides a theoretical description of multi-label naïve Bayes (MLNB) classifier. For a testing sample $x = (x^1, x^2, \cdots, x^n) \in \mathbf{X}$ associated with label set $L_x \subset \Omega$, let $\vec{L}_x$ denote its label vector in which the *i*th component $\vec{L}_x(i) = 1$ if $\lambda_i \in L_x$ and $\vec{L}_x(i) = 0$ if $\lambda_i \notin L_x$. Assume that H_1^i is the event that x has label λ_i and H_0^i is the event that x has no label λ_i. The category vector can be predicted using the following principle:

$$\vec{L}_x(i) = \arg\max_{b\in\{0,1\}} P(H_b^i \big| x), \quad i = 1,2,\cdots,s \tag{6.9}$$

Using the Bayes rule, we can transform Eq. (6.9) into

$$\vec{L}_x(i) = \arg\max_{b\in\{0,1\}} \frac{P(H_b^i)P(x\big|H_b^i)}{P(x)}, \quad i = 1,2,\cdots,s \tag{6.10}$$

By assuming the class conditional independence assumption among features, we can rewrite Eq. (6.10) as

$$\vec{L}_x(i) = \arg\max_{b\in\{0,1\}} P(H_b^i)\prod_{k=1}^{n} P(x^k \left| H_b^i \right.), \quad i = 1,2,\cdots,s \tag{6.11}$$

In practice, we often use the additive form of Eq. (6.11) that is given below:

$$\vec{L}_x(i) = \arg\max_{b\in\{0,1\}} \ln P(H_b^i) + \sum_{k=1}^{n} \ln P(x^k \left| H_b^i \right.), \quad i = 1,2,\cdots,s \tag{6.12}$$

In addition, the conditional probabilities in Eq. (6.11) are often replaced by their kernel density estimations. If the Gaussian probability density function is assumed, we can obtain the following equation:

$$\vec{L}_x(i) = \arg\max_{b\in\{0,1\}} \left\{ \ln P(H_b^i) - \sum_{k=1}^{n} \frac{(x^k - \mu_k^{ib})^2}{2\sigma_k^{ib}} - \sum_{k=1}^{n} \ln \sigma_k^{ib} \right\}, \; i = 1,2,\cdots,s \tag{6.13}$$

where μ_k^{ib} and σ_k^{ib} are respectively the mean and standard deviation of feature k with respect to label i.

It should be pointed out that the MLNB described above is essentially a set of naïve Bayes classifiers for single-label classification problems. In the case that there exist a large number of features, the computation of Eq. (6.13) may exceed the floating precision of a computer. So Zhang et al. (2009) derived a variant of Eq. (6.13) for computation purpose. In our study, since feature selection is carried out before training the naïve Bayes classifiers, Eq. (6.13) can be directly taken for use. However, as discussed in previous chapters, the class-conditional independence assumption may

not hold in real world applications. Therefore, we propose the use of ICA as a feature extraction method for MLNB.

6.4.2 ICA-MLNB classification scheme

The effectiveness of ICA as a feature extraction method for naïve Bayes classifier has been empirically demonstrated by some earlier studies. However, none of any previous studies dealt with the use of ICA in MLNB for solving multi-label classification problems. In the previous chapters of this thesis, we have shown that CC-ICA performs better than ICA for naïve Bayes classifier. In the case of MLNB, since the label-based transformation method described in Section 6.3.1 is used, the multi-label classification problem is finally transformed into a set of binary classification problems. Since there are only two classes for each problem, in this thesis we shall only investigate the use of ICA in MLNB, which is referred to as the ICA-MLNB classification scheme.

As pointed out in previous chapters, many classification problems may deal with only a few of samples with respect to the number of features. A well known example is microarray data analysis in which a dataset consists of a small number of samples but a huge number of genes. This "curse of dimensionality" problem also occurs in multi-label classification problems. As such, feature selection is often a necessary step before using ICA to do feature transformation. Here we choose the mutual information –based feature selection method as used in Chapter 5, which is based on the MRMR criterion as suggested by Peng et al. (2005).

In our proposed ICA-MLNB scheme, the label-based transformation is first applied to transform the original multi-label classification problem into a set of binary classification problems, each of which is corresponding to one label. For each binary classification problem, we use the dataset to do feature extraction using the mutual-information -based MRMR criterion and ICA. The independent components obtained are then used to train the binary naïve Bayes classifier for the dataset. The complete description of the ICA-MLNB scheme is given below.

Step 1. Split the dataset into training and test datasets.

Step 2. For both training and test datasets, transform the multi-label classification problems into a set of (s) binary classification problems.

Step 3. For a binary classification problem,

3.1 Do feature selection for the training data using the mutual information – based MRMR criterion;

3.2 Do feature transformation using ICA or CC-ICA for the training data and get the transformation matrix.

Step 4. Use the training data after feature extraction to learn a naïve Bayes classifier.

Step 5. For the test data,

5.1 Choose the same features as selected in Step 3.1;

5.2 Do feature transformation using the same transformation matrix as estimated in Step 3.2;

5.3 Use the new test data obtained and the classifier learned in Step 4 to perform the classification task.

Step 6. Repeat Steps 3 to 5 until all the binary classification problems are solved. Combine their classification results to assess the performance of the MLNB.

6.5 Empirical study

We empirically examine whether ICA could help to improve the classification performance of MLNB. Two real-world datasets, which were obtained from the website of the machine learning and knowledge discovery group in the Aristotle University of Thessaloniki, are used in our empirical study. The first is to predict the gene functional classes of the Yeast dataset, which has been investigated by Elisseeff and Weston (2002) and Zhang et al. (2009). The Yeast dataset consists of 2417 genes, each of which is characterized by 103 features. Each gene is associated with a set of functional classes, which is structured into a tree with leaves representing the functional categories. At the finest level of the tree, the number of the functional classes may reach as high as 190. The top level of the tree consists of 14 functional class categories, which are the labels we attempt to use ICA-MLNB to predict in this study. For each gene, the average number of the labels is 4.24.

The second empirical study is about natural scene classification, which deals with the prediction of label set for a number of natural scene images. The dataset was initially proposed and studied by Boutell et al. (2004) and later used by many multi-label classification studies. The natural scene dataset consists of 2407 samples and 294 features. Each sample is associated with at most six labels simultaneously, which include beach, fall foliage, sunset, field, mountain and urban. The average number of labels for each sample image is 1.07.

Ten-fold cross-validation is used to assess the performance of our proposed ICA-MLNB scheme. For each of the two datasets, we divide it into 10 parts with approximately equal sizes. Every time we will choose one of the 10 parts as test data for examining the performance of MLNB and ICA-MLNB. The remaining nine parts are taken as training data for learning MLNB and ICA-MLNB. The process is repeated for 10 times so that each of the 10 parts is used for test data once. We then compute the *Hamming loss* values for all the possible scenarios and use them to compare the performance of ICA-MLNB and MLNB. It should be pointed out that other metrics in addition to *Hamming loss*, e.g. those described earlier in this chapter, have also been used for assessing the performance of multi-label classifiers. However, our empirical study does not use other metrics due to the following two reasons. Firstly, past empirical studies have shown thatusing different metrics may lead to different conclusions in terms of classification performance, which makes the interpretation of the results difficult. As *Hamming loss* is one of the most popular metrics, we choose it rather than other metrics for use. Secondly, the use of *Hamming loss* does not require us to define the real-valued function as mentioned in Section 6.2, which avoids the introduction of subjective factors to a certain degree.

Since the number of features selected by the mutual information -based MRMR criterion may affect classification results, we also empirically examine the effect of feature size on classification performance. Figure 6.1 shows the average *Hamming loss* values for MLNB and ICA-MLNB classification of Yeast data when the number of features selected varies from 11 to 20. It can be seen that the ICA-MLNB classification scheme performs better than MLNB in terms of the *Hamming*

loss metric. The average *Hamming loss* for ICA-MLNB is always below 0.02, which is much lower than that for MLNB.

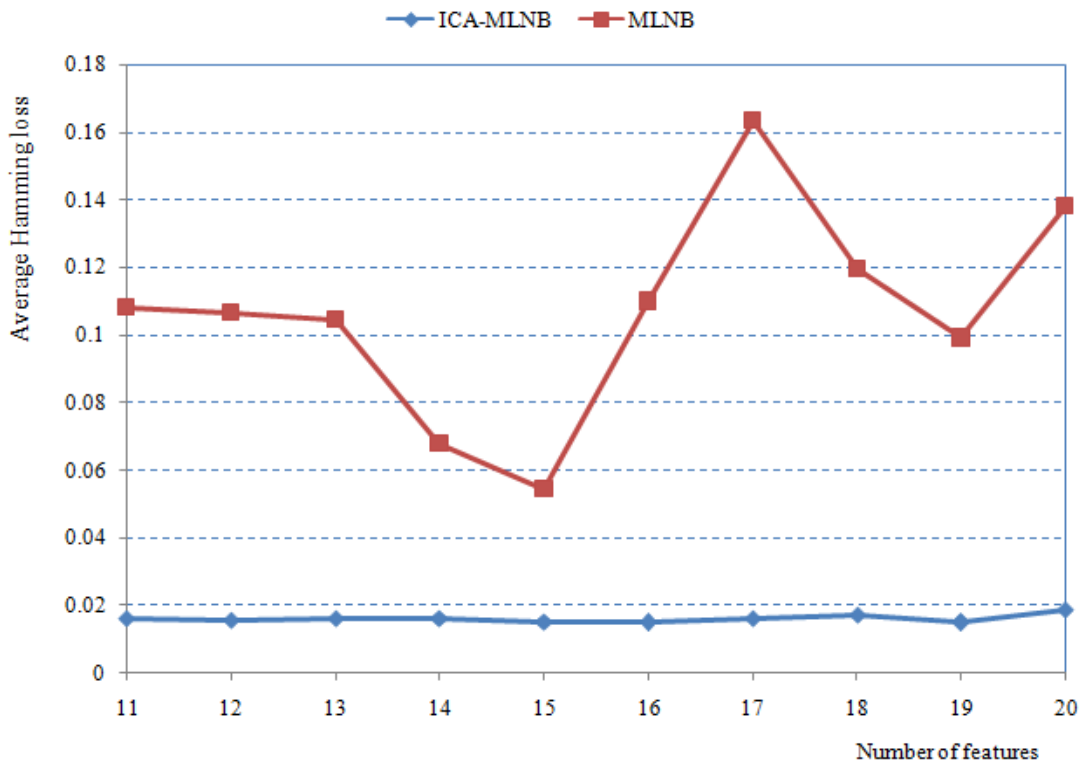


Fig. 6.1. The average Hamming loss for MLNB and ICA-MLNB classification of Yeast data when the number of features varies from 11 to 20

In order to examine the variation of the *Hamming loss* for different classifiers, we also present the comparative boxplots of the Hamming loss values for MLNB and ICA-MLNB by fixing the number of features selected at 11, 14, 17 and 20 in Fig. 6.2. It is found that in general there exist little differences in the classification performance of MLNB and ICA-MLNB when the number of features is relatively smaller. However, when the number of features becomes larger, the variation of *Hamming loss* for MLNB becomes much higher. In addition, it can also been observed from Fig. 6.2 that there are several very large *Hamming loss* values for the MLNB classifier.

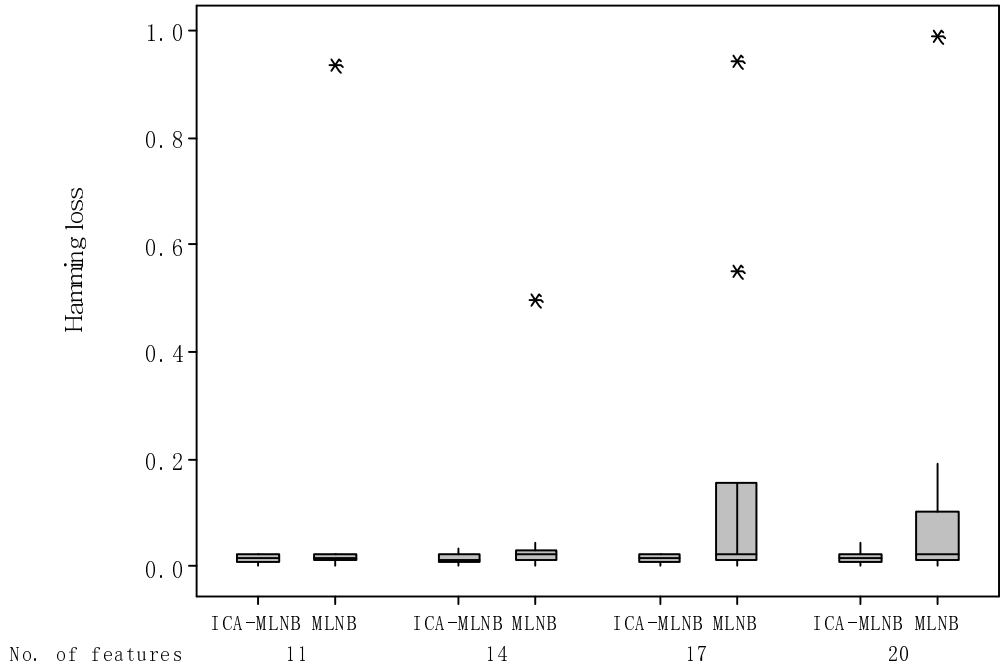


Fig. 6.2. Comparative boxplots of Hamming loss for MLNB and ICA-MLNB classification of Yeast data with various feature sizes

Figure 6.3 shows the average *Hamming loss* values for MLNB and ICA-MLNB classification of natural scene data when the number of features selected varies from 11 to 20, and Fig. 6.4 provides the comparative boxplots for several pre-defined scenarios. It can be seen from Fig. 6.3 that ICA-MLNB always has smaller average *Hamming loss* values than MLNB. In addition, the average *Hamming loss* for ICA-MLNB looks more stable than MLNB, which might be an indication of its better classification performance. As shown in Fig. 6.4, the variations of *Hamming loss* for ICA-MLNB are often smaller than those for MLNB.

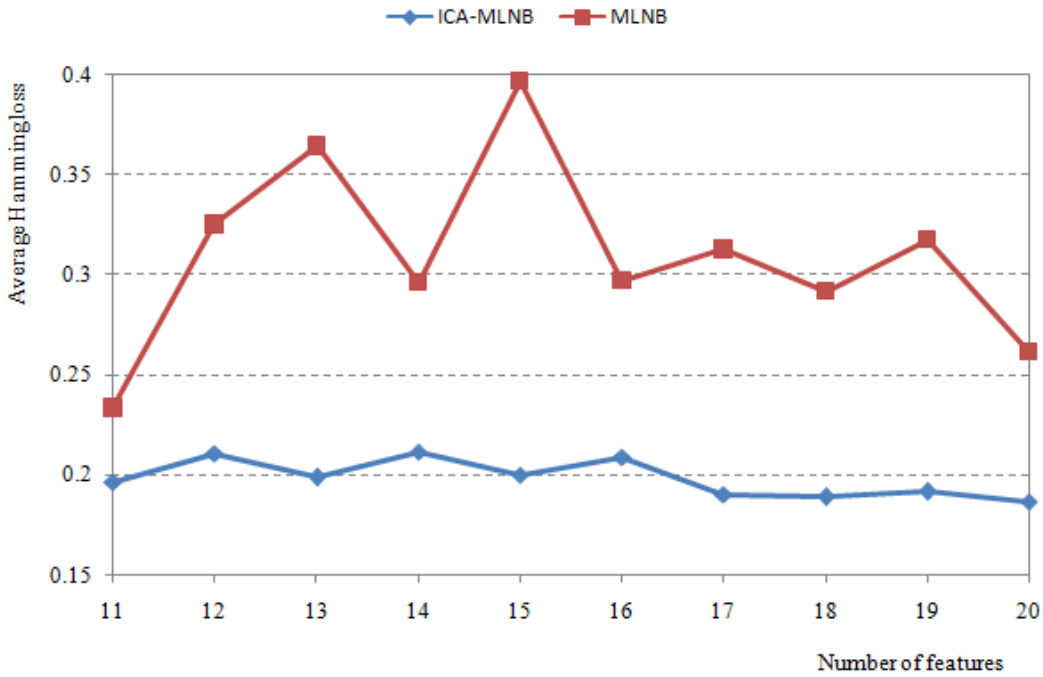


Fig. 6.3. The average Hamming loss for MLNB and ICA-MLNB classification of natural scene data when the number of features varies from 11 to 20

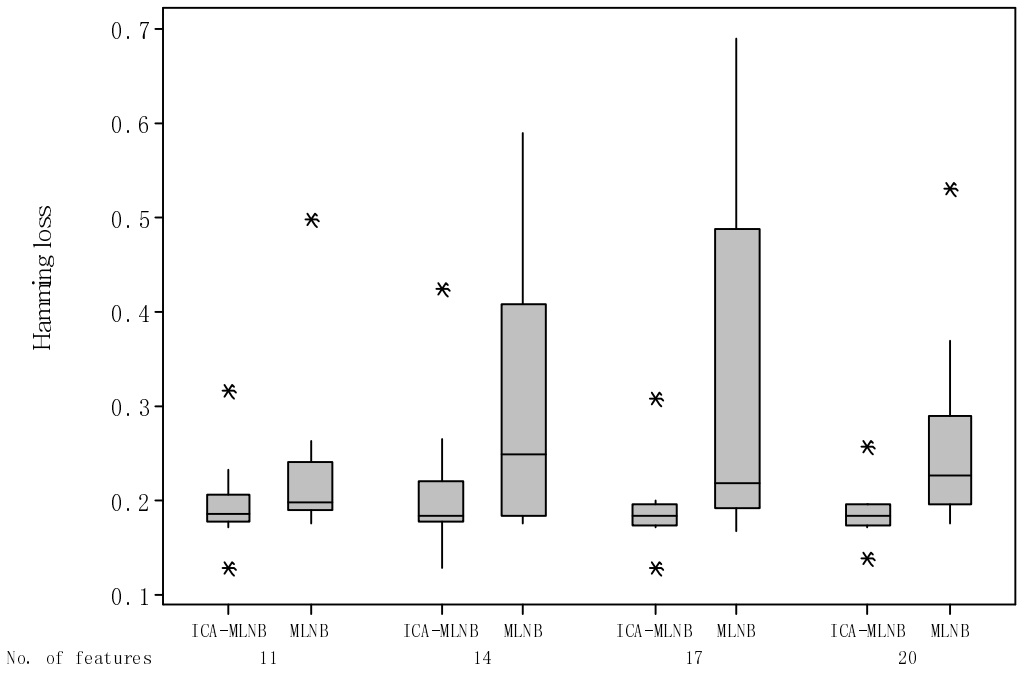


Fig. 6.4. Comparative boxplots of Hamming loss for MLNB and ICA-MLNB classification of natural scene data with various feature sizes

In summary, our experiments on Yeast and natural scene datasets show that ICA-MLNB often performs better than MLNB in terms of the *Hamming loss* metric. In most cases, ICA-MLNB has not only a smaller average *Hamming lo*ss value, but also a smaller variation in *Hamming loss*. It implies that ICA as a feature extraction technique can effectively improve the performance of MLNB.

6.6 Conclusion

Multi-label classification has received increasing attention in different application domains of pattern classification. Various approaches have been proposed in the literature for solving multi-label classification problems, among which MLNB can be treated as an important extension to traditional naïve Bayes for single-label classification.

Despite the usefulness of MLNB, none of previous studies attempt to incorporate ICA as a feature extraction tool into it. As such, in this chapter we propose the ICA-MLNB scheme for multi-label classification. Our experimental results on two real-world datasets have shown that in general ICA-MLNB has not only smaller average *Hamming loss* values but smaller variations in the metric than MLNB. It may be an indication that ICA can improve the classification performance of MLNB in solving multi-label classification problems. As the main purpose of this chapter is to examine the effectiveness of ICA in improving MLNB, we do not compare the performance of ICA-MLNB with other multi-label classifiers. Further research may be carried out to extend this study by using more datasets and comparing ICA-MLNB with other multi-label classifiers.

CHAPTER 7 CONCLUSIONS AND FUTURE RESEARCH

This thesis contributes to several methodological and application issues in applying ICA to the naïve Bayes classifier. In this chapter we will summarize and discuss the main results of our research work as described in previous chapters. Possible future research will also be presented.

7.1 Summary of results

In Chapter 3, we present a comparative study of PCA, ICA and CC-ICA as alternative feature extraction methods for naïve Bayes classifier. Our experimental results have shown that all of the three feature extraction methods can improve the performance of naïve Bayes classifier. In most cases, CC-ICA integrated with naïve Bayes outperforms PCA and ICA integrated with naïve Bayes in terms of classification accuracy, which offers clear evidence on the suitability of CC-ICA as a feature extraction method for naïve Bayes classifier.

The use of CC-ICA often requires a large number of samples. When the sample size is much less than the number of the features, e.g. in the case of microarray data analysis, its direct use may become infeasible. We therefore present a sequential feature extraction approach for naïve Bayes classification of microarray data in Chapter 4, which starts from gene selection by stepwise regression. The data on the genes selected are then transformed by CC-ICA, which makes the new features after transformation become as class-conditionally independent with each other as

possible. Our experimental results on five microarray datasets demonstrate the effectiveness of the sequential feature extraction approach in improving the classification performance of naïve Bayes classifier in microarray data analysis.

The research work presented in Chapter 4 makes the use of CC-ICA as a feature extraction method becomes more applicable for naïve Bayes classification of microarray data. However, in some cases the sample sizes for some classes may be too small so that the implementation of CC-ICA is still infeasible after feature selection. To address this problem, we extend CC-ICA and propose PC-ICA for naïve Bayes classification of microarray data in Chapter 5. Compared to CC-ICA, PC-ICA attempts to implement ICA within each partition consisting of several small-size classes rather than each class. As such, PC-ICA encompasses ICA and CC-ICA as two special cases. Experimental results on several microarray datasets have shown that PC-ICA often has better performance than ICA in naïve Bayes classification of microarray data.

Our research in Chapters 4 and 5 is based on the assumption that naïve Bayes is used to solve single-label classification problems. However, in the real world a number of classification problems are essentially multi-label problems. Although the usefulness of multi-label naïve Bayes (MLNB) in dealing with multi-label classification problems has been demonstrated by earlier studies, none of previous studies incorporate ICA into MLNB. Therefore, in Chapter 6 we investigate the usefulness of ICA as a feature extraction method for MLNB classification of multi-label classification problems. Specifically, we propose the ICA-MLNB scheme for multi-label classification. Our experimental results on two real-world datasets have

shown that in general ICA-MLNB usually has better classification performance than MLNB, which may be an indication of the usefulness of ICA as a feature extraction method for MLNB classification of multi-label problems.

7.2 Possible future research

Despite the contributions described above, the work reported in this thesis has inevitably some limitations where further research may be carried out. Areas where further research would be fruitful are summarized as follows.

In our sequential feature extraction approach for naïve Bayes classification, feature selection is done through stepwise regression because of its simplicity and effectiveness. In the literature there are also a number of other feature selection techniques. It would therefore be meaningful to investigate whether various feature selection techniques would substantially affect the performance of naïve Bayes classifier in microarray data analysis.

As pointed out in Chapter 5, when CC-ICA cannot be applied due to the very small sample sizes for some classes, PC-ICA can be used as an alternative feature extraction technique for naïve Bayes classification of microarray data. However, a necessary step for using PC-ICA is to group different classes into some partitions. Although we have given some descriptions on how to group classes into partitions, further investigations on the methods for doing the grouping task would still be worthwhile while endeavor.

In Chapter 6 we propose the ICA-MLNB scheme for solving multi-label classification problems. As the main purpose of this chapter is to examine the

effectiveness of ICA in improving MLNB, we only compare the performance of ICA-MLNB with that of MLNB in our experiments. Further research may be carried out to extend this study by using more datasets and comparing ICA-MLNB with other multi-label classifiers based on more evaluation metrics. It is also possible to extend the ICA-MLNB scheme by studying the effect of CC-ICA in MLNB.

This thesis is mainly about methodological developments. The experimental studies presented in various chapters are based on some public datasets. Clearly, future research may be carried out to apply our proposed methods and algorithms to some real-world applications. Finally, ICA, as a feature extraction method, has been used for different classifiers in addition to naïve Bayes. However, this thesis only investigates the applicability of ICA and its variants for naïve Bayes classifier. Future research may be carried out to explore the use of ICA for more advanced Bayesian classifiers. It would therefore be very meaningful to compare naïve Bayes with other popular classifiers in which ICA is used as a feature extraction method in a more comprehensive manner.

BIBLIOGRAPHY

Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J., 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. PNAS 96, 6745-6750.

Armstrong, S.A., Staunton, J.E., Silverman, L.B., Pieters, R., den Boer, M.L., Minden, M.D., Sallan, S.E., Lander, E.S., Golub, T.R., Korsmeyer, S.J., 2002. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. Nature Genetics 30, 41-47.

Amari, S.I., Cichocki, A., Yang, H.H., 1996. A new learning algorithm for blind source separation. Advances in Neural Information Processing Systems 8, 757-763.

Bach, F.R., Jordan, M.I., 2002. Tree-dependent Component Analysis. In: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence.

Bach, F.R., Jordan, M.I., 2003a. Beyond independent components: Trees and clusters. Journal of Machine Learning Research 4, 1205-1233.

Bach, F.R., Jordan, M.I., 2003b. Finding clusters in independent component analysis. In: Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation, pp. 891-896.

Bae, U., Lee T., Lee, S., 2000. Blind signal separation in teleconferencing using ICA mixture model. Electronics Letters 36, 680-682.

Bartlett, M.S., Sejnowski, T.J., 1997. Independent components of face images: A representation for face recognition. In: Proceedings of the 4th Annual Joint Symposium on Neural Computation, Pasadena, CA.

Bell, A., Sejnowski, T., 1995. An information-maximization approach to blind separation and blind deconvolution. Neural Computation 7, 1129-1159.

Bhattacharjee, A., Richards, W.G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E.J., Lander, E.S., Wong, W., Johnson, B.E., Golub, T.R., Sugarbaker, D.J., Meyerson, M., 2001. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. PNAS 98, 13790-13795.

Borgne, H.L., Guerin-Dugue, A., Antoniadis, A., 2004. Representation of images for classification with independent features. Pattern Recognition Letters 25, 141–154.

Boutell, M.R., Luo, J., 2005. Beyond pixels: Exploiting camera metadata for photo classification. Pattern Recognition 38, 935-946.

Boutell, M.R., Luo, J., Shen, X., Brown, C.M., 2004. Learning multi-label scene classification. Pattern Recognition 37, 1757-1771.

Bressan, M., Guillamet, D., Vitria, J., 2001. Using an ICA representation of high dimensional data for object recognition and classification. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, I-1004-I-1009.

Bressan, M., Vitria, J., 2002. Improving naïve Bayes using class-conditional ICA. In: Garijo, F.J., Riquelme, J.C., Toro, M. (eds.): Advances in Artificial Intelligence - IBERAMIA 2002, pp. 1-10. Springer-Verlag, Berlin.

Bressan, M., Vitria, J., 2003. On the selection and classification of independent features. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 1312-1317.

Brida, J.G., Gómez, D.M., Risso, W.A., 2009. Symbolic hierarchical analysis in currency markets: An application to contagion in currency crises. Expert Systems with Applications 36, 7821-7828.

Cao L.J., Chong, W.K., 2002. Feature extraction in support vector machine: a comparison of PCA, KPCA and ICA. In: Proceedings of the 9th International

Conference on Neural Information Processing (ICONIP'OZ), vol. 2, pp. 1001-1005.

Cardoso, J.F., Laheld, B.H., 1996. Equivariant adaptive source separation. IEEE Transactions on Signal Processing 44, 3017-3030.

Chen, X., 2006. Margin-based wrapper methods for gene identification using microarray. Neurocomputing 69, 2236-2243.

Chen, X., Jing, Z., Xiao, G., 2007. Nonlinear fusion for face recognition using fuzzy integral. Communications in Nonlinear Science and Numerical Simulation 12, 823-831.

Chen, Y., Hsu, C., Chou, S., 2003. Constructing a multi-valued and multi-labeled decision tree. Expert Systems with Applications 25, 199-209.

Cheng, J., Liu, Q., Lu, H., 2004. Texture classification using kernel independent component analysis. In: Proceedings of 17th International Conference on Pattern Recognition, pp. 23-26.

Cheng, W.W., Hullermeier, E., 2009. Combing instance-based learning and logistic regression for multilabel classification. Pattern Recognition 76, 211-225.

Chuang, C.F., Shih, F.Y., 2006. Recognizing facial action units using independent component analysis and support vector machine. Pattern Recognition 39, 1795-1798.

Comon, P., 1994. Independent component analysis: A new concept. Signal Processing 36, 287-314.

de Carvalho, A.C.P.L.F., Freitas, A.A., 2009. A tutorial on multi-label classification techniques. In: A. Abraham et al. (Eds.), Foundations of Computational Intelligence, Vol. 5, SCI 205, pp. 177-195.

Deniz, O., Castrillon, M. Hernandez, M., 2003. Face recognition using independent component analysis and support vector machines. Pattern Recognition Letters 24, 2153-2157.

Ding, C., Peng, H., 2005. Minimum redundancy feature selection from microarray gene expression data. Journal of Bioinformatics and Computational Biology 3, 185-205.

Domingos, F., Pazzani, M., 1997. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 103-130.

Donato, G., Bartlett, M., Hager, J., Ekman, P., Sejnowski, T., 1999. Classifying facial actions. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 974-989.

Elisseeff, A., Weston, J., 2002. A kernel method for multi-labelled classification. In: T.G. Dietterich, S. Becker and Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems, Vol. 14, MIT Press, Cambridge, pp. 681-687.

Fan, L., Poh, K.L., 2007. A comparative study of PCA, ICA and class-conditional ICA for naïve Bayes classifier. Lecture Notes in Computer Science (LNCS) 4507, 16-22.

Fan, L., Poh, K.L., 2008. Improving the naïve Bayes classifier. In: J.R.R. Dopico, J. Dorado, A. Pazos (eds.), Encyclopedia of Artificial Intelligence, pp. 879-883. IGI Publishing.

Fan, L., Poh, K.L., Zhou, P., 2009. A sequential feature extraction approach for naïve Bayes classification of microarray data. Expert Systems with Applications 36, 9919-9923.

Fan, L., Poh, K.L., Zhou, P., 2010. Partition-conditional ICA for Bayesian classification of microarray data. Expert Systems with Applications 37, 8188-8192.

Fortuna, J., Schuurman, D. Capson, D., 2002. A comparison of PCA and ICA for object recognition under varying illumination. In: Proceedings of 16th International Conference on Pattern Recognition, pp. 11-15.

Fortuna, J., Capson, D., 2004. Improved support vector classification using PCA and ICA feature space modification. Pattern Recognition 37, 1117 – 1129.

Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian network classifiers. Machine Learning 29, 131-163.

Gilmore, E., Frazier, P., Chouikha, M., 2004. An independent component analysis based image classification scheme. In: Proceedings of IEEE International Conferernce on Acoustics Speech and Signal Processing, pp. 577-580.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531-537.

Gordon, G.J., Jensen, R.V., Hsiao, L.L., Gullans, S.R., Blumenstock, J.E., Ramaswamy, S., Richards, W.G., Sugarbaker, D.J., Bueno, R., 2002. Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. Cancer Research 62, 4963-4967.

Govindan, A., Deng, G., Kalman, J., Power, J., 1998. Independent component analysis applied to electrogram classification during atrial fibrillation. In: Proceedings of International Conference on Pattern Recognition, pp. 1662-1664.

Guan, A.X. Szu, H.H., 1999. A local face statistics recognition methodology beyond ICA and/or PCA. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN'99), Vol. 2, pp. 1016-1021.

Gurwicz, Y., Lerner, B., 2005. Bayesian network classification using spline-approximated kernel density estimation. Pattern Recognition Letters 26, 1761-1771.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157-1182.

Guyon, I., Weston, J., Barnhill, S., 2002. Gene selection for cancer classification using support vector machines. Machine Learning 46, 389-422.

Hall, M., 2007. A decision tree-based attribute weighting filter for naïve Bayes. Knowledge-Based Systems 20, 120-126.

Hashimoto, W., 2002. Separation of independent components from data mixed by several mixing matrices. Signal Processing 82, 1949-1961.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning (2nd ed.). New York: Springer.

Haykin, S., 1999. Neural Networks: A Comprehensive Foundation. Prentice Hall, New Jersey.

Herrero G.G., Gotchev, A., Christov, I., Egiazarian, K., 2005. Feature extraction for heartbeat classification using independent component analysis and matching pursuits. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 4, pp. 725-728.

Hoya, T., Hori, G., Bakardjian, H., Nishimura, T., Suzuki, T., Miyawaki, Y., Funase, A., Cao, J., 2003. Classification of single trial EEG signals by a combined principal + independent component analysis and probabilistic neural network approach. In: Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), pp. 197-202.

Hoyer, P., Hyvärinen, A., 2000. Independent component analysis applied to feature extraction from colour and stereo images. Network: Computation in Neural Systems 11, 191-210.

Hsu, C.W., Lin, C.J., 2002. A comparison of methods for multi-class support machines. IEEE Transactions on Neural Networks 13, 415-425.

Huang, X., Pan, W., 2003. Linear regression and two-class classification with gene expression data. Bioinformatics 19, 2072-2078.

Hyvärinen, A., Hoyer, P.O., Inki, M., 2001a. Topographic independent component analysis. Neural Computation 13, 1527–1558.

Hyvärinen, A., Karhunen, J., Oja, E., 2001b. Independent Component Analysis. John Wiley & Sons, New York.

Hyvärinen, A., Oja, E., 1997. A fast fixed-point algorithm for independent component analysis. Neural Computation 9, 1483-1492.

Hyvärinen, A., Oja, E., 2000. Independent component analysis: algorithms and applications, Neural Networks 13, 411-430.

Jain, A.K., Duin, P.W., Mao, J., 2000. Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 4-37.

Jain, A., Huang, J., 2004a. Integrating independent components and linear discriminant analysis for gender classification. In: Proceedings of the 6th International Conference on Automatic Face and Gesture Recognition, pp. 159-163.

Jain, A., Huang, J., 2004b. Integrating independent components and support vector machines for gender classification. In: Proceedings of the 17th International Conference on Pattern Recognition, pp. 558-561.

Jutten, C., Herault, J., 1991. Blind separation of sources, part 1: an adaptive algorithm based on neuromimetic architecture. Signal Processing 24, 1-10.

Kapoor, A., Bowles, T., Chambers, J., 2005. A novel combined ICA and clustering technique for the classification of gene expression data. In: Proceedings of IEEE

International Conference on Acoustics, Speech, and Signal Processing, Vol. 5, pp. 621 – 624.

Karvonen, J., Simila, M., 2001. Independent component analysis for sea ice SAR image classification. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, pp. 1255-1257.

Kerr, G., Ruskin, H.J., Crane, M., Doolan, P., 2008. Techniques for clustering gene expression data. Computers in Biology and Medicine 38, 283-293.

Kelemen, A., Zhou, H., Lawhead, P., Liang, Y., 2003. Naïve Bayesian classifier for microarray data. In: Proceedings of the International Joint Conference on Neural Networks, pp. 1769-1773.

Kim, K.J., Cho, S.B., 2004. Prediction of colon cancer using an evolutionary neural network. Neurocomputing 61, 361-379.

Kim, K.J., Cho, S.B., 2006. Ensemble classifiers based on correlation analysis for DNA microarray classification. Neurocomputing 70, 187-199.

Kim, K.J., Choi, S., 2006.Tree-dependent components of gene expression data for clustering. Lecture Notes in Computer Science (LNCS) 4132, 837–846.

Kim, T.K., Kim, H., Hwang, W., Kittler, J., 2004. Independent component analysis in a local facial residue space for face recognition. Pattern Recognition 37, 1873 – 1885.

Kocsor, A., Tóth, L., 2004. Application of kernel-based feature space transformations and learning methods to phoneme classification. Applied Intelligence 21, 129-142.

Kolenda, T., Hansen, L., Larsen, J., Winther, O., 2002. Independent component analysis for understanding multimedia content. In: Proceedings of 12th IEEE Workshop on Neural Networks for Signal Processing, pp. 757-766.

Kotani, M., Ozawa, S., 2005. Feature extraction using independent components of each category. Neural Processing Letters 22, 113-124.

Kwak, N., Choi, C.H., Choi, J.Y., 2001. Feature extraction using ICA. Lecture Notes in Computer Science (LNCS) 2130, 568-573.

Kwak, N., Choi, C., Ahuja, N., 2002. Face recognition using feature extraction based on independent component analysis. In: Proceedings of International Conference on Image Processing, pp. 337-340.

Kwah, N., Choi, C., 2003. Feature extraction based on ICA for binary classification problems. IEEE Transactions on Knowledge and Data Engineering 15, 1374-1388.

Kwak, N., 2008. Feature extraction for classification problems and its application to face recognition. Pattern Recognition 41, 1701-1717.

Langley, P., Iba, W., Thompson, K., 1992. An analysis of Bayesian classifiers. In: Proceedings of the Tenth National Conference on Artificial Intelligence, p. 223-228. AAAI Press, San Jose, CA.

Laubach, M., Shuler, M., Nicolelis, M., 1999. Independent component analyses for quantifying neuronal ensemble interactions. Journal of Neuroscience Methods 94, 141-154.

Li, Z., He, Y., Chu, F., 2005. Application of the blind source separation in machine fault diagnosis: a review and prospect. Mechanical Systems and Signal Processing 13, 1-3.

Lee, S.I., Batzoglou, S., 2003. Application of independent component analysis to microarrays. Genome Biology 4, R76.

Lee, T.W., Lewicki, M.S., 2002. Unsupervised image classification, segmentation, and enhancement using ICA mixture models. IEEE Transactions on Image Processing 11, 270-279.

Lee, T.W., Lewicki, M.S., Sejnowski, T.J., 2000. ICA mixture models for unsupervised classification of non-Gaussian classes and automatic context switching in blind signal separation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 1078-1089.

Leo, M., D'Orazio, T., Distante, A., 2003. Feature extraction for automatic ball recognition: comparison between wavelet and ICA preprocessing. In: Proceedings of 3rd International Symposium on Image and Signal Processing and Analysis, pp. 587-592.

Liu, K.H., Li, B., Wu, Q.Q., Zhang, J., Du, J.X., Liu, G.Y., 2009a. Microarray data classification based on ensemble independent component classification. Computers in Biology and Medicine 39, 953-960.

Liu, K.H., Li, B., Zhang, J., Du, J.X., 2009b. Ensemble component selection for improving ICA based microarray data prediction models. Pattern Recognition 42, 1274-1283.

Melissant, C., Ypma, A., Frietman, E.E., Stam, C.J., 2005. A method for detection of Alzheimer's disease using ICA-enhanced EEG measurements. Artificial Intelligence in Medicine 33, 209-222.

Oliveira, P.R., Romero, R.A.F., 2004. Enhanced ICA mixture model for unsupervised classification. Lecture Notes in Artificial Intelligence (LNAI) 3315, 205-214.

Park, H.S., Yoo, S.H., Cho, S.B., 2007. Forward selection method with regression analysis for optimal gene selection in cancer classification. International Journal of Computer Mathematics 84, 653-668.

Peng, H., Long, F., Ding, C., 2005. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1226-1238.

Pérez, A., Larrañaga, P., Inza, I., 2009. Bayesian classifiers based on kernel density estimation: Flexible classifiers. International Journal of Approximate Reasoning 50, 341-362.

Prasad, M.N., Sowmya, A., Koch, I., 2004. Feature subset selection using ICA for classifying emphysema in HRCT images. In: Kittler, J., Petrou, M., Nixon, M.S. (eds.): Proceedings of the 17th International Conference on Pattern Recognition, pp. 515-518.

Sanchez-Poblador, V., Monte-Moreno, E., Solé-Casals, J., 2004. ICA as a preprocessing technique for classification. Lecture Notes in Computer Science (LNCS) 3195, 1165-1172.

Sandberg, R., Winberg, G., Bränden, C., Kaske, A., Ernberg, I., Cöster, J., 2001. Capturing whole-genome characteristics in short sequences using a naïve Bayesian classifier. Genome Research 11, 1404-1409.

Shah, C.A., Arora, M.K., Robila, S.A., Varshney, P.K., 2002. ICA mixture model based unsupervised classification of hyperspectral imagery. In: Proceedings of the 31st Applied Imagery Pattern Recognition Workshop, pp. 29-35.

Shah, C.A., Watanachaturaporn, P., Varshney, P.K., Arora, M.K., 2003. Some recent results on hyperspectral image classification. In: Proceedings of the IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data, pp. 346 – 353.

Shah, C. and P. Varshney, 2004. A higher order statistical approach to spectral unmixing of remote sensing imagery. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, 1065-1068.

Smith, L.I., 2002. A Tutorial on Principal Component Analysis. Available at: http://users.ecs.soton.ac.uk/hbr03r/pa037042.pdf.

Stone, J., 2004. Independent Component Analysis: A Tutorial Introduction. The MIT Press.

Suri, R., Syst, I., Torrance, C., 2003. Application of independent component analysis to microarray data. In: Proceedings of International Conference on Integration of Knowledge Intensive Multi-Agent Systems, pp. 375-378.

Szu, H., 2002. Unsupervised classification by spectral ICA. In: Proceedings of the 9th International Conference on Neural Information Processing, 1760-1765.

Tsoumakas G., I. Katakis, 2007. Multi-label classification: An overview. International Journal of Data Warehouse and Mining, Vol. 3, No. 3, pp. 1-13.

Vitria, J., Bressan, M., Radeva, P., 2007. Bayesian classification of cork stoppers using class-conditional independent component analysis. IEEE Transactions on Systems, Man and Cybernetics C37, 32-38.

Widodo, A., Yang, B.S., Han, T., 2007. Combination of independent component analysis and support vector machines for intelligent faults diagnosis of induction motors. Expert Systems with Applications 32, 299-312.

Yang, Y., Qiu, Y., Lu, C., 2005. Automatic target classification experiments on the MSTAR SAR images. In: Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05), pp. 2-7.

Yang, Y., Webb, G.I., 2002. A comparative study of discretization methods for naïve Bayes classifier. In: Proceedings of PKAW 2002, The 2002 Pacific Rim Knowledge Acquisition Workshop, Tokyo, Japan, pp. 159-173.

Yu, S.N., Chou, K.T., 2008. Integration of independent component analysis and neural networks for ECG beat classification. Expert Systems with Applications 34, 2841-2846.

Yu, S.N., Chou, K.T., 2009. Selection of significant independent components for ECG beat classification. Expert Systems with Applications 36, 2088–2096.

Zhang, M.L., Pena, J.M., Robles, V., 2009. Feature selection for multi-label naïve Bayes classification. Information Sciences 179, 3218-3229.

Zhang, M., Wang, Z.J., 2009. MIMLRBF: RBL neural networks for multi-instance multi-label learning. Neurocomputing 72, 3951-3956.

Zhang, M.L., Zhou, Z.H., 2007. ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition 40, 2038-2048.

Zhang X., Ramani, V., Long, Z., Zeng, Y., Ganapathiraju, A., Picone, J., 1999. Scenic beauty estimation using independent component analysis and support vector machines. In: Proceedings of 1999 IEEE Southeastcon, pp. 274-277.

Zheng, C.H., Huang, D.S., Shang, L., 2006. Feature selection in independent component subspace for microarray data classification. Neurocomputing 69, 2407-2410.