STATISTICAL PATTERN RECOGNITION: LOCALITY-PRESERVING EMBEDINGS AND ENSEMBLE OF RULES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTING
OF HONG KONG POLYTECHNIC UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Ben Niu

Student ID: 03901735R

November 2007

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(signed)

_____NIU Ben_____(signed)

Acknowledgements

First of all, I would like to thank my academic advisors, Professor Simon Chi-keung Shiu and Professor Sankar Kumar Pal. I feel very fortunate and honored for having had the opportunity to work with them. I thank Prof. Shiu and Prof. Pal from the bottom of my heart for their support, patience, and devotion, and for putting so much effort and time into guiding me.

I would also like to thank Dr. Jian Yang sincerely for his initiative and guidance. Part of the dissertation is based on a paper with Jian, and he brought to me many insightful views that benefited my research and writing.

I would like to thank Professor Qiang Yang, for his tremendous help and guidance in fulfilling my PhD thesis. Prof. Yang made my life in Hong Kong a delightful experience of improvement.

I would like to thank Professor Hong-Va Leong for being the chairman of my oral exam. Prof. Leong is one of the nicest persons you could ever meet, and I look forward to doing postdoctoral research with him.

Finally, I want to express my deepest gratitude and love for my wonderful parents for their love, care, and support throughout the years.

Table of Contents

List of Figures

List of Tables

Database Supports for High-Dimensional Data Indexing: Locality-based Embeddings and Ensemble of Rules

Abstract

Database Management Systems (DBMS) create index structures to help with data retrieval and analysis. Developing the state-of-the-art indexing techniques is the fundamental issue in computer science and engineering.

A critical problem to be tackled in data indexing concerns the high dimensionality of the data. As we know that for low-dimensional data with only a small number of attributes, one can simply index with a key attribute in one-dimensional space. Further, for multi-dimensional data with lower hundred attributes, spatial indexing methods, e.g. KD-tree, can be employed to speedup the searching and retrievial based on similarity metrics. However, in many real world applications, such as time series analysis, content based multi-media retrieival, Microarray analysis, and brain electroencephalogram maps, the data we encounter can be of much higher dimensions, say, thousands of attributes. The traditional methods like B-tree and KD-tree loose their effectiveness due to the curse-of-dimensionality.

To address this difficulty, we can reduce the dimensionality by either feature extraction or selection. By feature extraction, we seek the embedding subspaces for optimal transform to get the low-dimensional representation (the index features) of the original data. By feature selection we choose only a subset of the attributes, the significant features, and use them to characterize the distribution of the data samples. Among these, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are popular for feature extraction. They have been successfully applied to facial image databases for biometric authentication. The optimal embedding subspaces are known as Eigenfaces and Fisherfaces for PCA and LDA respectively. Also, they prove to work effectively on the handwritten digit databases for postcode recognition. On the other hand, the feature selection methods, such as the decision tree and its variants, reduce the dimensionality without transforms, thus avoiding the loss of the semantic meaning of the attributes in the problem domain. The selected features and the rules extracted from the decision trees are therefore human understandable, and helpful to make interpretations on the processes underlying the data.

Recent advances in dimensionality reduction highlight three techniques, namely, the image-based projection, the locality-based embedding, and the ensemble of decision trees. By image-based projection, we needn't have to convert the 2-dimensional input images (or other types of samples given in the matrix form) to the 1-dimensional vectors before the training step. Moreover, it can reduce significantly the size of the matrix in the associated eigenproblems. Based on this technique the 2-dimensional PCA (2DPCA) and

LDA (2DLDA) have been developed, and evaluated on a variety of the facial image databases. The experimental results indicate that the computational efficiency and the recognition accuracy are both improved dramatically.

Locality-based embedding on the other hand is a natural extension of the manifold learning methods, which stress to preserve the local information of data density during embedding. The key idea is that we can infer the global structure of the dataset by maximally preserving the patterns of the data in local neighborhoods. The index features extracted describe the local geometry of the samples, desirable for the nearest neighbor based searches which rely mainly on the local information for retrieval. By locality-based embedding, the recently proposed Laplacianfaces method proves to outperform the PCA and the LDA on facial data. It has aroused considerable interests as an alternative to support high-dimensional data indexing.

Another technique is the ensemble of decision trees, specifically, the CS4 algorithm. This method involves the construction and utilization of a committee of trees, instead of only one tree, for feature selection and prediction. It is plausible in Bioinformatics applications as it can derive the product rules that are biologically meaningful to interpret and guide the wet lab experiments by biologists. Moreover, it turns out to be more accurate than the traditional decision tree method in cancer prediction based on Microarrays.

In this work, we contribute to develop 4 new techniques for dimensionality reduction based on the above mentioned methods.

Firstly, we develop 2-dimensional Laplacianfaces. It is known that 2DPCA, 2DLDA and Laplacianfaces can outperform PCA and LDA respectively. However, it is unclear whether the Laplacianfaces, by leveraging image-based projection, can be further improved against 2DPCA and 2DLDA. Thereby, we develop the 2D Laplacianfaces by intergrating the two techniques, locality preserving and image-based projection. The algorithm is evaluated on the facial image databases, FERET and AR. We find that the computational efficiency is significantly improved compared with Laplacianfaces. The training time and memory complexity is reduced from $O(m^2 \times n^2)$ to only $O(m \times n)$, where m and n are the number of rows and columns of the sample image. 2D Laplacianfaces is also more accurate than 2DPCA and 2DLDA by utilizing the local information to help with recognition.

Secondly, we develop a technique, unsupervised discriminant projection (UDP). In addition to the local information, we also consider the global information in formulating the optimizing criterion. The goal is set to preserve the local density of the data while maximizing the non-local global scatter. On facial image databases UDP outperforms consistently the Locality Preserving Projections (LPP) and PCA, and outperforms LDA when the training sample size per class is small. On the PolyU Palmprint database, UDP achieves the top query accuracy of 99.7%, compared with 86.0%, 95.7%, and 99.0% of PCA, LDA and Laplacianfaces.

Thirdly, based on the idea of locality preserving we also propose another method called Mutual Neighborhood based Discrimiant Projection (MNDP). As the errors of classification derive mainly from the samples on the class boundaries, it is important that the geometry of the class boundary should be better preserved for projection. We construct the mutual neighborhoods to highlight those samples that are on the boundary and most likely contribute to the prediction errors. The features extracted from the facial and the handwritten database indicates that it is significantly better than PCA and LDA when the size of the training sample is modest. The problem of singularity in LDA can also be successfully addressed.

Finally, besides 2D Laplacianfaces and UDP, we also present an algorithm, adaptive CS4 (ACS4), for feature selection and prediction. ACS4 can identify the most discriminant features and grow a committee of trees for prediction. We evaluate ACS4 on the biology database for DNA methylation analysis. The number of the index features for cell line classification is reduced significantly from 49 to only 2. The computational and the wet-lab costs for cancer diagnosis can thus be reduced by about 20 times in contrast to the previously reports. Meanwhile, we also propose a strategy for adaptive clustering on the gene methylation database. The results of clustering confirm that DNA methylation plays the dominant role in the process of tumourgenesis and embryogenesis in human cells.

Chapter 1. Introduction

Over the past few years, Web Search Engines, Internet File Sharing, XML Databases, Biometrics, and Bioinformatics have aroused considerable interests in Computer Science, Molecular Biology, and Software Engineering. These technologies not only facilitate our daily lives but also excite new ideas in fundamental research. A central problem in Database and Internet Information Retrieval is the management of high-dimensional data. Consider that in biomedical research the human genome sequence consists of about 3 billion letters of A, G, C, and T. A typical SNP array contains more than 500,000 testing locis to be analyzed. The access, the storage and the exchange of such high dimensional data is chanlleging.

To address the difficulty, we can reduce the dimensionality of the data while avoiding the loss of the useful information. A spectrum of techniques for dimensionlity reduction has been developed over the past few years. They fall into two categories, feature extraction and feature selection. The algorithms for feature extraction consist of Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), manifold learning, Locality Preserving Projections (LPP), and Two-dimensional PCA (2DPCA), while feature selection covers the ensemeble of trees and the association rules in our context.

Based on these methods, we develop 3 approaches for feature extraction. They are the Two-dimesional Laplacianfaces, Unsupervised Discriminant Projection, and Mutual Neighborhood based Discriminant Projection. They have been applied to the facial image databases for evaluation. Also, we propose an algorithm for feature selection, called adaptive CS4 (ACS4) and apply it analyze the profile of DNA methytion in human cancer and embryonic stem cells. Extensive experiments have been performed to demonstrate the effectiveness of the proposed methods.

1.1 Background

XML Databases, Internet Information Retrievial, Biometrics, and Bioinformatics are the leading technologies underlying the next generation of Internet applications. Intelligent features will add more efficiencies, flexibilities and reliabilities to the information system to guide users' consumption by satisfying their information needs.

Extensible Markup Lanaguage (XML) language is a simple, very flexible text format to facilitate the exchanges of a wide variety of data on the web. XML databases allow data to be imported, accessed and exported in the XML format so that different information systems can share the structured data via the Internet. XML files can contain not only the text data, but also the serilized multimedia information, like graphics, images, audios and videos. XML databases are more suitable in handling the data of complex structures compared with the relational databases. This makes them ideal for advanced applications such as Computer Aided Design (CAD), Digital Entertainments, Geographical Information System, and e-Commerce (for details please see appendix). As XML is

mainly designed to facilitate data exchange, the techniques of dimensionality reduction is thus important to save the cost for data transfer.

We can also use dimensionality reduction to help with Internet File Sharing. Distributed computing allows us to share file and data over Internet for free (freedom and free-of-charge). Though there are still some legal arguments, many leading IT organizations, companies, and Universities have began to invest fundings into this research area. Several prototype systems have been established to enable the Peer-to-Peer data searching and sharing. Current infrastructure for indexing in P2P network are still based on the text keywords, the audio and video features, for example, the text in an image or the facial image of a movie character, cannot be utilized directly to help with data retrieval. Hopefully, we expect that dimensionlity reduction and pattern recognition can be combined together to offer users with more flexibility and usability in P2P data sharing.

Biometrics is the area where dimensionality reduction is most extensively used. Iris, palmprint, and face are the physical traits characterizing the human identity. In biometric databases it is necessary to reduce the dimensionality of the sample images so that the traditional indexing methods can be utilized to accelerate the searching process for real-time authentication. For Biometric problems, recognition accuracy is the most important performance indicator of the system. It is ideal that one can achieve the top accuracy using the fewest features.

There are more than one thousand Bioinformatic databases in the public domain. The data are of great value for biomedical researches. The Gene Expression Ominibus (GEO) Mircroarray database maintains the data of gene expression gathered from 6,839 wet-lab experiments and 174,333 samples. A big part of the data show the gene expression levels in human diseases. The array data are high-dimensional, containing typically the expression dosage of hundreds of genes. With the advance of the Gene Chip technology, the dimensionality can increase further as more genes are being included for testing. An important task in array analysis is to detect the genetic patterns that are associated with the human diseases. By feature selection, the genes that are irrelevant to the diseases can be filtered, thus the relation among the marker genes can be detected. Based on the patterns of gene expression we can interpret how the diseases arise from the aberrant changes of DNA.

1.2 Research problems

There are a number of methods for feature extraction and selection. Principal Component Analysis is an unsupervised method. It doesn't utilize the class information to help with dimensionality reduction. The optimizing criterion of PCA is to find a subspace for projection where the variance of the data can be maximized. The projected data are decorrelated thus the redundant information can be removed. PCA is popular due to its mathematic simplicity and computational efficiency. It proves to be the method that is most energy compact with the minimal reconstruction error. It is optimal for data representation. When applied to human facial image data, PCA is well known as the

Eigenfaces method. PCA however is unsupervised. It is sensitive to the variation of the lighting conditions, facial expressions and poses in face recognition. It is widely accepted that PCA is more suitable for data compression than for pattern recognition.

Linear Discriminant Analysis in contrasted to PCA is a supervised method for feature extraction. It seeks the subspace for projection where the between-class scatter can be maximized while the within-class scatter is minimized. The optimizing criterion of LDA reflects actually the Bayesian error of classification. By maximing the LDA criterion the error classification can be minimized. LDA has been applied to face recognition and compared with PCA. When the size of the training data is large enough the class centers can be estimated accurately and LDA can outperform PCA significantly. PCA and LDA analyze the global patterns of the training data for feature extraction. The local geometry around the training samples is not well preserved after reducing the dimensionality. The nearest neighbor based classifiers however rely heavily on the local information for classification.

Manifold learning, i.e., Locally Linear Embedding, Isometric Mapping, and Laplacian Eigenmap assume that the distribution of the samples is on the manifold. LLE finds the low dimensional representation of the samples so they can be reconstructed by its closest neighbors in the feature space. Isometric mapping is an extension of Multi-Dimensional Scaling (MDS) where the affinity among the training samples is measure by the geodesic distance, i.e., the length of the shortest path on the neighborhood graph. Laplacian Eigenmap preserves the local density by imposing a penalty function such that if the training samples are near neighbors in the original space they are kept still close to each other in the feature space. Manifold learning is better than PCA and LDA because it captures the nonlinear structure of the sample data. However, it doesn't provide an explicit transform for dimension reduction when the test samples are presented. To overcome this problem the Laplacianfaces method has been developed.

Laplacianfaces extends the idea of manifold learning. It preserves the locality of the sample space and constructs the projections for dimension reduction on new testing samples. On the facial image data, Laplacianfaces outperforms PCA, LDA in accuracy. It has also been utilized for handwritten character recognition. The classes are better separated with Laplacianfaces method than with PCA and LDA.

1.3 Methodology and results

To reduce the computational complexity of Laplacianfaces, we first develop a method called 2D Laplacianfaces by leveraging the technique of image based projection. This technique has been successfully applied to PCA and LDA generating 2DPCA and 2DLDA. The size of the matrix in the associated eigen equation decreases dramatically from $O(m^2 \times n^2)$ to only $O(m \times n)$, where *m* and *n* are the number of rows and columns of the sample image. 2D Laplacianfaces is also more accurate than 2DPCA and 2DLDA by utilizing the local information to help with recognition.

Also, we develop unsupervised discriminant projection (UDP). Besides the local information, we consider the global information in formulating the optimizing criterion. The goal is to preserve the local density of the data while maximizing the non-local global scatter. UDP outperforms consistently the Locality Preserving Projections (LPP) and PCA, and outperforms LDA when the training sample size per class is small on facial data. On the PolyU Palmprint database, UDP achieves the top query accuracy of 99.7%, compared with 86.0%, 95.7%, and 99.0% of PCA, LDA and Laplacianfaces.

Based on the idea of locality preserving we also propose another method called Mutual Neighborhood based Discrimiant Projection (MNDP). As the errors of classification derive mainly from the samples on the class boundaries, it is important that the geometry of the class boundary should be better preserved for projection. We construct the mutual neighborhoods to highlight those samples that are on the boundary and most likely contribute to the prediction errors. The features extracted from the facial and the handwritten database indicates that it is significantly better than PCA and LDA when the size of the training sample is modest. The problem of singularity in LDA can also be successfully addressed.

Finally, we present adaptive CS4 (ACS4) for feature selection and classification. ACS4 can identify the most discriminant features and grow a committee of trees for prediction. We evaluate ACS4 on the biology database for DNA methylation analysis. The number of the index features for cell line classification is reduced significantly from 49 to only 2. The computational and the wet-lab costs for cancer diagnosis can thus be reduced by about 20 times in contrast to the previously reports. Meanwhile, we also propose a strategy for adaptive clustering on the gene methylation database. The results of clustering confirm that DNA methylation plays the dominant role in the process of tumourgenesis and embryogenesis in human cells.

Chapter 2. Problems of indexing in high-dimensional space

Indexing high-dimensional data is essential for large-scale information retrievial and data mining. Consider the human brains, perhaps the most complex and powerful system on earth. It consists of a huge number of neurons, more than 10^{11} at the scale, and each of them is connected with the tens of thousands of others to concert the brain work. It is estimated that there are more than 10^{15} such connections, the adaptation of which raises the high-level human intelligence. Our brains operate efficiently storing and processing the inputs from the biological sensors. Our daily experiences and knowledge are represented in the form of the chemical states of the neurons and their connections, which characterizes the low-dimensional features of the original inputs. They are the index features to recall our memory and experiences and aid our decisions.

It is chanllenging to understand and model the human brains in computers. The current state-of-the-art of computational methods is still quite primitive. Yet, to make machines learn there have been a number of the plausible works developed during the past. For example, Artificial Neural Networks (ANNs) and Decision Trees (DTs) are two successful methods with real world applications. They can be utilized to index the high-dimensional data for classification and clustering. In web mining, biometrics, and bioinformatics, ANNs, DTs and other machine learning methods have gained widespread popularities as database supports for decision making.

In section 2.1, we will first go through several applications where the indexing of high-dimensional data is involved. We then discuss in section 2.2 the problem of curse-of-dimensionality to highlight the necessity of high-dimensional indexing to support database applications.

2.1 Indexing data in high-dimensional space

With the rapid growth of the semiconductor and the software industry, the expenses of the digital equipments, digital cameras, music and digital video players, and the associated storage devices have dropped significantly. This has spawned a variety of applications, such as Multimedia Information Systems, CAD/CAM, Geographical Information Systems, Bio-medical Imaging, Biometrics, Stock Market and Time Serie Analysis. The applications require storing large amounts of high-dimensional data to be later searched, retrieved, and analyzed. The costs of the bandwidth, the CPU and the storage resources rocket high, making it inscalable for large domains, e.g., web search engine. Also, the useful information can be easily overwhelmed in the high-dimensional space by noises, and hardly be detected. To address this problem, we can reduce the dimensionality of data while maxmimally preserving their most useful features, which can be utilized as index to improve the speed of retrieval and accuracy of classification. They also help to detect the hidden patterns of the true processes underlying the data. Here we cover two problems that involve the processing of high-dimensional data in Biometrics and Bioinformatics.

2.1.1 Biometrics

Biometrics studies how to automatically recognize a person based on his/her physiological or behavioral characteristics. Fingerprint recognition systems, for example, have been widely used for years. Other human traits for identity recognition are face, palm print, iris, DNA sequence, speech and signatures, as illustrated in Figure 2.1. Also different biometric schemes can be combined to increase the accuracy of authentication.

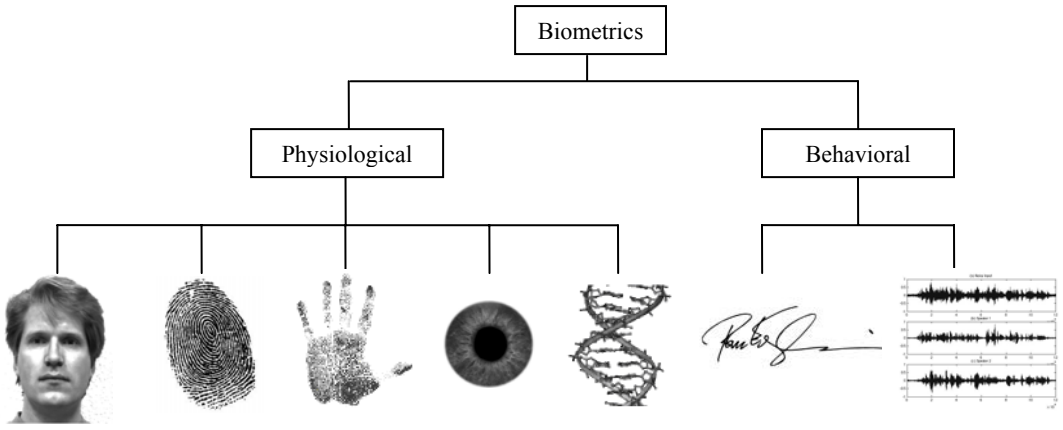


Figure 2. 1 Biometric identity based on human facial features, fingerprint, palmprint, iris, DNA, signature, and voices (from left to right)

Bioinformatics systems are applicable to a wide variety of institutions and organizations. For instance, in constructing the criminal justice system, the U.S. customs departments at the international airports accept the fingerprints as the identity. For military and government services, speech and iris recognition adds a security layer to the sensitive information. Recently, Biometics has found new applications in the World Wide Web, such as search engines, and more recently in digital entertainment. A Bioinformatric system has the typical structure as indicated in Figure 2.2,

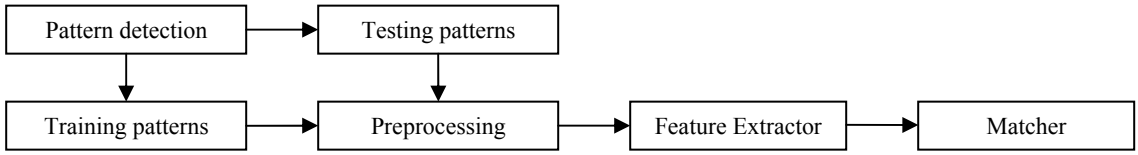


Figure 2. 2 Diagram of Biometric system

Basically, a biometric DBMS manages all the collected biometic patterns. These sample patterns are matched against the inputs to answer the users' queries. There are usually 6 subcomponents in the system. The pattern detection module detects and extracts the patterns from the raw inputs, e.g., it can indentify which part in a given image is a human face, or a road sign. The detected patterns are then segmented and extracted by preprocessing, before taking the steps of feature extraction, and recognition. Then, the set

of the patterns will be split into two parts, one for training the classifier and the other for testing and evaluation.

In biometrics, face detection, face recognition, and face synthesis are the problems in focus. Indexing is highly plausible for such applications because the size of the facial data can be large due to the dimensionality and the huge number of samples. It also requires the real time response to queries, making it necessary to have the compact representation of the images for efficiency.

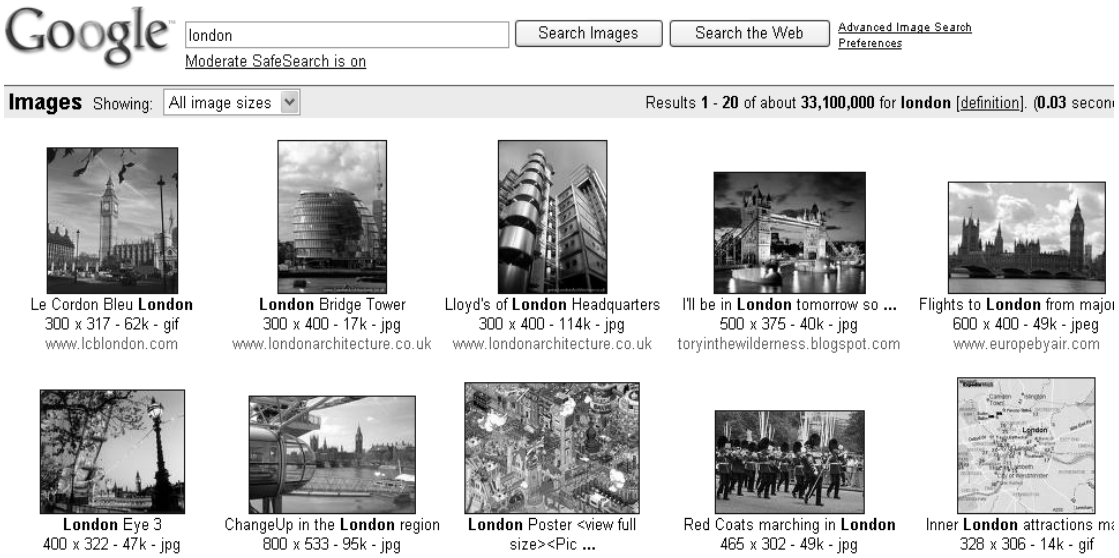
Face detection

Face detection determines the locations and the sizes of human faces in the digital images, ignoring anything else such as buildings, trees, and bodies in the background. Ideally, it can work effectively regardless of the variation of the facil orientations, sizes, colors, and illuminations, as illustrated in Figure 2.3.

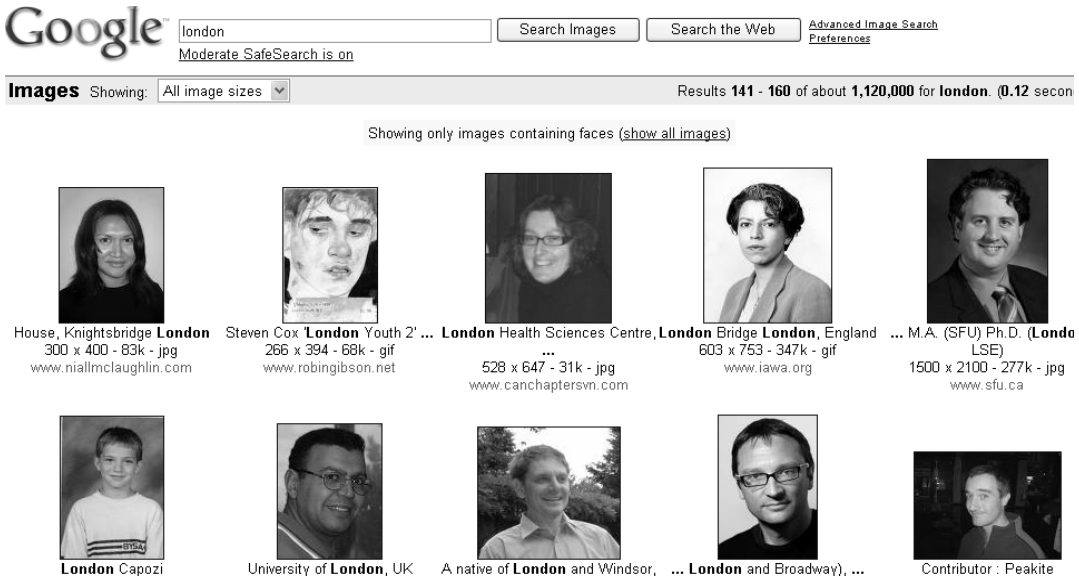
Figure 2. 3 Face detection in gray scale image

The faces are detected. Then, they are segmented out from the image and stored into the database for later use. The face detection method can also be utilized for the content based image retrieval by infering whether an image contains the human face(s). This technique has been recently adopted by the Google search engine to refine its feedbacks to users based on the contents of images. It allows the users to narrow down their searches to only those images that contain the human faces. This is achieved by enabling

the face detection functions, as shown in Figure 2.4 the results returned from the Google search engine.



(a)



(b)

Figure 2. 4 Face detection capability provided by Google Search Engine (a) searching with keyword 'London' returns city photos (b) searching keyword 'London' with face detection enabled returns only facial images.

Figure 2.4 (a) is the case when the keyword, city name 'London', is the query term. None of the returned images contain human faces but buildings, roads, and maps about the city. After face detection is enabled, only those images with human faces are returned, shown in (b). The results are informative to help find out those people who have the links to the city 'London', with their facial identities.

As the size of the Web is huge, so is the number of the images, dimensionality reduction has to be utilized to meet the real-time requirement. The ANN model can be trained for such purpose in face detection. In doing so, 2 datasets, facial, and non-facial, can be created for ANN training and testing. After training, given a testing input, the network fires its neurons through the connections to predict whether the image contains human face(s) or not. The learned connection weights attached to the neurons are the features for classification.

Face recognition

Face recognition is easy for humans, but difficult for machines. One is still capable of recognizing his/her friend without seeing each other over 20 or 30 years. Computers, however, are incapable of doing this. As the complex nature of the neural activities involved in face recognition is still not fully understood, it remains a challenge to the computer scientists to construct the facial recognition systems as competitive as humans. Though still quite primitive, there have been a number of efforts to improve the performance of face recognition, among which Eigenfaces and Fisherfaces are the methods most widely known.

The central problem in face recognition is to achieve the most compact representation of the facial images that can lead to the best performance of recognition. Also, like in face detection, it should be able to handle effectively the environment changes including the variations of illuminations, poses, expressions, and ages. This is trivial for humans but chanllenging for computers. Ideally, they should not degrade the performance of the recognition. Figure 2.5 lists the facial images involving the above mentioned conditions. Figure 2.5 (a) shows the front view of a subject. Computers can extract the features of the image by using either the Eignfaces or Fisherfaces method. Yet, as the pose of the face shifts toward left, Figure 2.5 (b, d, e), or right (c, f), the low dimensional feature of the images can loose tis effectiveness and be biased for recognition.

When the facial images are rendered with both the texture and the color, as shown in Figure 2.6, dimensionality reduction can be performed on each of them first. The obtained features containing both the texture and the color information can then be combined and utilized for recognition.

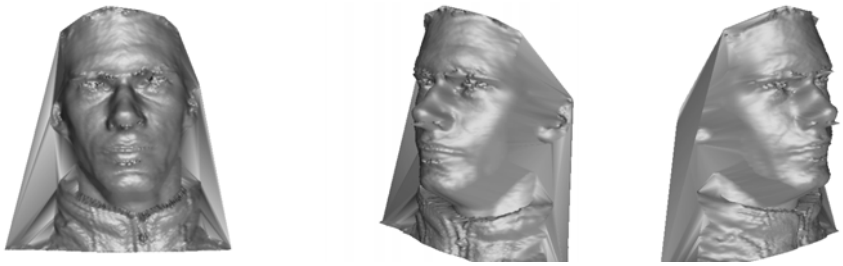
(a) (b) (c)

(d) (e) (f)

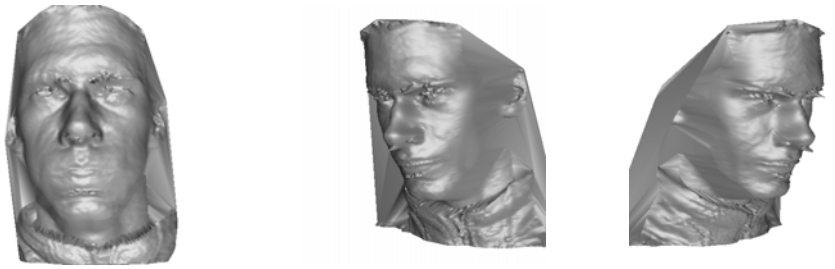
Figure 2. 5 Two dimensional grayscale sample face images from M2VTS database (a) Frontal view (b) Slight left view (c) Slight right view (d) Full left view (e) Left view (f) Full right view

Figure 2. 6 A sample of two dimensional facial image with texture.

A critical problem in face recognition based on 2D still image is that pose variations can radically change the recognition rates. Features extracted from the frontview of the images cannot be successfully applied to recognize the same person in the sideviews to the left or right. This problem can be avoided by adopting the 3 dimensional depth images. By adding one more dimension to feature extraction, we can rotate the facial images to register them to a template for pose normalization. Figure 2.7 demonstrates the 3D facial images of an individual in various poses. The Iterative Closest Point (ICP) algorithm widely used in computer graphics can then be employed to normalize the faces. Hence, feature extraction can be taken in a similar way for 2D faces. Likewise, the textures and the colors can be combined with the geometry of the depth images to train the classifier, as shown in Figure2.8 of a 3D facial image with the texture and the color maps.



(a) (b) (c)



(d) (e) (f)

Figure 2. 7 Three dimensional facial images without texture. (a) Frontal view (b) Upper left view (c) Upper right view (d) With Facial Expression (e) Lower left view (f) Lower right view

Figure 2. 8 A sample of three dimensional facial image with texture.

Though 3D tends to be better than 2D in handling poses, we still lack the knowledge whether the human vision system work truely in 3D. Moreover, capturing the 3D face images in outdoor from faraway is still quite difficult at the present stage due to the limits of sensors. In addition, efficient algorithm has to be developed and implemented to enable the fast normalization of high-resolution 3D point clouds. The problem of 3D face recognition is still challenging, yet it has arrested intensive attentions in recent years in biometrics. We can expect that the next generation of face recognition systems will be a composition of the advanced sensory technologies for face image production and detection, highly efficient graphic algorithms for face registration, and the effective algorithms for face feature extraction and recognition.

Face (character) synthesis in digital entertainments

Digital entertainment is a highly profitable industry with billions of dollars of investiments every year, and it is still growing fast. A major type of the computer games involves the online role playing, where game players are the virtual characters with custom-designed physical lookings, such as the equipments they have, the clothes they wear, and the hair styles. This flexibility makes the game more immersive and playable, giving the full experience to users for fun, and the game producers for market.

Particulary, the industry is starting to integrate the facial elements into the games' virtual world to add more reality. A recent example is given by the release of the online Golf game 'Tiger Woods PGA Tour 08'. Game players are allowed to upload their 2D digital images to the game website, where the game software can synthesize and convert them to the 3D face models. The depth facial images are then assembled with the avatars to

produce the virtual characters that represent the players in the virtual world, as shown in Figure 2.9.

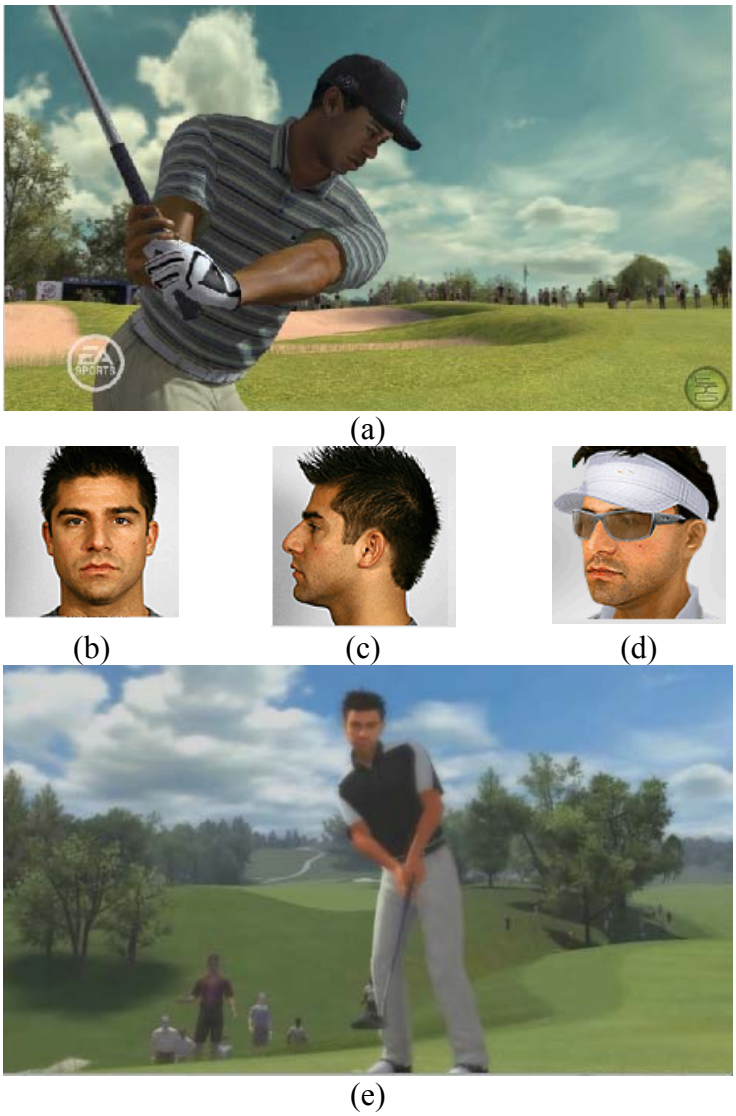


Figure 2. 9 (a) Tiger Wood 3D image designed and implemented by artists in the game (b, c) frontal and side images uploaded by the a game player (d) 3D face models synthized from 2D images (e) Player in the game

As more and more players join the game online, there can be millions of users world-wide with their established 3D face images in the database. Thus, it is intriguing to collect the facial data from the virtual-world, extract the discriminant features for indexing, storage, transmission, and matching. Game users hence can search over the Internet based on the physical appearances of other game players to either team-up or make friends in the virtual world.

2.1.2 Microarray data analysis

Bioinformatics is another research area where we have to deal with the high-dimensional data. Human genome consists of the 23 pairs of chromosomes containing about 3 billion base pairs of A, T, C, and G. These letters encode the instructions of human life, the genes. Genes are transcribed and translated to the proteins which are the building blocks of the cells in human body. In Figure 2.10, we indicate the central dogma in molecular biology to show how this happens. Fundamentally, DNA is transcribed to RNA. Some of the RNAs are the messangers to be translated into the proteins. Some others act as the machinearies to interact with the proteins and DNAs to regulate the transcription of DNA. Proteins can also inter-play with DNA to modulate the expressions of genes. And DNA can duplicate during cell proliferations.

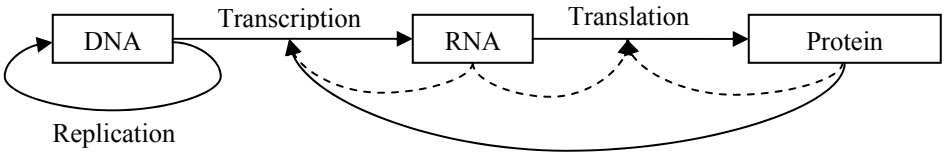


Figure 2. 10 Central dogma of molecular biology

Many types of human diseases, cancers, PD (Parkinson's disease), heart strokes, and hyper tensions, are susceptible to the genetic mutations. These malfunctioned genes are aberrently suppressed for expression or they are unable to be repaired from the damages in the sequence. The proteins thereby cannot be produced for normal cell functions. For instance, cancer cells are most likely depleted of the immunological genes. Without the control, the cells can reproduce infinitely and develop into tumors, whereas for the cells with the genes a procedure called apoptosis can be started to kill the tumor cells by mediating them to suicide.

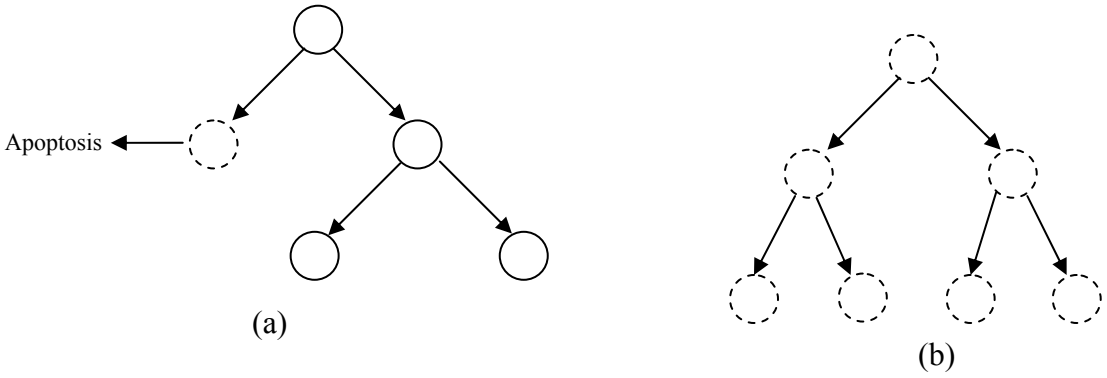


Figure 2. 11 Differentiation of normal and cancer cells (a) normal cells mediate Apoptosis to stop tumor progress (b) cells with defected genes reproduce infinitely to become tumors

There are several types of Microarray technologies to measure the cells' molecular activities. Gene expression array tests the dosages of the gene expressions by reading the RNA level. Protein array quantifies the intensity of proteins to compare the functions of the cells. SNP array tests whether there is any mutational change at some specified loci on the chromosomes. As there are about 20,000 to 30,000 human genes and millions of SNPs in the genome, the array data are usually of high-dimensions and small-sample-size,

as shown in Figure 2.12 and 2.13 the gene expression and the SNP arraies, where each spot represents a gene or SNP.

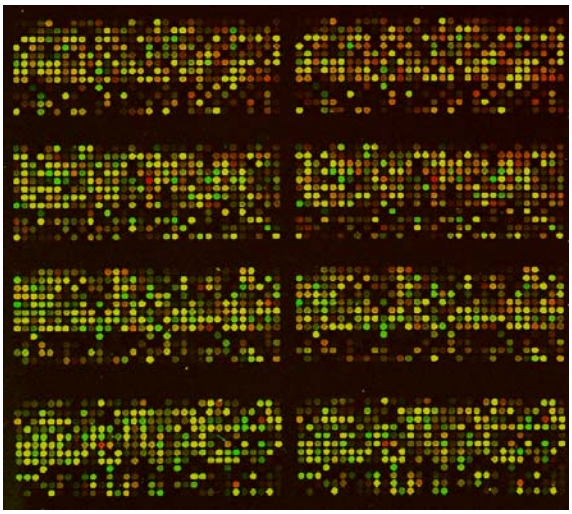


Figure 2. 12 A sample image of gene expression array

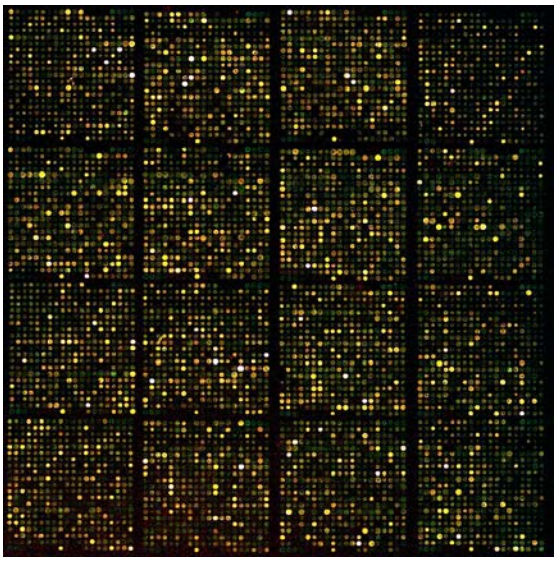


Figure 2. 13 A sample image of SNP array

It is important to identify the genes and the SNPs associated with the human diseases. As the dimensionality is high and the sample size is small, it is chanllenging to detect the useful genetic patterns buried in the high-dimensional space for prediction. Clustering analysis ablso becomes difficult due to the noises.

2.2 Problem: KD-tree indexing and curse of dimensionality

To highlight the effects of dimensionality, we can consider the case of k-dimensional tree (KD-tree) for data retrieval. In the KD-tree algorithm, the efficient measurement of similarity between to objects is important. It is analogous to human memory. We can recall the previous experiences, or facts, to support the current decisions based on the similarities between two events or scenarios, like finding the K nearest neighbors of the present case. KD-tree can work efficiently to an extent to help with the classification and retrieval of the similar patterns of users' interest. However, KD-tree is not quite scalable to the dimensionality of data, making it necessary to introduce the techniques for dimension reduction to overcome the curse of dimensionality.

2.2.1 k-Nearest neighbor search

kNN chooses first a similarity metric, usually the person's correlation score, or other types of norms to measure the closeness of 2 instances. Given a query, it calculates its distance to all the samples in the database. The closest ones to the query are the nearest 'neighbors'. They can be used for classification, or simply returned to the users for further analysis.

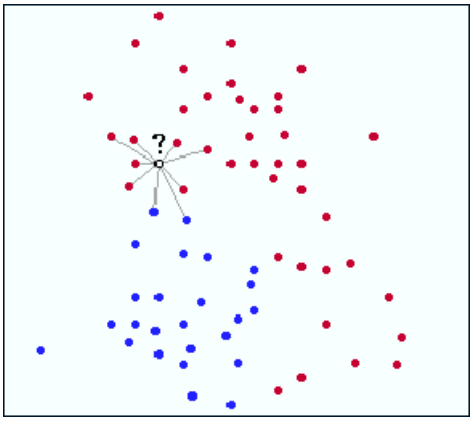


Figure 2. 14 K nearest neighbor classifier

For example, Figure 2.14 illustrates the case where kNN is applied for binary classification in 2-dimensional space. The closest 9 neighbors to the query pattern are found by calculating the similarities. As there are 7 votes from class one, more than 2 from class two, the query sample is classified to class one.

kNN is quite effective for classification on the data set with large number of samples. However, it is a lazy learning method and the query has to be matched against all the training samples for classification. This is time consuming and computationaly expensive. A solution to address this deficiency is to create an index by using the KD-tree.

2.2.2 KD-tree

The kD-tree is a natural generalization of the standard one-dimensional binary search tree. It partitions the searching space by using the splitting planes perpendicular to the coordinate axes. In construting the tree, one goes down from the root to the leaf cycling through the axes to select the point that generates the splitting planes. Given a list of N points, the algorithm to construct the KD-tree is as follows:

```
Algorithm KD-tree Construction
Input: list of points pointList; depth; dimension k
Output: IDs of matched nodes
Begin
  If pointlist is empty
     return nil;
  Else
     axis = dept mod k;
     sort point list on axis and choose median as pivot element;
     select median from pointList;
     create tree node;
     node.location =  median;
     node.leftChild = kdtree(points in pointList before median, depth+1);
     node.rightChild = kdtree(points in pointList after median, depth+1);
     return node;
   Endif
End
```

The time and the storage complexity of KD-tree construction is $O(N \log N)$ and $O(Nk)$ respectively. To find out the nearest neighbors to a query, the root node is examined first. The subtrees containing the target point are then tranversed. Recursively, it browses from the root to the leaf nodes to match. Because in each step of searching a large portion of the samples are discarded, the query time complexity reduces to $O(\log N)$. Nearest neighbor search with KD-Tree can work effectively on the dataset of moderate dimensions, say, about 30 or 40. Yet, as the dimensionality of the sample space rises, it becomes increasingly difficult to get the closest neighbors as the size of the tree become too huge to be searched. The curse of dimensionality has to be addressed to support the database applications (see appendix). The development of the technologies that can reduce the dimensionality of data while retaining their useful features is thus necessary. They should help to reduce the computation time, the storage, and the bandwidth costs, without trading off the accuracy of retrieval. During the past years, 2 techniques have been developed for this purpose, namely, feature extraction and feature selection.

Feature extraction reduces the dimensionality of data through linear transforms. Nonlinear extensions can be made by kernel method. In Chapter 3 and 4, we will present 2 classical techniques for feature extraction. Chapter 3 contributes to two baseline systems, Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA), and their kernel extensions. Some recent works are discussed in Chapter 4 about artificial neural network, manifold learning, locality based embedding, and 2D image based projection. Based on the results in Chapter 3 and 4 we develop 2D Laplacianfaces and evaluate it on the facial image databases.

Chapter 3. Classical methods: PCA and LDA

PCA and LDA are two simple yet powerful methods for feature extraction and dimensionality reduction. PCA seeks the projections that maximize the variance of the data, while LDA optimizes the Fisher's criteria to extract the features that generate the low Bayesian errors in classification.

3.1 PCA and Kernel PCA with applications to face and handwritten digit databases

3.1.1 Method

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

$$C = \frac{1}{M}\sum_{j=1}^{M} x_j x_j^T . \tag{3.1}$$

by solving the eigenvalue equation,

$$\lambda v = Cv . \tag{3.2}$$

The first p eigenvectors with the largest eigenvalues are selected for embedding to extract the features. However, PCA fails to capture the nonlinear structure of the dataset. To address the deficiency kernel PCA (KPCA) is developed by using the kernel based method. In linear PCA, for the eigenvalues $\lambda \geq 0$ and the associated eigenvectors $v \in R^N$,

$$\lambda v = Cv = \frac{1}{M}\sum_{j=1}^{M} (x_j \cdot v) x_j . \tag{3.3}$$

All solutions v with $\lambda \neq 0$ thereby must lie in the span of $x_1,...,x_M$. In KPCA, the dot product is computed in a high dimensional space induced by a nonlinear map,

$$\Phi : R^N \to F, x \mapsto X \tag{3.4}$$

The dimensionality of the feature space F can be infinite. Assume that the data are centeralized, i.e., $\sum_{k=1}^{M} \Phi(x_k) = 0$, the covariance matrix in F thus takes the form,

$$\overline{C} = \frac{1}{M}\sum_{j=1}^{M} \Phi(x_j)\Phi(x_j)^T \tag{3.5}$$

Thereby, we need to find out the eigenvalues λ and eigenvectors $V \in / F - \{0\}$ satisfying,

$$\lambda V = \overline{C} V \tag{3.6}$$

Similar to Eq. 3.3, all solutions V with $\lambda \neq 0$ must lie in the span of $\Phi(x_1),...,\Phi(x_M)$,

$$\lambda V = \overline{C} V = \frac{1}{M} \sum_{j=1}^{M} \left(\Phi(x_j) \cdot V\right) \Phi(x_j). \tag{3.7}$$

So, there are the coefficients α_i ($i = 1,...,M$) satisfying,

$$V = \sum_{i=1}^{M} \alpha_i \Phi(x_i). \tag{3.8}$$

For all $k = 1,...,M$, define an $M \times M$ matrix K by $K_{ij} = \left(\Phi(x_i) \cdot \Phi(x_j)\right)$, the Eigen equation in the projected space with respect to V can be written as,

$$M\lambda K\alpha = K^2 \alpha, \tag{3.9}$$

where α denotes the column vector with entries $\alpha_1,...,\alpha_M$. To find solutions of Eq. (9), we can solve the eigen problem,

$$M\lambda\alpha = K\alpha, \tag{3.10}$$

Let $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_M$ denote the eigenvalues of K, $\alpha^1,...,\alpha^M$ the eigenvectors, with λ_p being the first p leading eigenvalues.

In KPCA, we can compute the nonlinear projections onto the eigenvectors V^k in F ($k = 1,...,p$). Let x be a test point, with an image $\Phi(x)$ in F, then

$$(V^k \cdot \Phi(x)) = \sum_{i=1}^{M} \alpha_i^k (\Phi(x_i) \cdot \Phi(x)), \tag{3.11}$$

Ther are several popular kernel functions in use, e.g., the Gaussian RBF kernel and the polynomial kernel, where c is the window width for the RBF kernel function, and the number of degree for the polynomial kernel.

Gaussian RBF kernel $\quad K(x, y) = \exp\left(\frac{-\|x - y\|^2}{c}\right)$

Polynomial kernel $\quad K(x, y) = [(x \cdot y) + 1]^c$

To summarize, the steps of KPCA are as follows, note that linear PCA is a special case of KPCA when the kernel function is the inner product.

Algorithm

Algorithm Kernel Principal Component Analysis
Input: *M* training samples of *N* dimensions
Output: *M* feature vectors of *p* dimensions
Begin
Compute the centralized kernel matrix $\widetilde{K}$ from the training samples.
Diagonalize $\widetilde{K}$ to extract the principal components $V^1,...,V^M$.
Project the samples onto the first p components to get the p-dimensional feature vectors.
End

To centralize the kernel matrix, consider in the feature space *F*, given any Φ and any set of observations $x_1,...,x_M$, the points,

$$\widetilde{\Phi}(x_i) = \Phi(x_i) - \frac{1}{M}\sum_{i=1}^{M}\Phi(x_i), \tag{3.12}$$

we go on to define the covariance matrix $\widetilde{K}$, where

Computational complexity

Suppose we have M training samples of *N* dimensions and we are going to reduce them to *p* dimensions, the memory cost is $O(M^2)$ to maintain the kernel matrix. Solving the associated equation will take $O(M^2 p)$ time using numerical methods such as SVD or Lanczos' method.

$$\begin{aligned}\widetilde{K}_{ij} &= \left(\widetilde{\Phi}(x_i)\cdot\widetilde{\Phi}(x_j)\right)\\ &= \left(\left(\Phi(x_i) - \frac{1}{M}\sum_{m=1}^{M}\Phi(x_m)\right)\cdot\left(\Phi(x_j) - \frac{1}{M}\sum_{n=1}^{M}\Phi(x_n)\right)\right)\\ &= K_{ij} - \frac{1}{M}\sum_{m=1}^{M}1_{im}K_{mj} - \frac{1}{M}\sum_{n=1}^{M}K_{in}1_{nj} + \frac{1}{M^2}\sum_{m,n=1}^{M}1_{im}K_{mn}1_{nj}\\ &= \left(K - 1_M K - K1_M + 1_M K 1_M\right)_{ij}\end{aligned} \tag{3.13}$$

Thus $\widetilde{K}$ can be computed directly from *K* to solve the eigenproblem. For feature extraction, we compute projections of centered Φ images of test patterns *t* onto the eigenvectors of the covariance matrix of the centered points,

$$(\widetilde{V}^k\cdot\widetilde{\Phi}(t)) = \sum_{i=1}^{M}\widetilde{\alpha}_i^k(\widetilde{\Phi}(x_i)\cdot\widetilde{\Phi}(t)).\,, \tag{3.14}$$

Consider a set of test points $t_1,...,t_L$, and define two $L \times M$ matrices by $K_{ij}^{test} = \left(\Phi(t_i) \cdot \Phi(x_j)\right)$ and $\widetilde{K}_{ij}^{test} = \left(\Phi(t_i) - \frac{1}{M}\sum_{m=1}^{M}\Phi(x_m)\right) \cdot \left(\Phi(x_j) - \frac{1}{M}\sum_{n=1}^{M}\Phi(x_n)\right)$,

thereby,

$$\widetilde{K}^{test} = K^{test} - 1_M^{'} K - K^{test} 1_M + 1_M^{'} K 1_M , \tag{3.15}$$

where $1_M^{'}$ is the $L \times M$ matrix with all entries equal to $1/M$.

3.1.2 Applications

As KPCA finds the directions that maximize the variance of data, it can help to remove the noises from data. We will use first the handwritten digit dataset to demonstrate its effectiveness for denoise. PCA can also be utilized for pattern recognition. It reduces the dimensionality of the input signals while retaining their low-dimnesional features for classification.

Denoise

KPCA is successful when applied to denoise the handwritten digit dataset compared with linear PCA. To do so, the first step is to construct the principal components and the nonlinear projections. The images with noises can then be projected onto the components to get the coefficients of KPCA. Then we select only those leading components, discarding those minor ones to reconstruct the image. The reconstruction procedure is based on the least square method to minimize the information loss between the original and the reconstructed images.

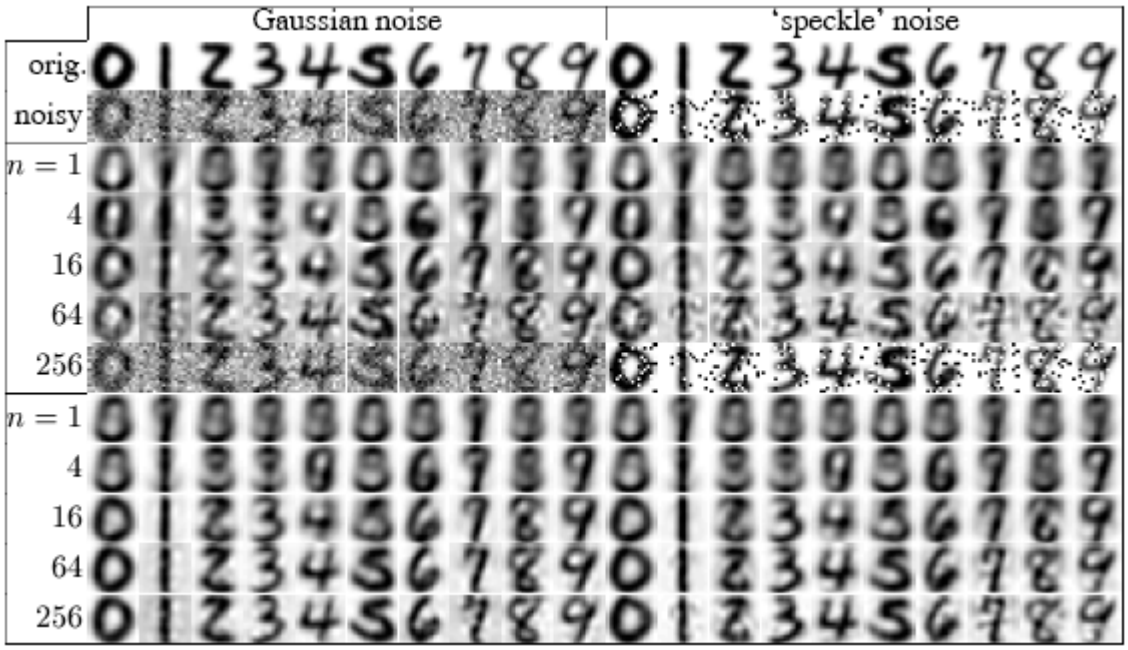


Figure 3. 1 De-Noising of USPS data. The left half: *top*: the first occurrence of each digit in the test set, *second row*: the upper digit with additive Gaussian noise, *following five rows*: the reconstruction for linear PCA using n = 1, 4, 16, 64, 256 components, and, *last five rows*: the results of KPCA using the same number of components. The righft half shows the same but for 'speckle' noise. (Taken from [3.1])

By using the nonlinear kernels, KPCA achieves better results of reconstruction than linear PCA with the same number of components.

Handwritten digit recognition

In addition, KPCA is also widely used for handwritten digit recognition. By reducing the dimensionality of the data, it becomes computationally more efficient to retrieve the matched patterns for recognition. Similar to denoise, in the first step, KPCA projections will be constructed by computing the principal components. All the images are then projected into a lower dimensional space for retrieval. Given an input pattern for testing, it is mapped to the feature space where kNN can match its closest neighbors for classification. When the linear kernels are adopted, the components computed from the handwritten dataset can be viewed as images, as shown in Figure 3.2.

Figure 3. 2 First 10 components computed from the UCI handwritten digit dataset (Taken from [3.2])

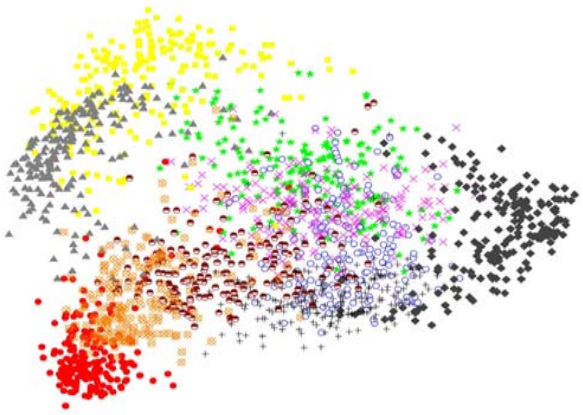


Figure 3. 3 Two dimensional PCA visualization of the training data after projection onto the first 2 components (Taken from [3.3])

Since KPCA is unsupervised, the class information of the data is not fully utilized for training. As illustrated in Figure 3.3., there are overlaps among the digits 1-10, and this will cause prediction errors. To summarize, KPCA is better for data compression and denoise processing but not necessarily suitable for classification.

3.2 LDA and KFDA with applications to face and handwritten digit databases

LDA and its kernel extension KFDA are supervised methods for dimensionality reduction. Different from Kernel PCA, they use the class information to train the models, so that the ratio of the between-class scatter to the within-class scatter can be maximized to reduce the Bayesian error of classification.

3.2.1 Method

Let $X_1 = \{x_1^1,...,x_{l_1}^1\}$ and $X_2 = \{x_1^2,...,x_{l_2}^2\}$ be samples from two different classes. Fisher's linear discriminant (LDA) is given by the vector w which maximizes the fisher's criterion

$$J(w) = \frac{w^T S_B w}{w^T S_W w}, \tag{3.16}$$

where

$$S_B = (m_1 - m_2)(m_1 - m_2)^T, \tag{3.17}$$

$$S_W = \sum_{i=1,2} \sum_{x \in X_i} (x - m_i)(x - m_i)^T, \tag{3.18}$$

are the between and within class scatter matrices respectively and m_i is defined by $m_i = \frac{1}{l_i} \sum_{j=1}^{l_i} x_j^i$. Intuitively, maximizing $J(w)$ is to find a direction which maximizes the distance among the projected class means while minimizing the data variance within each class.

The Fisher's criterion is actually a simplified version of the Baysian error assuming the conditional covariance is normal and equal. Even with these assumptions, it turns out that the simplified criterion is quite effective in real world applications. The fisher's discriminant components w can be obtained by solving the associated equation derived from (16), i.e.,

$$S_W^{-1} S_B \cdot w = \lambda w, \tag{3.19}$$

But real world data can be complex and the linear discriminant method is inadequate to make good separations between classes. To address this deficiency, like what we have done on Kernel PCA, we can play the kernel tricks. In a similiar fashion, the data in the original space will be mapped to a high dimensional space by a nonlinear function.

Let Φ be a nonlinear mapping to the feature space F. To find the linear discriminant projection in F, we need to maximize

$$J(w) = \frac{w^T S_B^\Phi w}{w^T S_W^\Phi w}, \tag{3.20}$$

where $w \in F$, S_B^Φ and S_W^Φ are the corresponding matrices in F, i.e.,

$$S_B^\Phi = (m_1^\Phi - m_2^\Phi)(m_1^\Phi - m_2^\Phi)^T, \tag{3.21}$$

$$S_W^\Phi = \sum_{i=1,2} \sum_{x \in X_i} (\Phi(x) - m_i^\Phi)(\Phi(x) - m_i^\Phi)^T, \tag{3.22}$$

with $m_i^\Phi = \frac{1}{l_i} \sum_{j=1}^{l_i} \Phi(x_j^i)$.

Now, we need to find a formulation for Eq. 3.20 so the dot products of the input patterns can be computed by using the kernel functions. Again, from the theory of reproducing kernels we know that any solution $w \in F$ must lie in the span of all training samples in F. Thereby, we can have an expansion for w in the form,

$$w = \sum_{i=1}^{l} \alpha_i \Phi(x_i), \tag{3.23}$$

Using formula 3.23 and according to the definition of m_i^Φ we have

$$\begin{aligned} w^T m_i^\Phi &= \frac{1}{l_i} \sum_{j=1}^{l} \sum_{k=1}^{l_i} \alpha_j k(x_j, x_k^i) \\ &= \alpha^T M_i \end{aligned}, \tag{3.24}$$

where we can compute $(M_i)_j = \frac{1}{l_i} \sum_{k=1}^{l_i} k(x_j, x_k^i)$ and replace the dot products by the kernel function. Now consider the numerator of (20), by using the definition of S_B^Φ and the relation in formula 3.24 it follows

$$w^T S_B^\Phi w = \alpha^T K_b \alpha, \tag{3.25}$$

where $K_b = (M_1 - M_2)(M_1 - M_2)^T$. Considering the denominator, by using (23) we arrive at

$$w^T S_w^\Phi w = \alpha^T K_w \alpha, \tag{3.26}$$

where $K_w = \sum_{j=1,2} K_j (I - 1_{l_j}) K_j^T$, K_j is a $l \times l_j$ matrix with $(K_j)_{nm} = k(x_n, x_m^j)$, I is the identity and 1_{l_j} the matrix with all entries $1/l_j$. Combining 3.25 and 3.26 we can rewrite Fisher's linear discriminant in F by maximizing

$$J(\alpha) = \frac{\alpha^T K_b \alpha}{\alpha^T K_w \alpha}, \tag{3.27}$$

Eq. 3.27 can be solved by finding the leading eigenvector of $K_w^{-1} K_b$. The projection of a test pattern t onto w is given by

$$(w \cdot \Phi(t)) = \sum_{i=1}^{l} \alpha_i k(x_i, t). \tag{3.28}$$

It should be noted that the matrix K_w can be singular when the number of the training samples is small, and normalization has to be carried out in order to make it invertible to get the solution. There are basically two strategies for this purpose. One is that we simply add a multiple of the identity matrix to K_w, i.e., replace K_w by $\overline{K}_w$, where,

$$\overline{K}_w = K_w + \mu. \tag{3.29}$$

μ is a small positive number.

The other approach is that we can perform KPCA first to remove the null space of K_w, then LDA is carried out in the KPCA transformed space to reduce the dimensionality further. The null space of K_w yet can contain discriminative information and should not be simply discarded. Yang [3.4] et al has recently developed a method to combine the discriminant information derived from both the regular space (with non-zero eigenvalues) and the irregular space (null space) to help with classification.

After we get the discriminant components, we can select those with the largest eigenvalues for feature extraction. To have *p* dimensional feature vectors we can choose *p* fisher components for projections. To summarize, the KFDA method takes the following steps.

Algorithm

Algorithm Kernel Fisher Discriminant Analysis
Input: l_i (i = 1, 2) training samples of N dimensions from 2 classes
Output: l feature vectors of p dimensions
Begin
Compute the between class kernel matrix K_b.
Compute the within class kernel matrix K_w.

Normalize the within class kernel matrix to get $\overline{K}_w$.

Diagonalize $\overline{K}_w^{-1} K_b$ to extract the discriminant components $w^1,...,w^p$.

Project the samples onto the first p components to get the p-dimensional feature vectors.

End

Computational complexity

The memory and time complexity of KFDA is at the same level as KPCA, i.e., $O(L^2)$ and $O(L^2 p)$ respectively. The actual memory consumption of KFDA is twice of that of the KPCA as it has to maintain both the between and the within class kernel matrices, while KPCA needs only to keep the total scatter in memory. To solve the equation, a variety of the numerical methods such as SVD, QR and Lanczos' method can be utilized.

3.2.2 Applications

KFDA have be used widely in pattern recognition and machine learning due to its mathematical simplicity and global optimality. It is well accustomed to several problems including face recognition and handwritten digit prediction.

Face recognition

When LDA is applied to human facial data, it is called Fisherfaces, which is an important method in Biometrics. Specifically, compared with the eigenfaces method based on linear PCA, LDA proves to be more robust to the variations of pose, facial expression, and particularly the illumination of the facial images.

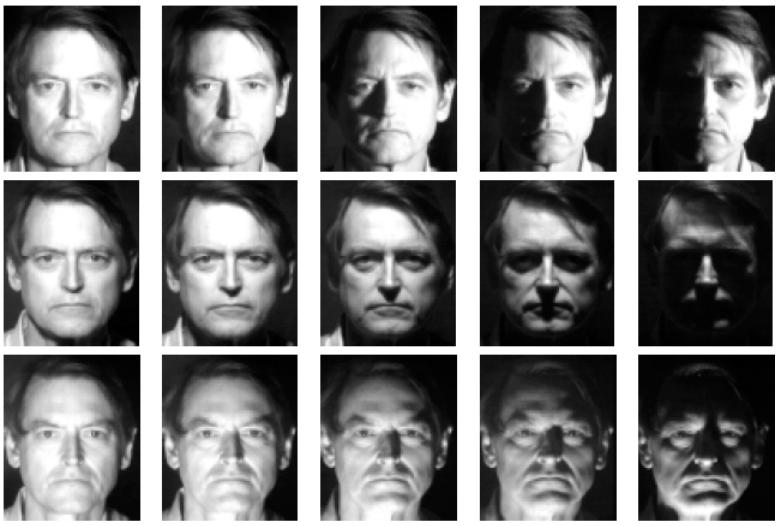


Figure 3. 4 Sample images from the Harvard Database with variations of illuminations (Taken from [3.5])

Experiment results indicate that Eigenfaces and Fisherfaces both perform perfectly when lighting is nearly frontal, as shown in Figure 3.4, however, as lighting is moved off the axis, there is a significant difference between the performances of the two methods. The Fisherfaces method produces lower error rates than the Eigenfaces.

Handwritten digit prediction

Yang et al. [3.4] experimented on the Concordia University CENPARMI handwritten numeral database as shown in Figure 3.5 to compare KFDA with LDA. The input images include 256-dimensional Gabor transformation features. 100 samples are randomly chosen from each of the ten classes for training, while the remaining 500 samples are used for testing. They run the system 10 times for evaluation. Their results indicate that KFDA is significantly bettern than LDA in accuracy.

Figure 3. 5 Sample images from the CENPAMI handwritten digit database

KPCA and KFDA are two statistical methods for dimensionality reduction and feature extraction. KPCA is well suited for data compression, representation and denoising, while KFDA by virtue of class information can extract the features more suitable for classification. They have been successfully applied to solve the handwritten digit and the face recognition problems. The kernel extension of LDA, KFDA, can outperform significantly LDA. On human facial database, LDA turns out to be more robust than PCA in handling the illuminations.

Recently, several nonparametric methods have been proposed for dimensionality reduction and feature extraction as new alternatives. They are the GA based ANN, the manifold learning, the locality based embedding, and the 2D image projections, based on which we developed our own methods.

Chapter 4. Recent advances: GA-based neural networks, Manifold Learning, locality based embedding, and 2D image projections

In this chapter, we investigate 4 approaches recently developed for dimensionality reduction. Artificial Neural Networks can adapt the strength of its connections among the neurons to represent the high-dimensional inputs in a compact form, much like how the human brains work. The integration of the genetic algorithm with ANNs computationally mimics the evolution of our neural systems in struggling against the natural environment for survival and development. Manifold learning, locally linear embedding, and Isometric Mapping, has aroused considerable interests recently in the area of cognitive science and computation. It establishes a theoretical framework based on the concept of locality preserving. Extensions have been made by leveraging the local features of data to help solve the recognition problems. Moreover, the advances in image based projection allow us to reduce significantly the memory and the computational complexities of PCA and LDA. Based on these approaches, particularly manifold learning and image based projection we develop a new method 2D Laplacianfaces in Chapter 5.

4.1 Construction of neural networks: an evolutionary approach

The human brains are the highly optimized machinnaries of evolution. They control the central nervous system (CNS), by way of the cranial nerves and spinal cord, the peripheral nervous system (PNS). They regulate virtually all human activities ranging from the lower unconscious actions such as heart rate, respiration, and digestion and higher civilized mental activities such as thought, reason, and abstraction.

Physically, the human brains work through the collaborations of the neurons firing in concert for actions. Recent functional magnetic resonance imaging (fMRI) technology has shown us how the neural systems operate and correlate with each other for functions in spatial time, as illustrated in Figure 4.1. Evidences indicate that the approximately 100 billion neurons in human brain are partitioned into the functional groups and these regional functional components are not independent from each other. Figure 4.2 illustrates the active areas of human cerebral when a blind subject read Braille characters. It is intriguing that the vision domain is also highly active for blind subjects in addition to the language sections. This implies that the human cognition process is highly complex and involves multiple functional groups to co-work in human behavior.

Due to this complexity it is still a long way from the full understanding of the brain model by computational means. But plausible efforts have been made by computer scientists and neurologists to establish the simplified models of human brains *in-silico*.

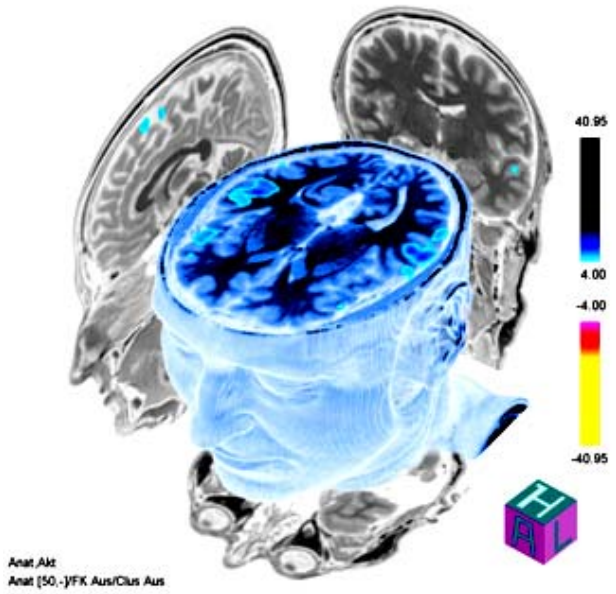


Figure 4. 1 3D FMRI image of neural activities in human brain (Taken from [4.1])

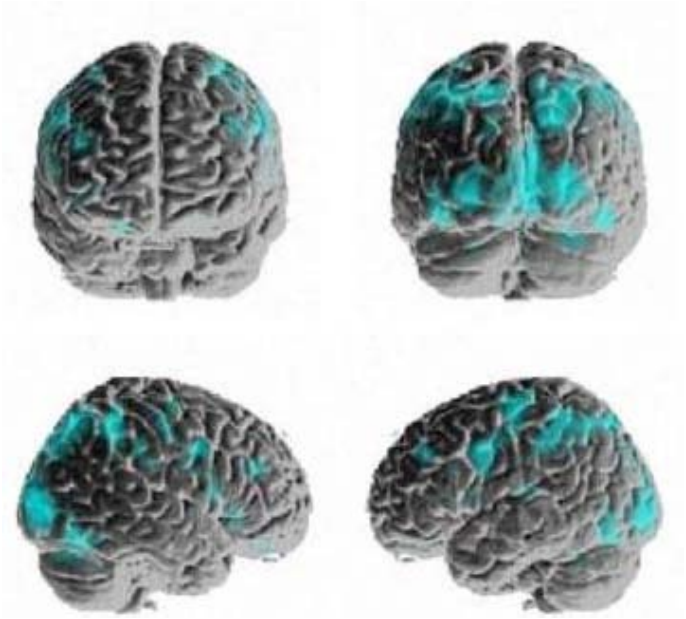


Figure 4. 2 FMRI image of blind subjects reading Braille characters (Taken from [4.2])

Researchers have developed the Artificial Neural Networks (ANNs) which can learn the connection weights among the neurons to represent our daily experiences and knowledge. Though still quite primitive at the present stage ANNs can solve the problems such as image compression and object recognition effectively. It is important to understand how neural networks evolve and diversify in nature. It is already known that the genetic factors, such as DNA mutaions, must have played an important role in upgrading our

neural abilities over the other species for civilization. Figure 4.3 illustrates how the brain capacity of Homo Sapiens and its ancestors changes gradually under the force of natural selection in 4 millions of years. The genetic changes contribute a lot to the more advanced human brains and the intelligence.

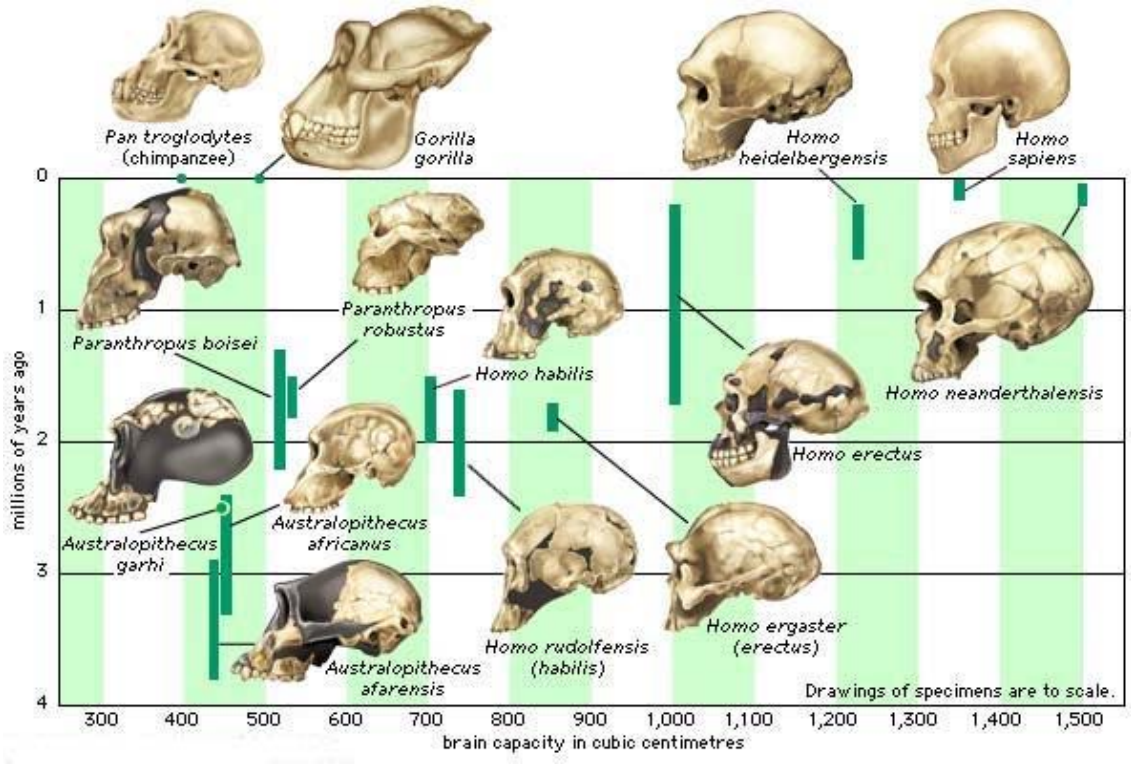
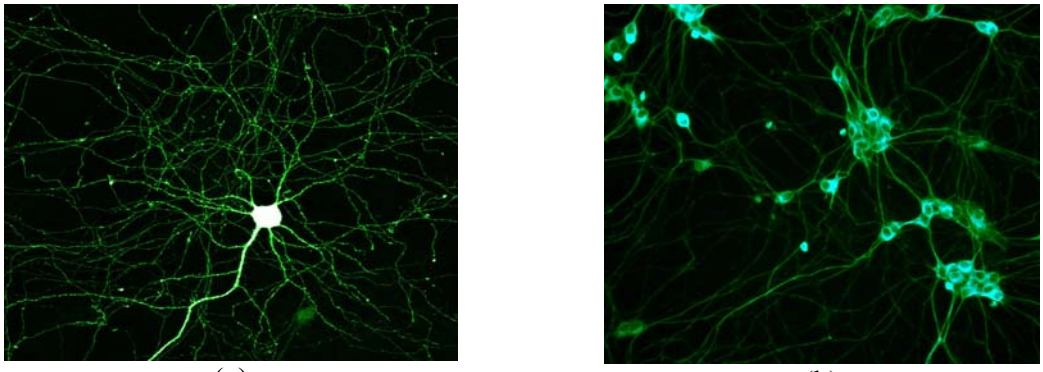


Figure 4. 3 Evolution of brain capacity (Taken from [4.3])

It is thus intriguing to combine the state-of-the-art evolutionary computation, e.g., genetic algorithm, with the ANNs to help understand how the neural networks can advance to the current stage in natural evolution.

4.1.1 Neural networks for compression and object recognition

The Neural networks consist of a huge number of neurons that are connected with each other for perception, memory, reasoning and other human activites, as illustrated in Fig 4.4 (a) and (b) the basic structures of the neural nets. For vision problems, it is believed that vision nerve cells can process the visual stimuli in 2 modes. In the passive mode, the neurons sense the input signal and decompose it into subbands, like taking a Gabor transform [4.4]. The response coefficients encoding the frequency and the orientation are transmitted to the cerebral for memory and further processing. In the active mode, we need to tell the difference between 2 similar objects. The cerebral actively instructs the nerves to focus and extract the discriminant features for memory and better accuracy of recognition.



(a) (b)

Figure 4. 4 Image of neurons (a) a single neuron (b) inter-connected neurons (Taken from [4.5])

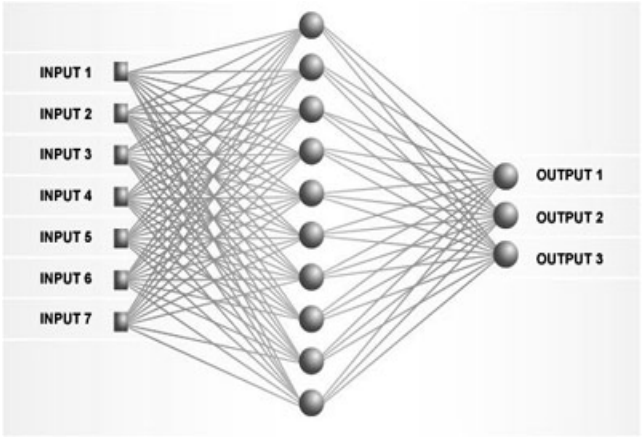


Figure 4. 5 Artificial neural network with input, hidden and output layers (Taken from [4.6])

Figure 4.5 depicts the structure of a typical artifical neural. The input can be the Gabor filtered features or the raw input images, the outputs are usually the class labels for classification or the attribute values to be predicted. The first layer in the network takes the input. The hidden layer contains the computing units and forwards the result to the third layer, the output layer. The connections among the neurons are trained to reduce the error between the actual and the expected outputs. There are a variety of training algorithms for training neural networks. Two popular ANN models are the Back Propagation and the Radial Basis Function nets, which utilize the gradient descent rule, as shown Eq. 4.1, to learn the weights of connections.

$$\begin{aligned} w_{ij} &\leftarrow w_{ij} + \Delta w_{ij} \\ \Delta w_{ij} &= -\mu \frac{\partial E}{\partial w_{ij}}. \end{aligned} \tag{4.1}$$

where w_{ij} is the weight between the *i*-th and the *j*-th neuron, μ is the step length, E is the error function.

ANNs have been successfully applied to solve real world problems. In handwritten digit recognition, custom designed multilayer ANNs by human experts can work effectively and outperforms Support Vector Machines [4.7] in accuracy. Also, it can be applied to face recognition. RBF neural networks [4.8] can be trained very fast to reduce the dimensionality of data for recognition.

4.1.2 Training neural networks by evolutionary computation

Genetic algorithm [4.9] basically involves 4 opertors: selection, crossover, mutation, and reproduction. It is established based on the law of 'survival the fittest'. During the millions years of evolution, the neural systems of the species diversified to the environmental changes for survival and development. This adaptation is stimulated under the force of both the genetic variations and the natural selections, where the nerual system is highly optimized to achieve the fitness of better life. A computational model can be much helpful to highlight this process by applying the genetic algorithm to the construction of the ANNs.

Pal et al. [4.10] presented an evolutionary design of neural networks and demonstrate its effectiveness for classification tasks. The fundamental idea is to encode the connection weights of the network as the chromosomes for selection, crossover, mutation and reproduction. The fitness values of the chromosomes are calculated based on the accuracy performance of classification.

Chromosome representation

The connection weights of the neural network are encoded into a binary string of 16 bit length, where [000...0] decodes to -128 and [111...1] decodes to 128. The chromosome is formed by concatenating all the strings. A population of 64 chromosomes was initialized.

Crossover

Because the length of the chromosome is large, single point crossover is not effective. Multiple point crossover is adopted, where the distance between two crossover points is a random variable between 8 and 24 bits and the crossover probability is fixed at 0.7.

Mutation

The mutation rate is allocated according to the fitness value. For the chromosomes already good enough the mutation probability is set as low as 0.01, and for those of low fitness value it is as high as 0.4, where 1 is switch to 0 and vice versa.

Choice of fitness function

The fitness of the chromosome is calculated based on the accuracy of classification f ,

$$f = \frac{No.\ of\ Correctly\ Classified\ Sample\ in\ Training\ Set}{Total\ No.\ of\ Samples\ in\ Training\ Set}. \qquad (4.2)$$

After a number of iterations, the training procedure of the network is terminated with the accuracy of classification being above a specified threshold. It is generally accepted that GA based ANN is better than those trained with the gradient methods only if the size of the problem is very large. In other words, it can converge faster and be more optimized when the number of the neurons in the network is large. For problem involving only a small number of neurons it is better to use the gradient based method to converge faster and generate more optimized results.

4.2 Isometric mapping

Human brains can extract from its high-dimensional sensory inputs-30,000 auditory nerve fibers or 10^6 optic nerve fibers-a manageably small number of perceptually relevant features. Manifolding learning seeks to discover the nonlinear degrees of freedom that underlie complex natural observations, such as human handwriting or images of a face under different viewing conditions. In the following sections, we investigate 3 manifold learning methods, Isometric Mapping (Isomap), Locality based Linear Embeddings (LLE), Laplacian Eigenmap (LE).

A manifold is a topological space which is locally Euclidean. Generally, any object which is nearly 'flat' on small scales is a manifold, Figure 4.6 shows an example.

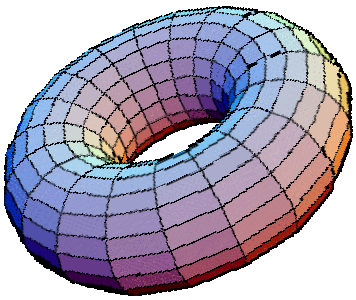


Figure 4. 6 A manifold surface in 3-dimensional space (Taken from [4.11])

The problem of manifold learning is defined as follows:

Let $Y \subset R^d$ be a low dimensional manifold, $f : Y \to R^D$ is a continuous mapping, where $D > d$. The dataset $\{y_i\}$ is randomly generated, and $\{x_i = f(y_i)\}$. The problem of manifold learning is to reconstruct $\{y_i\}$, when $\{x_i\}$ is given.

Isomap is in fact a natural extension of the classic Mutli-dimensional scaling (MDS) algorithm, which is a general technique for display D-dimensional data in 2 dimensional space. It is simple to implement, efficiently computable, and guarantee to discover the true structure of data lying on or near a linear subspace of the high dimensional input space. Classical MDS finds an embedding that preserves the interpoint distances, equivalent to PCA when those distances are Euclidean. However, in real world applications many data sets contain essential nonlinear structures that are invisible to MDS or PCA, as shown in Figure 4.7 the challenge of nonlinearity with data on a 2-dimensional 'Swiss roll'. The Isomap method has to be proposed to address this deficiency.

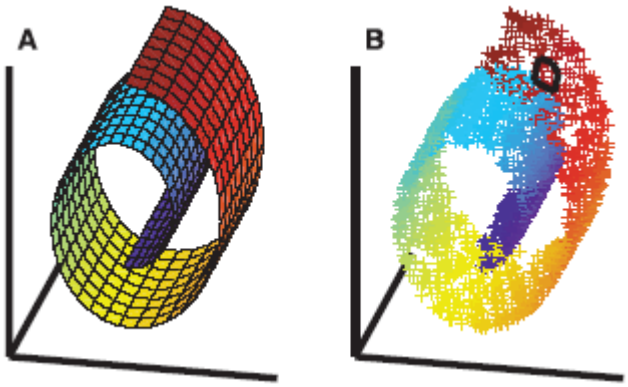


Figure 4. 7 Swiss roll (Taken from [4.12])

The key idea of Isomap is the utilization of the geodesic manifold distance instead of the Euclidean metric to reflect the true low dimensional geometry of the manifold. For neighboring points, input space distance provides a good approximation to geodesic distance. For faraway points, geodesic distance derived from the shortest path can be approximated by adding up a sequence of 'short hops' between neighboring points. A distance matrix based on the computation of the geodesic distances is then constructed based on which the classical MDS is carried out. The complete isometric feature mapping has three steps as summarized below.

Algorithm

Algorithm Isometric Mapping
Input: l training samples of N dimensions
Output: l feature vectors of d dimensions
Begin

Define the graph G over all data points by connecting points i and j if they are mutually the closest neighbors.

For any two nodes on G, x and y, compute the shortest path between them, the distance $d(x, y)$ is the length of the path. Construct the distance matrix $\tau(G)$

Let λ_p be the p-th eigenvalue (in decreasing order) of the matrix $\tau(G)$, and v_p^i be the i-th component of the p-th eigenvector. Then set the p-th component of the d-dimensional coordinate vector equal to $\sqrt{\lambda_p} v_p^i$.

End

By utilizing the geodesic distance the nonlinear structure of the data on the manifold can be preserved after projection. In Figure 4.8 (a) and (b), the swiss roll data cannot be unfolded with the traditional linear method such as PCA and LDA. This problem can be solved by using the Isomap method. The result after projecting the Swill roll data onto a 2 dimensional space is shown in Figure 4.8 (c).

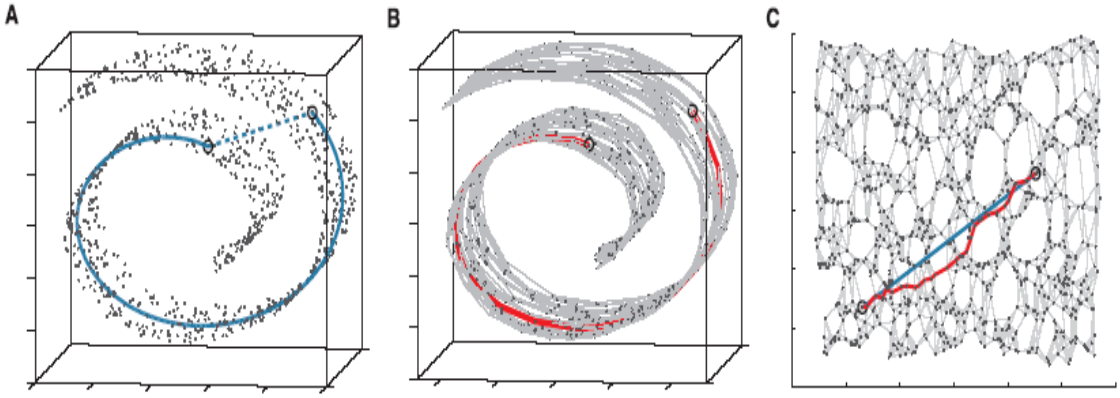


Figure 4. 8 Swiss roll unfolded using Isomap. Red path: approximation of the shortest path to compute the geodesic distance, Blue path: the shortest path on the manifold (Taken from [4.13])

4.3 Locally linear embedding

Another important method for unsupervised nonlinear dimensionality reduction on manifold is locally linear embedding (LLE). The fundamental idea of LLE is to compute a neighborhood-preserving embeddings that is globally optimal. By exploiting the local symmetries of linear reconstructions, LLE is able to learn the global structure of nonlinear manifold, such as those generated by images of faces or documents of text. Different from Isomap as presented in Section 4.2 LLE recovers the global nonlinear structure from locally linear fits, eliminating the need to estimate the pairwise distances between widely separated data points. LLE is based on simple geometric intuitions. Suppose the data consist of N real valued vectors X_i, each of dimensionality D, sampled from some underlying manifold. Provided there is sufficient data, and each data point and its neighbors lie on or close to a locally linear patch of the manifold. The local geometry of these patches can be characterized by linear coefficients that reconstruct each data point from its neighbors. Reconstruction errors are measured by the cost function,

$$\varepsilon(W) = \sum_i \left| X_i - \sum_j W_{ij} X_j \right|^2 . \tag{4.3}$$

which adds up the squared distanes between all the data points and their reconstructions. The weights W_{ij} summarize the contribution of the *j*-th data point to the *i*-th reconstruction. The weights W_{ij} can be computed by minimizing the cost function subjecto to two constraints: first, that each data point X_i is reconstructed only from its neighbors, enforcing $W_{ij} = 0$ if X_j does not blong to the set of neighbors of X_i; second that the rows of the weight matrix sum to one: $\sum_j W_{ij} = 1$. The optimal weights W_{ij} subject to these constraints are found by solving a least-squares problem. The constrained weights that minimize these reconstruction errors for a data point are invariant to rotations, rescalings, and translations of that data point and its neighbors. LLE can construct a neighborhood-preserving mapping to reduce the dimensionality of data. After we get the weights W_{ij} which reflect the intrinsic geometric properties of the data, we can use them to seek the optimal LLE transform. In the final step of the algorithm, each high-dimensional observation X_i is mapped to a low-dimensional vector Y_i representing global internal coordinates on the manifold. This is done by choosing *d*-dimensional coordinates Y_i to minimize the embedding cost function,

$$\Phi(Y) = \sum_i \left| Y_i - \sum_j W_{ij} Y_j \right|^2 . \tag{4.4}$$

The cost function is based on locally linear reconstruction errors, but here what to be optimized is no the weights W_{ij} but the coordinates Y_i. Eq. 4.4 defines a quadratic form in the vectors Y_i, which can be minimized by solving a sparse $N \times N$ eigenvalue problem, whose bottom *d* nonzero eigenvectors provide an ordered set of orthogonal coordinates centered on the origin. Implementation of the algorithm is straightforward. The data points were reconstructed from their *K* nearest neighbors, as measured by Euclidean distance or normalized dot products. Once neighbors are chosen the optimal weights W_{ij} and coordinates Y_i are computed by standard methods in linear algebra. The LLE and the Isomap methods share a common feature in that both of them characterize the global geometry of data by examining the local patterns of the manifolds. One advantage of LLE over Isomap is that it avoids the needs to find the shortest path to compute the geodesic distance. So for sparse data where the distance derived from the shortest path can be highly biased LLE tends to be more effective. Figure 4.9 illustrates the idea of 'locality persevering' in LLE.

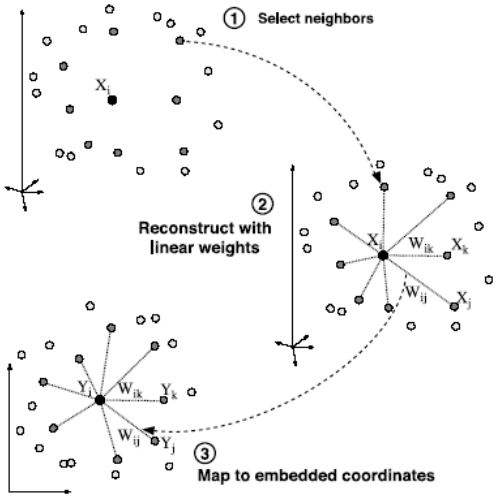


Figure 4. 9 Data points are constructed from their closest neighbors on the manifold (Taken from [4.12])

The Swiss roll can be unfolded successfully with LLE, as shown in Figure 4.10

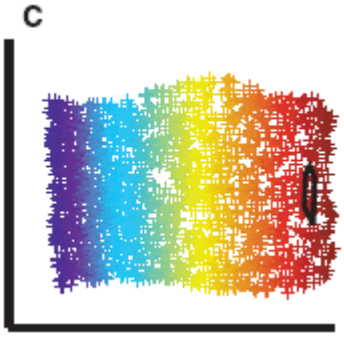


Figure 4. 10 Swiss roll unfolded with LLE (Taken from [4.12])

Algorithm

Algorithm Locally linear embedding
Input: *l* training samples of *N* dimensions
Output: *l* feature vectors of *d* dimensions
Begin
Identify the *K* closest neighbors for each point in the training set.

Optimize to compute the weights W in the error function $\varepsilon(W) = \sum_i \left| X_i - \sum_j W_{ij} X_j \right|^2$.

Fix the weight W, compute the first d coordinates Y by minimizing the reconstruction error. $\Phi(Y) = \sum_i \left| Y_i - \sum_j W_{ij} Y_j \right|^2$

End

On facial data, LLE method can identify the two dimensions, facial expression and pose, to characterize the distribution of the images, as illustrated in Figure 4.11.

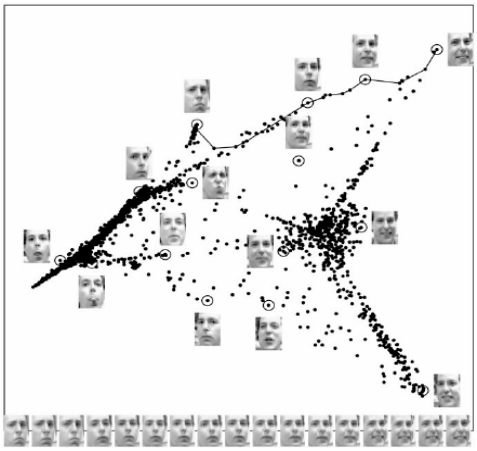


Figure 4. 11 LLE representation of facial images in 2D expression-pose space (Taken from [4.12])

4.4 Laplacian Eigenmap

Laplacian Eigenmap (LE) is another approach based on the idea of locality preserving for dimensionality reduction and feature extraction. Similar but different from LLE, it doesn't seek to preserve the exact local geometry of data. Instead, the optimizing criterion is set to maintain the 'closeness' of the samples to their neighbors.

The core algorithm is as straightforward as LLE, involving a few local computations and one sparse eigenvalue problem. The solution reflects the intrinsic geometric structure of the manifold since the embedding maps for the data come from approximations to a natural map that is defined on the entire dataset. Moreover, the locality preserving character of the LE algorithm makes it relatively insensitive to outliers and noise. Also, the algorithm implicitly emphasizes the natural clusters in the data. Such clustering structure can be biologically plausible in serving as the basis for the development of categories in perception.

Given k points $x_1,...,x_k$ in R^l, a weighted graph with k nodes, one for each point, is constructed. A set of edges connect the neighboring points to each other.

Algorithm Laplacian Eigenmap
Input: l training samples of N dimensions
Output: l feature vectors of d dimensions
Begin

For $i, j = 1,\ldots,l$

If x_i and x_j are mutually the closest neighbors

Connect x_i to x_j with an edge;

Endif

Endfor

For $i, j = 1,\ldots,l$

If x_i and x_j are connected

The weight W_{ij} of edge connecting x_i and x_j is 1;

Else

$W_{ij} = 0$.

Endif

Endfor

Compute eigenvalues and eigenvectors for the generalized eigenvector problem: $Ly = \lambda Dy$, where D is diagonal weight matrix, its entries are column sums of W, $D_{ii} = \sum_j W_{ji}$, $L = D - W$ is called the Laplacian matrix which is symmetric, and positive semidefinite.

Let $Y_0,...,Y_d$ be the solutions of equation, ordered according to their eigenvalues with y_0 having the smallest eigenvalue. The low dimensional representation of a given input X_i is thus $(Y_{i1},...,Y_{id})$.

End

Consider the problem of mapping the weighted connected graph G to a line so that connected points stay as close together as possible. The objective is to choose $y_i \in R$ to minimize

$$\sum_{i,j}(y_i - y_j)^2 W_{ij} \tag{4.5}$$

with appropriate constraints. Let $Y = (y_1,...,y_l)^T$ be the map from the graph to the real line. First, note that for any Y, we have

$$\frac{1}{2}\sum_{i,j}(y_i - y_j)^2 W_{ij} = Y^T LY \tag{4.6}$$

where $L = D - W$. Note that W_{ij} is symmetric and $D_{ii} = \sum_j W_{ij}$. Thus $\sum_{i,j}(y_i - y_j)^2 W_{ij}$ can be written as

$$\sum_{i,j}(y_i^2 + y_j^2 - 2y_i y_j)W_{ij} = \sum_i y_i^2 D_{ii} + \sum_j y_j^2 D_{jj} - 2\sum_{i,j} y_i y_j W_{ij} = 2Y^T LY \tag{4.7}$$

Also, a constraint $Y^T DY = 1$ is added to remove an arbitrary scaling factor in the embedding. Figure 4.12 shows the result of LE embedding of the handwritten digits. Compare with PCA we can see clearly that the global geometry of the dataset is better preserved, and the overlapping among the digit features is reduced. This can help with the classifications to reduce the prediction errors.

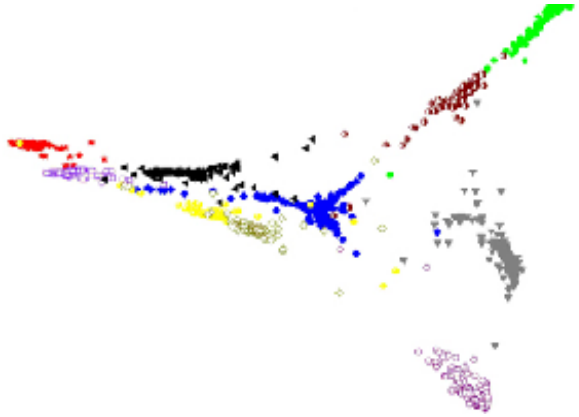


Figure 4. 12 Two dimensional LE visualization of the training data after projection onto the first 2 components (Taken from [4.14])

In summary, deriving the global structure of the dataset from the local patterns of data distribution is a brilliant idea in embedding. By doing so we can unfold the manifold to transform the data to a space where they are linearly separable.

Another important technique for embedding is called image based projection which can be used to improve significantly the computational efficiency for dimensionality reduction and feature extraction. Accordingly, two algorithms by leveraging this technique, i.e., two dimensional eigenfaces and fisherfaces, have been developed and tested on the facial image databases.

4.5 Image based projection: Two dimensional PCA

The image based projection is proposed by Yang [4.15] et al. It can be applied to reduce significantly the storage and the computation complexities. By using this technique, the recognition accuracy can also be improved on the facial image database.

4.5.1 Idea and algorithm

Let X denote an n-dimensional unitary column vector. Our idea is to project image A, an $m \times n$ random matrix, onto X by the following linear transformation:

$$Y = AX \tag{4.8}$$

Thus, we have an m-dimensional projected vector Y, which is called the projected feature vector of image A. To determine a good projection vector X, the total scatter of

the projected samples can be introduced to measure the discriminatory power of the projection vector X. The total scatter of the projected samples can be characterized by the trace of the covariance matrix of the projected feature vectors. From this point of view, the following criterion is adopted:

$$J(X) = tr(S_x) \tag{4.9}$$

where S_x denotes the covariance matrix of the projected feature vectors of the training samples and $tr(S_x)$ denotes the trace of S_x. The physical significance of maximizing the criterion in formula 4.9 is to find a projection direction X, onto which all samples are projected, so that the total scatter of the resulting projected samples is maximized. The covariance matrix S_x can be denoted by

$$\begin{aligned} S_x &= E(Y - EY)(Y - EY)^T = E\big[AX - E(AX)\big]\big[AX - E(AX)\big]^T \\ &= E\big[(A - EA)X\big]\big[(A - EA)X\big]^T \end{aligned} \tag{4.10}$$

So,

$$tr(S_x) = X^T \big[E(A - EA)^T (A - EA)\big] X \tag{4.11}$$

Let us define the following matrix

$$G_t = E\big[(A - EA)^T (A - EA)\big] \tag{4.12}$$

The matrix G_t is called the image covariance (scatter) matrix. It is easy to verify that G_t is an $n \times n$ nonnegative definite matrix from its definition. G_t can be evaluated directly using the training image samples. Suppose that there are M training image samples in total, the j-th training image is denoted by an $m \times n$ matrix $A_j (j = 1,...,M)$, and the average image of all training samples is denoted by $\overline{A}$. Then, G_t can be evaluated by

$$G_t = \frac{1}{M} \sum_{j=1}^{M} (A_j - \overline{A})^T (A_j - \overline{A}). \tag{4.13}$$

Alternatively, the criterion in (2) can be expressed by

$$J(X) = X^T G_t X \tag{4.14}$$

where X is a unitary column vector. This criterion is called the generalized total scatter criterion. The unitary vector X that maximizes the criterion is called the optimal projection axis. Intuitively, this means that the total scatter of the projected samples is maximized after the projection of an image matrix onto X. The optimal projection axis X_{opt} is the unitary vector that maximizes $J(X)$, i.e., the eigenvector of G_t corresponding to the largest eigenvalue. In general, it is not enough to have only one optimal projection axis. A set of projection axes, $X_1,...,X_d$, can be selected subjected to the orthonormal constraints and maximizing the criterion $J(X)$, that is,

$$\begin{cases} X_i^T X_j = 0, i \neq j, i, j = 1,..., d. \\ \{X_1,..., X_d\} = \arg\max J(X) \end{cases} \tag{4.15}$$

In fact, the optimal projection axes, $X_1,..., X_d$, are the orthonormal eigenvectors of G_t corresponding to the first d largest eigenvalues.

4.5.2 Feature extraction

The optimal projection vectors of 2DPCA, $X_1,..., X_d$, are used for feature extraction. For a given image sample A, let

$$Y_k = AX_k, \ k = 1, 2,..., d. \tag{4.16}$$

Then, a family of projected feature vectors, $Y_1,..., Y_d$, which are called the principal component (vectors) of the sample image A can be obtained. It should be noted that each principal component of 2DPCA is a vector, whereas the principal component of PCA is a scalar. The principal component vectors obtained are used to form an $m \times d$ matrix, $B = [Y_1,..., Y_d]$, which is called the feature matrix or feature image of the image sample A. After a transformation by 2DPCA, a feature matrix is obtained for each image. Then, a nearest neighbor classifier is used for classification. Here, the distance between two arbitrary feature matrices, $B_i = [Y_1^{(i)},..., Y_d^i]$ and $B_j = [Y_1^{(j)},..., Y_d^j]$, is defined by

$$d(B_i, B_j) = \sum_{k=1}^{d} \left\| Y_k^{(i)} - Y_k^{(j)} \right\|_2 \tag{4.17}$$

where denotes the Euclidean distance between the two principal component vectors Y_k^i and Y_k^j. Suppose that the training samples are $B_1, B_2,..., B_M$ (where M is the total number of training samples), and that each of these samples is assigned a given identity (class) ω_k. Given a test sample B, if $d(B, B_l) = \min_j d(B, B_j)$ and $B_l \in \omega_k$ then the resulting decision is $B \in \omega_k$. By virture of the image based projection, the recognition rate has been improved significantly while the computational cost is reduced, as shown in Table 4.1 and 4.2. The dimension and the number of components used are also listed.

Table 4. 1 Recognition rate: 2DPCA compared with PCA

# Training samples/class	1	2	3	4	5
PCA (Eigenfaces)	66.9(39)	84.7(79)	88.2(95)	90.8(60)	93.5(37)
2DPCA	76.7(112,2)	89.1(112,2)	91.8(112,6)	95.0(112,5)	96.0(112,3)

Table 4. 2 Computational efficiency: 2DPCA compared with PCA

# Training samples/class	1	2	3	4	5
PCA (Eigenfaces)	44.45	89.00	139.36	198.95	304.61
2DPCA	10.76	11.23	12.59	13.40	14.03

Chapter 5. Two dimensional Laplacianfaces, Unsupervised Discriminant Projection and Mutual neighborhood based discriminant projections

Based on the methods discussed in Chapter 4, we proposed three methods: two dimensional Laplacianfaces, unsupervised discriminant projection (UDP), and mutual neighborhood based projection (MNDP), by using the ideas of locality preserving and image based projection. 2D laplacianfaces is not only computatinally more efficient but also more accurate than the other methods in comparision. Also, the other two methods, UDP and MNDP based on the concept of locality can outperform the PCA and LDA on the facial and the palmprint databases.

5.1 Two dimensional Laplacianfaces

Recently, the Laplacianfaces method is developed and tested for face recognition [5.1]. It is a generalization of the locally linear embedding (LLE) [5.2] algorithm, which handle effectively the nonlinearity of the image space for dimensionality reduction. The main idea of the Laplacianfaces is to find a low dimensional representation of the data that can maximally preserve their locality, the pattern of distribution of the data around the local neighborhoods. Different from the Eigenfaces and the Fisherfaces, which search for the optimal projections by analyzing the global patterns of the data density, the Laplacianfaces method seeks the projection that can best preserve the local geometry of the training samples. The features learned maintain the locality of the training data, making it robust to the outlier samples for training and suitable for classification with the kNN method. The Laplacianfaces can outperform significantly the popular Eigenfaces and Fisherfaces methods on the Yale, the MSRA and the PIE face databases [5.1]. Yet, all of these methods are computationally inefficient due to the size of the matrix to be handled in the associated eigen equations.

To address the deficiency, Liu et al. [5.3] and Yang et al. [5.4] developed the two dimensional Eigenfaces, whereas Yang et al. [5.5], [5.6], Xiong et al. [5.7], and Jing et al. [5.8] developed the two dimensional Fisherfaces methods. Both of these methods dramatically reduced the complexity of the algorithms from $O(m^2 \times n^2)$ to $O(m^2)$, or $O(n^2)$. These methods also reduce the size of the matrices in the eigen equations, allowing them to be more accurately evaluated. The two dimensional Eigenfaces and the Fisherfaces methods prove to be effective, but it is still unclear whether image based projection can also be applied to improve the performance of Laplacianfaces. We develop the two dimensional Laplacianfaces method to answer the question. The result of extensive experiments on a variety of the benchmark databases shows that this method can outperform the one-dimensional Laplacianfaces, the two dimensional Eigenfaces and Fisherfaces methods respectively. We first provide a detailed description on the proposed 2D Laplacianfaces algorithm, then show how it differs from the standard Laplacianfaces method.

5.1.1 Idea and algorithm

Let X denote an n- dimensional unitary column vector. A represents an image of m rows and n columns. In the one dimensional Laplacianfaces method, the sample image, A, has to be transformed to form a vector of $m \times n$ dimensions prior to training. Instead, in the new algorithm, the two dimensional Laplacianfaces method, we project the image matrix directly onto the vector X

$$Y = AX . \tag{5.1}$$

The obtained m- dimensional vector Y is called the projection feature vector, which is the horizontal projection of the image A . Given a set of training images $T = \{A_1, \ldots, A_i, \ldots, A_j, \ldots, A_N\}$ the objective function of the two dimensional Laplacianfaces method is defined as

$$\underset{X}{Min} \sum_{ij} \left\| Y_i - Y_j \right\|^2 S_{ij} \tag{5.2}$$

where Y_i is the projection feature vector corresponding to the image A_i, $\|\cdot\|$ is the L_2 norm and S_{ij} is the similarity between the image A_i and A_j in the observation space and is defined as

$$S_{ij} = \begin{cases} \exp(-\left\| A_i - A_j \right\|^2 / t), & \text{if } A_i \text{ is among the k nearest neighbors of } A_j, \\ & \text{or } A_j \text{ is among the k nearest neighbors of } A_i, \\ 0, & \text{otherwise}, \end{cases} \tag{5.3}$$

where k is the size of the local neighborhood, and t is the window width determining the rate of decay of the similarity function. As shown in Eq. 5.2, the objective function imposes a heavy penalty if two arbitrary neighboring samples A_i and A_j in the original space are mapped far apart. Minimizing this function ensures that if A_i and A_j are near each other, their projection feature vectors Y_i and Y_j are close to each other as well. Therefore, the locality of the sample space can be maximally preserved when the original data are transformed to the feature space through projections. By taking several algebraic steps, the two dimensional Laplacianfaces method is formulated to minimize the following objective function

$$
\begin{aligned}
&\underset{X}{Min}\sum_{ij}\left\|Y_i - Y_j\right\|^2 S_{ij} \\
&= \sum_{ij}\left\|A_i X - A_j X\right\|^2 S_{ij} \\
&= \sum_{ij}[X^T (A_i - A_j)^T (A_i - A_j) X] S_{ij} \\
&= X^T [\sum_i A_i^T A_i \sum_j S_{ij} - \sum_{ij} A_i^T S_{ij} A_j] X \\
&= X^T A^T (D - S) A X \\
&= X^T A^T L A X
\end{aligned}
\tag{5.4}
$$

where $A^T = [A_1^T, \ldots, A_N^T]$ and $A = [A_1, \ldots, A_N]^T$ takes the mathematical operations as the $1 \times N$ and the $N \times 1$ block matrix, whose row and column consists of the image matrix A_i^T and A_i, $i = 1, ..., N$, respectively. D is the $N \times N$ block diagonal matrix, whose diagonal element is d_{ii}, $d_{ii} = \sum_j S_{ij}$, which is the sum of the similarity values of all the sample images to the *i*-th image in the original space. S is the similarity matrix, and L is called the Laplacian matrix. Both of these two matrices are of $N \times N$ dimensions. The entry of the matrix D indicates how important each point is. A constraint is imposed as follows

$$X^T A^T D A X = 1. \tag{5.5}$$

Hence, the two dimensional Laplacianfaces method is formulated as

$$
\begin{aligned}
&\underset{X}{Min}\ X^T A^T L A X \\
&\text{s.t. } X^T A^T D A X = 1.
\end{aligned}
\tag{5.6}
$$

In Eq. 5.5, the matrix D provides a natural measure on the importance of the training samples. In the original data space, the outlier samples have fewer close neighbors than those in the regions of high density of distribution. Some distortion of the local geometry near around these outliers after transformation is unlikely to have the significant impact on the result of classification. Hence, they are less important than those samples that have more close neighbors in determining the optimal directions of projection. In Eq. 5.6, by using the constraint, we are able to not only remove the arbitrary scaling factor of the projection vectors, but also take into consideration the importance of each sample for optimization [5.1].

By applying the Lagrange multiplier method, we are able to reduce Eq. 5.6 to a generalized eigen problem, as shown in Eq. 5.7

$$A^T LAX = \lambda A^T DAX , \tag{5.7}$$

where the matrices $A^T LA$ and $A^T DA$ are both of $N \times N$ dimensions, and L and D is symmetric and positive semidefinite. We can work out the optimal projection vector X by solving this equation. The eigenvectors associated with the first d smallest eigenvalues will be utilized for feature extraction.

5.1.2 Feature extraction

Let us denote the optimal projection vectors as $X_1,...,X_d$. For a given input image A, let $Y_i = AX_i$, $i = 1, \cdots, d$. A set of the projection feature vectors, Y_1, …, Y_d can then be obtained. Note that the features extracted in the two dimensional Laplacianfaces method are vectors, while in the original algorithm they are scalars. The projection vectors are used to form an $m \times d$ matrix $B = [Y_1, \ldots, Y_d]$ called the feature matrix of the sample image A.

5.1.3 Classification

After obtaining the feature matrix of all the training images, the one nearest neighbor classifier is then used for classification. The distance between any two feature matrices $B_i = [Y_{i1}, \ldots, Y_{id}]$ and $B_j = [Y_{j1}, \ldots, Y_{jd}]$ is defined by

$$d(B_i, B_j) = \sum_{p=1}^{d} \left\| Y_{ip} - Y_{jp} \right\| . \tag{5.8}$$

Suppose that the feature matrices are $B_1, \ldots, B_N$ and each of these samples is assigned a class label C. Given an input testing image B, if $d(B, B_l) = \min d(B, B_j)$ and B_l belongs to class C, then B is classified as belonging to C.

5.1.4 Experimental results

In this section, we experimentally evaluate the proposed two dimensional Laplacianfaces method on two well known face databases, FERET and AR. The FERET database is employed to test the performance of the face recognition algorithms when various numbers of samples are selected for training, while the AR database is used to assess its performance when the face images are taken with the variations of illuminations, facial expressions, and time sessions. The experiments are performed on a Pentium 4 2.6GHz PC with 512MB RAM memory under Matlab 7.1 platform.

Results on FERET database

The FERET face image database is a result of the FERET program that is sponsored by the US Department of Defense, through the Defense Advanced Research Products Agency (DARPA) [5.9]. In our evaluation, we choose a subset of the database that contains 1400 images collected from 200 individuals for examination. Specifically, seven facial images are captured for each subject with varying facial expressions, poses, and illumination conditions. In the preprocessing stage, the images are histogram equalized, manually cropped, and resized from the size of 80×80 to 40×40, to further reduce the computation and the memory costs of experiments. We perform six tests with various numbers of samples for training. Hence, in the k-th test, we select the first k images of each individual for training, and use the others for testing purpose. For Fisherfaces we add a regulatory term to avoid the singularity, so it is in fact the regularized Fisherfaces in our implemenations. The top recognition rates achieved in the six tests and the numbers of the projection vectors used for classification are presented in Table 5.1.

Table 5. 1 Top recognition rate (%) and number of components used

Method	Number of training samples of each class					
	1	2	3	4	5	6
Eigenfaces	69.5	73.6	81.8	87.7	90.8	90.8
	(64)	(77)	(78)	(55)	(48)	(72)
Regularized Fisherfaces	_	75.3	83.6	89.3	92.2	92.7
		(40)	(44)	(39)	(55)	(70)
Laplacianfaces	72.3	76.1	84.9	89.5	92.9	93.2
	(66)	(45)	(50)	(42)	(60)	(60)
2D Eigenfaces	71.8	75.7	83.7	88.2	90.8	91.2
	(2)	(2)	(3)	(4)	(3)	(3)
2D Fisherfaces	_	77.5	84.5	90.6	92.4	92.6
		(3)	(2)	(4)	(3)	(2)
2D Laplacianfaces	73.2	78.1	85.2	91.1	93.1	93.4
	(2)	(3)	(3)	(2)	(2)	(2)

It can be observed that when we choose only one sample from each class for training, the recognition rates of all the six methods are about 70% on average. Of all the methods, the proposed 2D Laplacianfaces is consistently better than the rest. Also, we note that the Fisherfaces (1D and 2D) fail to construct the within-class scatter matrix for feature extraction, as there is only one sample in each class available for training. When we increase the number of training samples from 1 to 6, the recognition rate gets improved. When we choose six samples for training and leave one sample for testing, the recognition rate reaches to its maximum of over 90% averagely. In all the six tests, the proposed 2D Laplacianfaces outperforms the 2D Eigenfaces and the 2D Fisherfaces, significantly and consistently. On the other hand, we also note that all the 2D methods show better performance than the 1D methods in terms of accuracy, which is consistent with the results obtained in [5.4-5.8]. In Figure 5.1, we show the average recognition rates of the first 40 projection vectors used for classification. For each dimension, the curve depicts the mean average of the recognition rates achieved using the various numbers of samples for training.

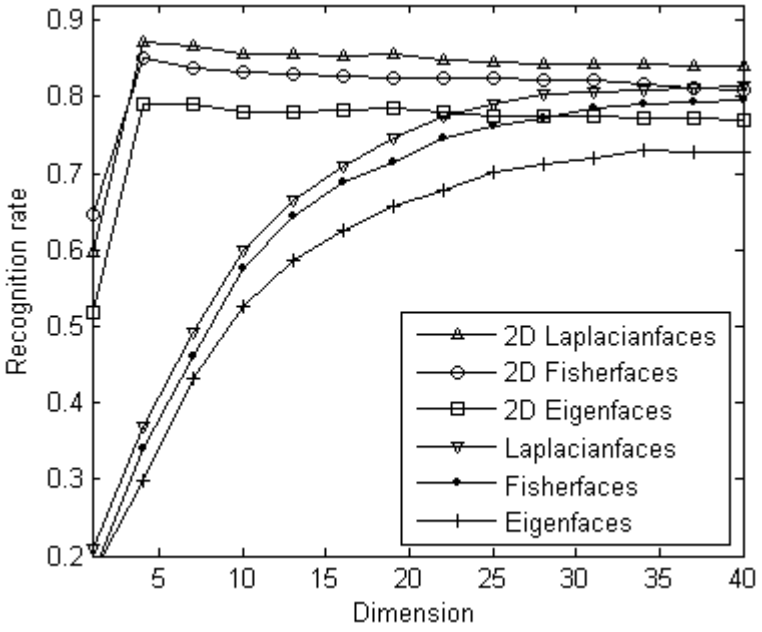


Figure 5. 1 Average recognition rate with varying dimension of projection vectors

In Figure5.1, the 2D Laplacianfaces is consistently more accurate than the 2D Fisherfaces and the 2D Eigenfaces methods. The 1D Laplacianfaces also outperforms the 1D Fisherfaces and the 1D Laplacianfaces. Here, we may note that for the 2D methods, an optimal number of projection vectors have to be carefully chosen in order to achieve the best result of classification. After testing on the first several eigenvectors, we can hardly improve the recognition rate by simply recruiting more projection vectors for classification. The reason why there is such performance loss is that for 2D methods, the selected leading eigenvectors (with the largest eigen values for 2D Eigenfaces and 2D Fisherfaces, and the smallest eigen values for 2D Laplacianfaces) are quite effective in explaining most of the discriminative information of the training data, yet, the remaining suboptimal eigenvectors are far less informative and incapable of providing further useful information for classification. The employment of these vectors can only bring up more noises that reduce the signal-to-noise ratio, which leads to the slight decreases of the recognition rates. In Table 5.2, we compare the computational and the memory space complexities of the six methods. Here, m and n is the number of the rows and the columns of the image matrix. L, M, and N is the number of the projection vectors, the testing and the training samples, respectively.

Table 5. 2 Time and memory complexities

Method	Complexity		
	Time (training)	Time (testing)	Memory
Eigenfaces/Fisherfaces	$O(m^2n^2L)$	$O(MNL)$	$O(m^2n^2)$
Laplacianfaces	$O(m^2n^2L + mnN^2)$	$O(MNL)$	$O(m^2n^2)$
2D Eigenfaces/2D Fisherfaces	$O(n^2L)$	$O(mMNL)$	$O(n^2)$
2D Laplacianfaces	$O(n^2L + mnN^2)$	$O(mMNL)$	$O(n^2)$

In Table 5.2, for the Eigenfaces and the Fisherfaces (1D and 2D), since we need to perform $O(MN)$ tests when using the nearest neighbor rule for classification and for each test it has the time complexity of $O(L)$ and $O(mL)$, the testing time is $O(MNL)$ and $O(mMNL)$ for the 1D and the 2D method, respectively. The memory cost is determined by the size of the matrices of the associated eigen equations, which is $O(m^2n^2)$ and $O(n^2)$ for the two types of methods. The training time complexity depends on both the size of the matrices in the eigen equations and the number of the projection vectors that are required to be computed. For Eigenfaces and Fisherfaces (1D and 2D), this is $O(m^2n^2L)$ and $O(n^2L)$, respectively. For the Laplacianfaces method, an extra time cost to construct the similarity matrix, i.e., $O(mnN^2)$, will be taken into account. Specifically, for the 1D and the 2D Laplacianfaces, we present and compare in Table 5.3 the CPU time for training and testing, and the size of the matrices of the eigen equations.

Table 5. 3 Time and memory space used for training and testing

Method	Average time (sec.) and memory cost				
	Time (training)	Time (testing)	Time (testing KDT)	Time (Total)	Size of matrix
Laplacianfaces	977.22	4.86	0.14	977.36	1600×1600
2D Laplacianfaces	1.59	7.72	0.18	1.77	40×40

In Table 5.3, while the one dimensional Laplacianfaces method takes averagely 977.22 seconds for training, our proposed two dimensional Laplacianfaces uses only 1.59 seconds. Moreover, the size of the matrix is reduced from 1600×1600 to 40×40, which significantly improves the memory efficiency of the algorithm. We may also note that the testing time of the 2D Laplacianfaces is 7.72, slightly higher than that of the 1D Laplacianfaces, 4.86 seconds. To improve the testing efficiency of the algorithm, we can exploit the *k*-dimensional tree method (KDT) [5.12] to accelerate the searching process of the nearest neighbor classification. The KDT method is a popular decision tree algorithm. It can recursively partition the sample space into a number of the smaller subsets for efficient pattern indexing and query. Given some input pattern for matching, it transverses the tree structure while making simple test at each branching node to discard a large portion of the data, so as to speed up the searching process. The query time complexity of the KDT algorithm is at worst $O(MN)$ and at best $O(M \log N)$, which is much lower than that of the simple kNN retrieval. In our experiment, by taking advantage of the KDT algorithm [5.10], the testing time of the two methods is reduced significantly from 4.86 to 0.14, and 7.72 to 0.18 (second), respectively, which makes 2D Laplacianfaces a practical choice for real world applications.

Results on AR database

The AR face database [5.11] consists of over 4,000 face images of 126 individuals taken in two time sessions under the variations of illuminations, facial expressions, and occlusion conditions. Each person has 26 images. In our experiment we consider using a subset of 14 images of each person for training and testing. Figure 5.2 shows the selected sample images of one subject.

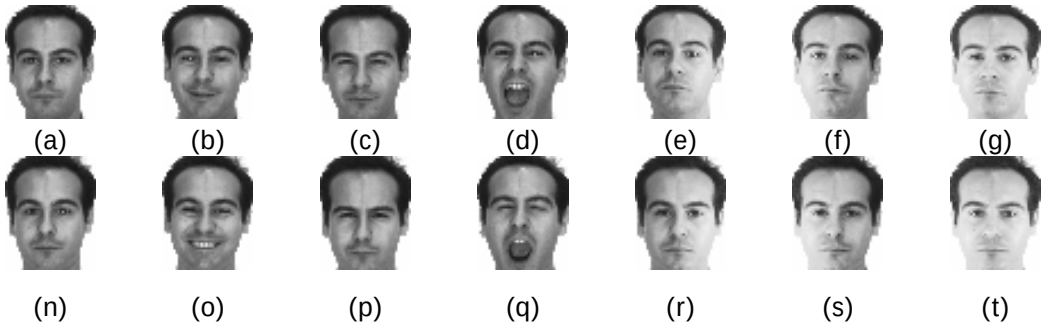


Figure 5. 2 Sample images for one subject of the AR database

In Figure 5.2, the images (a)-(g) and (n)-(t) are drawn from the first and the second time sessions respectively. For each session the first four images (a)-(d) and (n)-(q) involve the variation of facial expressions (neutral, smile, anger, scream) while the images (e)-(g) and (r)-(t) are taken under different lighting conditions (left light on, right light on, all sides light on). The images are manually cropped and scaled down to 50×40 pixels to reduce the computation and the memory costs of the experiment in the preprocessing stage. We design and perform three experiments to examine the performance of 2D Eigefaces, 2D Fisherfaces and 2D Laplacianfaces, under the variations of facial expressions, time sessions, and illumination conditions. The indices of the images of each person used in the three tests are listed in Table 5.4.

Table 5. 4 Indices of training and testing images

Data set	Experiment conditions		
	Illumination	Expression	Time
Training set	{e, s}	{a, n}	{a, b, c, d, e, f, g}
Testing set	{f, g, r, t}	{b, c, d, o, p, q}	{n, o, p, q, r, s, t}

Table 5.5 shows the top recognition rate, the number of the dimensions of feature vectors used for classification, and the testing time of the three algorithms.

Table 5. 5 Performance of three algorithms using image based projection technique

Experiment		Top recognition rate (%)	Dimension	Classification time (sec.)
Expression	2D Eigenfaces	95.4	10	5.547
	2D Fisherfaces	95.6	10	5.281
	2D Laplacianfaces	**97.8**	**4**	**4.765**
Time	2D Eigenfaces	65.2	22	42.42
	2D Fisherfaces	68.6	14	28.75
	2D Laplacianfaces	**71.5**	**4**	**17.66**
Illumination	2D Eigenfaces	80.2	27	12.375
	2D Fisherfaces	91.4	9	3.765
	2D Laplacianfaces	**93.7**	**3**	**1.975**

It can be seen that the proposed 2D Laplacianfaces method outperforms the 2D Fisherfaces and the 2D Eigenfaces methods in all the three tests. It improves the

recognition rate by 2.4%, 6.3%, 3.5% over the 2D Eigenfaces, and 2.2%, 2.9%, 2.3% over the 2D Fisherfaces, respectively. It requires fewer dimensions of projection vectors and time to achieve the top recognition rate as shown in column 5 of Table 5.5. Further, in Figure 5.3 to Figure 5.5 we also show the relationship between the accuracy rate of the three algorithms and the dimension of the feature vectors used for recognition.

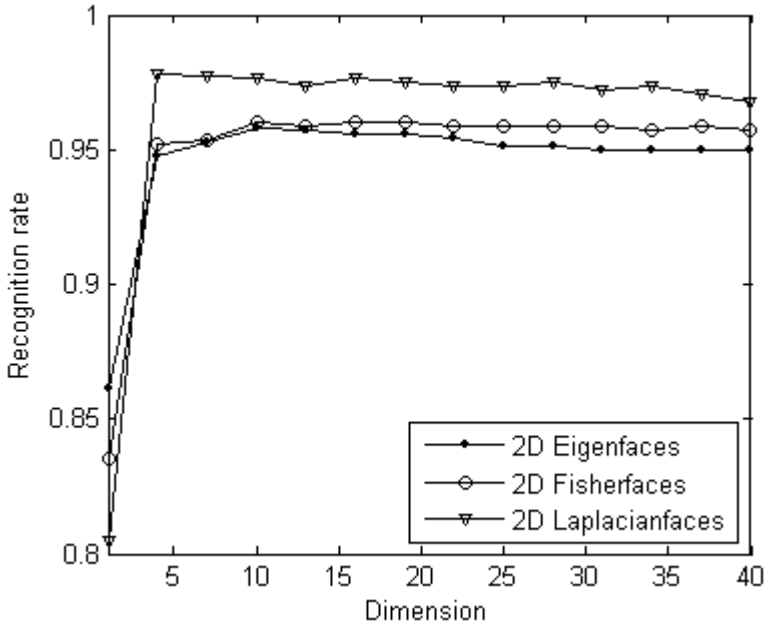


Figure 5. 3 Recognition rate over dimensions of feature vectors (Expressions)

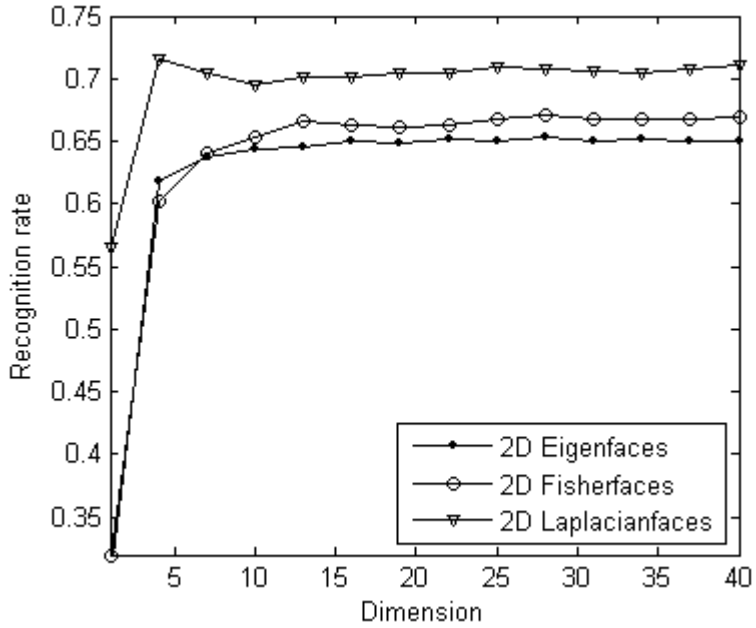


Figure 5. 4 Recognition rate over dimensions of feature vectors (Time)

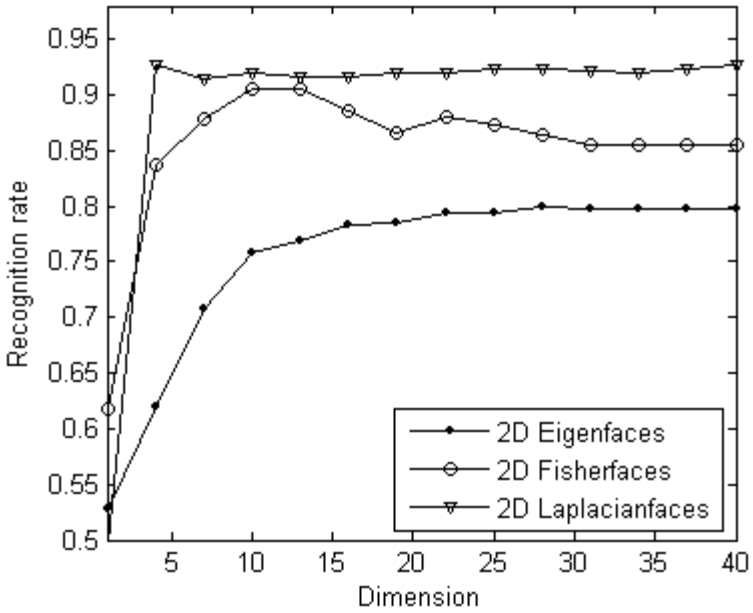


Figure 5. 5 Recognition rate over dimensions of feature vectors (Illumination)

In these figures, we can observe that the 2D Laplacianfaces method can explain most of the effective discriminative information with only a small number of projection vectors, as opposed to the other two methods, where more features have to be provided to achieve the top recognition rate. The 2D Laplacianfaces is also quite stable and consistent in outperforming the 2D Eigenfaces and the 2D Fisherfaces methods with various number of feature vectors, as indicated in the figures.

5.1.5 Conclusion

We developed the two dimensional Laplacianfaces method and applied it to the face recognition problem. The proposed method has the following three properties: First, it can maximally preserve the locality of the geometric structure of the sample space to extract the most salient features for classification. The learned local patterns of the training data are suitable for the neighborhood based kNN queries in the projected low dimensional feature space. Experimental results on the two well known face image databases, FERET and AR, indicate that the proposed 2D Laplacianfaces is more accurate than the 2D Eigenfaces and the 2D Fisherfaces that rely on the global information of the data space for analysis. Second, by taking advantage of the image based projection technique, 2D Laplacianfaces is computationally more efficient than the 1D Laplacianfaces for training. Both the training time and the memory efficiency of the algorithm are improved significantly. The recognition accuracy of the 2D Laplacianfaces is also better than that of the 1D Laplacianfaces as the size of the matrix is small, enabling the full optimization of the objective function. Third, the utilization of the k-d tree algorithm is quite effective in speeding up the kNN query process. By adopting the k-d tree method, the 2D Laplacianfaces is improved to be not only more efficient for training, but also as competitively fast as other methods for query and classification.

Finally, it should be pointed out that the application of our proposed 2D Laplacianfaces method is not limited to the face recognition problem. It can also be potentially utilized to address many other types of problems in pattern recognition, such as palm and finger print recognition, gesture recognition, audio and video clustering, gene microarray analysis, financial time-series predictions, and web document classification, etc., where the analysis of the high dimensional data is required.

5.2 Unsupervised Discriminant Projection (UDP)

Dimensionality reduction is the construction of a meaningful low-dimensional representation of high-dimensional data. Since there are large volumes of high-dimensional data in numerous real-world applications, dimensionality reduction is a fundamental problem in many scientific fields. From the perspective of pattern recognition, dimensionality reduction is an effective means of avoiding the "curse of dimensionality" [5.13] and improving the computational efficiency of pattern matching.

Researchers have developed many useful dimensionality reduction techniques. These techniques can be broadly categorized into two classes: linear and nonlinear. Linear dimensionality reduction seeks to find a meaningful low-dimensional subspace in a high-dimensional input space. This subspace can provide a compact representation of higher-dimensional data when the structure of data embedded in the input space is linear. PCA and LDA are two well-known linear subspace learning methods which have been extensively used in pattern recognition and computer vision areas and have become the most popular techniques for face recognition and other biometric applications [5.14-26, 5.51].

Linear models, however, may fail to discover essential data structures that are nonlinear. A number of nonlinear dimensionality reduction techniques have been developed to address this problem, with two in particular attracting wide attention: kernel-based techniques and manifold learning-based techniques. The basic idea of kernel-based techniques is to implicitly map observed patterns into potentially much higher dimensional feature vectors by using a nonlinear mapping determined by a kernel. This makes it possible for the nonlinear structure of data in observation space to become linear in feature space, allowing the use of linear techniques to deal with the data. The representative techniques are kernel principal component analysis (KPCA) [5.27] and kernel Fisher discriminant (KFD) [5.28, 5.29]. Both have proven to be effective in many real-world applications [5.30, 5.31, 5.32].

In contrast with kernel-based techniques, the motivation of manifold learning is straightforward, as it seeks to directly find the intrinsic low-dimensional nonlinear data structures hidden in observation space. The past few years have seen proposed many manifold-based learning algorithms for discovering intrinsic low-dimensional embedding of data. Among the most well-known are isometric feature mapping (ISOMAP) [5.34], local linear embedding (LLE) [5.35], and Laplacian Eigenmap [5.36]. Some experiments have shown that these methods can find perceptually meaningful embeddings for face or digit images. They also yielded impressive results on other artificial and real-world data sets. Recently, Yan et al. [5.35] proposed a general dimensionality reduction framework

called *graph embedding*. LLE, ISOMAP, and Laplacian Eigenmap can all be reformulated as a unified model in this framework.

One problem with current manifold learning techniques is that they might be unsuitable for pattern recognition tasks. There are two reasons for this. First, as it is currently conceived, manifold learning is limited in that it is modeled based on a characterization of "locality", a modeling that has no direct connection to classification. This is unproblematic for existing manifold learning algorithms as they seek to model a simple manifold, for example, to recover an embedding of one person's face images [5.33, 5.34, 5.35]. However, if face images do exist on a manifold, different persons' face images could lie on different manifolds. To recognize faces, it would be necessary to distinguish between images from different manifolds. For achieving an optimal recognition result, the recovered embeddings corresponding to different face manifolds should be as separate as possible in the final embedding space. This poses a problem that we might call "classification-oriented multi-manifolds learning". This problem cannot be addressed by current manifold learning algorithms, including some supervised versions [5.37-39], because they are all based on the characterization of "locality". The local quantity suffices for modeling a single manifold, but does not suffice for modeling multi-manifolds for classification purposes. To make different embeddings corresponding to different classes mutually separate, however, it is crucial to have the "non-local" quantity, which embodies the distance between embeddings. In short, it is necessary to characterize the "non-locality" when modeling multi-manifolds.

The second reason why most manifold learning algorithms, for example, ISOMAP, LLE and Laplacian Eigenmap, are unsuitable for pattern recognition tasks is that they can yield an embedding directly based on the training data set but, because of the implicitness of the nonlinear map, when applied to a new sample, they cannot find the sample's image in the embedding space. This limits the applications of these algorithms to pattern recognition problems. Although some research has shown that it is possible to construct an explicit map from input space to embedding space [5.40-42], the effectiveness of these kinds of maps on real-world classification problems still needs to be demonstrated.

Recently, He et al. [5.43, 5.44] proposed Locality Preserving Projections (LPP), which is a linear subspace learning method derived from Laplacian Eigenmap. In contrast to most manifold learning algorithms, LPP possesses the remarkable advantage that it can generate an explicit map. This map is linear and easily-computable, like that of PCA or LDA. It is also effective, yielding encouraging results on face recognition tasks. Yet as it is modeled on the basis of "locality", LPP, like most manifold learning algorithms, has the weakness of having no direct connection to classification. The objective function of LPP is to minimize the local quantity, i.e., the local scatter of the projected data. In some cases, this criterion cannot be guaranteed to yield a good projection for classification purposes. Assume, for example, that there exist two clusters of two-dimensional samples scattering uniformly in two ellipses C_1 and C_2 , as shown in Figure 5.6. If the locality radius δ is set as the length of the semi-major axis of the larger ellipse, the direction w_1 is a nice projection according to the criterion of LPP, since after all samples are projected onto w_1, the local scatter is minimal. But, it is obvious that w_1 is not good in terms of classification; the projected samples overlap in this direction. This example also shows

that the non-local quantity, i.e. the inter-cluster scatter, may provide crucial information for discrimination. In this chapter, we will address this issue and explore more effective projections for classification purposes.

Motivated by the idea of classification-oriented multi-manifolds learning, we consider two quantities, local and non-local, at the same time in the modeling process. It should be pointed out that we don't attempt to build a framework for multi-manifolds based learning in this chapter (although it is very interesting). We are more interested in its linear approximation, i.e., finding a simple and practical linear map for biometrics applications. To this end, we first present the techniques to characterize the local and non-local scatters of data. Then, based on this characterization, we propose a criterion, which seeks to maximize the ratio of the non-local scatter to the local scatter. This criterion, similar to the classical Fisher criterion, is a Rayleigh quotient in form. Thus, it is not hard to find its optimal solutions by solving a generalized eigen-equation. Since the proposed method does not use the class-label information of samples in the learning process, this method is called the unsupervised discriminant projection (UDP), in contrast with the supervised discriminant projection of LDA.

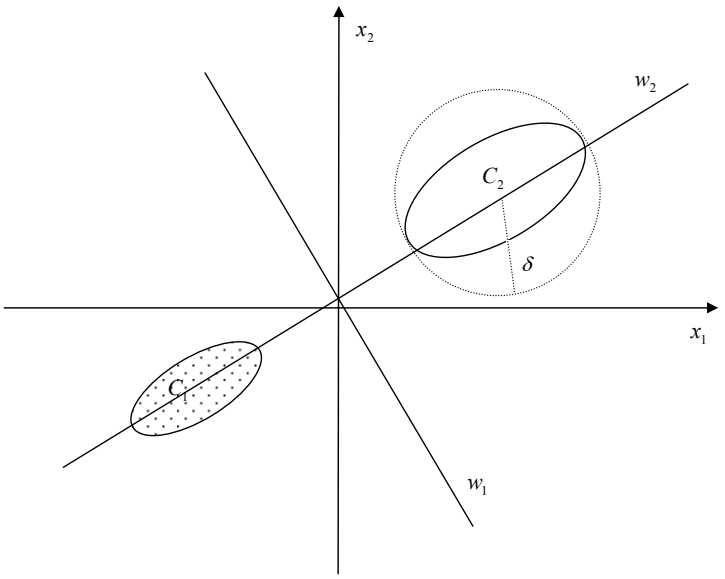


Figure 5. 6 Illustration of two clusters of samples in two-dimensional space and the projection directions

In contrast with LPP, UDP has direct relations to classification since it utilizes the information of the "non-locality". Provided that each cluster of samples in the observation space is exactly within a local neighbor, UDP can yield an optimal projection for clustering in the projected space, while LPP cannot. As shown in Figure 5.6, w_2 is a good projection direction according the criterion of UDP, which is more discriminative than w_1. In addition, UDP will be demonstrated more effective than LPP in real-world biometrics applications, based on our experiments with three face image databases and one palmprint database.

In the literature, besides LPP, there are two methods most relevant to ours. One is Marginal Fisher Analysis (MFA) presented by Yan et al. [5.45], and the other is Local Discriminant Embedding (LDE) suggested by Chen et al. [5.46]. The two methods are very similar in formulation. Both of them combine *locality* and *class label* information to represent the intra-class compactness and inter-class separability. So, MFA and LDE can be viewed as supervised variants of LPP, or as localized variants of LDA since both methods focus on the characterization of *intra-class locality* and *inter-class locality*. In contrast, the proposed UDP retains the unsupervised characteristic of LPP and seeks to combine *locality* and *globality* information for discriminator design.

The remainder of this paper is organized as follows. Section 5.2.1 outlines PCA and LDA. Section 5.2.2 develops the idea of UDP and the relevant theory and algorithm. Section 5.2.3 describes a kernel weighted version of UDP. Section 5.2.4 discusses the relations between UDP and LDA/LPP. Section 6 describes some biometrics applications and the related experiments. Section 7 offers our conclusions.

5.2.1 Outline of PCA

PCA seeks to find a projection axis such that the *global scatter* is maximized after the projection of samples. The *global scatter* can be characterized by the mean square of the Euclidean distance between any pair of the projected sample points. Specifically, given a set of M training samples (pattern vectors) $X_1, X_2, \cdots, X_M$ in R^N, we get their images $y_1, y_2, \cdots, y_M$ after the projection onto the projection axis w. The global scatter is defined by

$$J_T(w) = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(y_i - y_j)^2 \tag{5.9}$$

It follows that

$$\begin{aligned} J_T(w) &= \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(w^T X_i - w^T X_j)^2 \\ &= w^T\left[\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(X_i - X_j)(X_i - X_j)^T\right]w \end{aligned} \tag{5.10}$$

Let us denote

$$S_T = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(X_i - X_j)(X_i - X_j)^T \tag{5.11}$$

and the mean vector $m_0 = \frac{1}{M}\sum_{j=1}^{M} X_j$. Then, it can be proven that

$$S_T = \frac{1}{MM}\left[M\sum_{i=1}^{M} X_i X_i^T - \left(\sum_{i=1}^{M} X_i\right)\left(\sum_{j=1}^{M} X_j^T\right)\right] \tag{5.12}$$

Eq. 5.12 indicates that S_T is essentially the covariance matrix of data. So, the projection axis w that maximizes Eq. 5.10 can be selected as the eigenvector of S_T corresponding to the largest eigenvalue. Similarly, we can obtain a set of projection axes of PCA by selecting the d eigenvectors of S_T corresponding to d largest eigenvalues.

5.2.2 Outline of LDA

LDA seeks to find a projection axis such that the Fisher criterion (i.e., the ratio of the *between-class scatter* to the *within-class scatter*) is maximized after the projection of samples. The between-class and within-class scatter matrices S_B and S_W are defined by

$$S_B = \frac{1}{M}\sum_{i=1}^{c} l_i (m_i - m_0)(m_i - m_0)^T \tag{5.13}$$

$$S_W = \sum_{i=1}^{c} \frac{l_i}{M} S_W^{(i)} = \frac{1}{M}\sum_{i=1}^{c}\sum_{j=1}^{l_i} (X_{ij} - m_i)(X_{ij} - m_i)^T \tag{5.14}$$

where X_{ij} denotes the j-th training sample in class i; c is the number of classes; l_i is the number of training samples in class i; m_i is the mean of the training samples in class i; $S_W^{(i)}$ denotes the covariance matrix of samples in class i. It is easy to show that S_B and S_W are both non-negative definite matrix and satisfy $S_T = S_B + S_W$. The Fisher criterion is defined by

$$J_F(w) = \frac{w^T S_B w}{w^T S_W w} \tag{5.15}$$

The stationary points of $J_F(w)$ are the generalized eigenvectors $w_1, w_2, \cdots, w_d$ of $S_B w = \lambda S_W w$ corresponding to d largest eigenvalues. These stationary points form the coordinate system of LDA.

5.2.3 Basic Idea of UDP

As discussed in Introduction, the locality characterization-based model does not guarantee a good projection for classification purposes. To address this, we will introduce the concept of non-locality and give the characterizations of the non-local scatter and the local scatter. This will allow us to obtain a concise criterion for feature extraction by maximizing the ratio of non-local scatter to local scatter.

Characterize the Local Scatter

Recall that in PCA, in order to preserve the global geometric structure of data in a transformed low-dimensional space, account is taken of the global scatter of samples. Correspondingly, if we aim to discover the local structure of data, we should take account of the local scatter (or *intra-locality scatter*) of samples. The local scatter can be characterized by the mean square of the Euclidean distance between any pair of the projected sample points that are within any local δ-neighborhood ($\delta > 0$). Specifically, two samples X_i and X_j are viewed within a local δ-neighborhood provided that $\| X_i - X_j \|^2 < \delta$. Let us denote the set $U^\delta = \{(i,j) \big| \| X_i - X_j \|^2 < \delta\}$. After the projection of X_i and X_j onto a direction w, we get their images y_i and y_j. The local scatter is then defined by

$$J_L(w) = \frac{1}{2}\frac{1}{M_L}\sum_{(i,j)\in U^\delta}(y_i - y_j)^2 \propto \frac{1}{2}\frac{1}{MM}\sum_{(i,j)\in U^\delta}(y_i - y_j)^2 \tag{5.16}$$

where M_L is the number of sample pairs satisfying $\| X_i - X_j \|^2 < \delta$. Let us define the adjacency matrix H, whose elements are given below:

$$H_{ij} = \begin{cases} 1, & \| X_i - X_j \|^2 < \delta \\ 0 & otherwise \end{cases} \tag{5.17}$$

It is obvious that the adjacency matrix H is a symmetric matrix. By virtue of the adjacency matrix H, Eq. 5.16 can be rewritten by

$$J_L(w) = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}(y_i - y_j)^2 \tag{5.18}$$

It follows from Eq. 5.18 that

$$\begin{aligned} J_L(w) &= \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}(w^T X_i - w^T X_j)^2 \\ &= w^T\left[\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}(X_i - X_j)(X_i - X_j)^T\right]w \\ &= w^T S_L w \end{aligned} \tag{5.19}$$

where

$$S_L = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M} H_{ij}(X_i - X_j)(X_i - X_j)^T \tag{5.20}$$

S_L is called the local scatter (covariance) matrix. Due to the symmetry of H, we have

$$
\begin{aligned}
S_L &= \frac{1}{2}\frac{1}{MM}\left(\sum_{i=1}^{M}\sum_{j=1}^{M}H_{ij}X_iX_i^T+\sum_{i=1}^{M}\sum_{j=1}^{M}H_{ij}X_jX_j^T-2\sum_{i=1}^{M}\sum_{j=1}^{M}H_{ij}X_iX_j^T\right)\\
&= \frac{1}{MM}\left(\sum_{i=1}^{M}D_{ii}X_iX_i^T-\sum_{j=1}^{M}\sum_{j=1}^{M}H_{ij}X_iX_j^T\right)=\frac{1}{MM}(XDX^T-XHX^T)\\
&= \frac{1}{M}XLX^T
\end{aligned}
\tag{5.21}
$$

where $X=(X_1,X_2,\cdots,X_M)$, and D is a diagonal matrix whose elements on diagonal are column (or row since H is a symmetric matrix) sum of H, i.e., $D_{ii}=\sum_{j=1}^{M}H_{ij}$. $L=D-H$ is called the local scatter kernel (LSK) matrix in this chapter (this matrix is called the Laplacian matrix in [5.36]).

It is obvious that L and S_L are both real symmetric matrices. From Eqs. 5.19 and 5.21, we know that $w^TS_Lw\geq 0$ for any nonzero vector w. So, the local scatter matrix S_L must be non-negative definite.

In the above discussion, we use δ-neighborhoods to characterize the "locality" and the local scatter. This way is geometrically intuitive but unpopular because in practice it is hard to choose a proper neighborhood radius δ. To avoid this difficulty, the method of K-nearest neighbors is always used instead in real-world applications. The K-nearest neighbors method can determine the following adjacency matrix H, with elements given by:

$$
H_{ij}=\begin{cases}1, & \textit{if } X_j \textit{ is among K nearest nieghbors of } X_i\\ & \textit{and } X_i \textit{ is among K nearest nieghbors of } X_j\\ 0 & \textit{otherwise}\end{cases}
\tag{5.22}
$$

The local scatter can be characterized similarly by a K-nearest neighbor adjacency matrix if Eq. 5.17 is replaced by Eq. 5.22.

Characterize the Non-local Scatter

The non-local scatter (i.e., the inter-locality scatter) can be characterized by the mean square of the Euclidean distance between any pair of the projected sample points that are outside any local δ-neighborhood ($\delta>0$). The non-local scatter is defined by

$$
J_N(w)=\frac{1}{2}\frac{1}{M_N}\sum_{(i,j)\notin U^\delta}(y_i-y_j)^2 \propto \frac{1}{2}\frac{1}{MM}\sum_{(i,j)\notin U^\delta}(y_i-y_j)^2
\tag{5.23}
$$

where M_N is the number of sample pairs satisfying $\|X_i-X_j\|^2\geq\delta$. By virtue of the adjacency matrix H in Eq. 5.17 or 5.22**,** the non-local scatter can be rewritten by

$$J_N(w) = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(1-H_{ij})(y_i - y_j)^2 \tag{5.24}$$

It follows from Eq. 5.24 that

$$J_N(w) = w^T\left[\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(1-H_{ij})(X_i - X_j)(X_i - X_j)^T\right]w = w^T S_N w \tag{5.25}$$

where

$$S_N = \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(1-H_{ij})(X_i - X_j)(X_i - X_j)^T \tag{5.26}$$

S_N is called the non-local scatter (covariance) matrix. It is easy to show S_N is also a non-negative definite matrix. And, it follows that

$$\begin{aligned} S_N &= \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(1-H_{ij})(X_i - X_j)(X_i - X_j)^T \\ &= \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(X_i - X_j)(X_i - X_j)^T - \frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}H_{ij}(X_i - X_j)(X_i - X_j)^T \\ &= S_T - S_L \end{aligned} \tag{5.27}$$

That is, $S_T = S_L + S_N$. Thus, we have $J_T(w) = J_L(w) + J_N(w)$.

Determine a Criterion: Maximizing the Ratio of Non-local Scatter to Local Scatter

The technique of Locality Preserving Projection (LPP) [5.43] seeks to find a linear subspace which can preserve the local structure of data. The objective of LPP is actually to minimize the local scatter $J_L(w)$. Obviously, the projection direction determined by LPP can ensure that, if samples X_i and X_j are close, their projections y_i and y_j are close as well. But LPP cannot guarantee that if samples X_i and X_j are not close their projections y_i and y_j are not either. This means that it may happen that two mutually distant samples belonging to different classes may result in close images after the projection of LPP. Therefore, LPP does not necessarily yield a good projection suitable for classification.

For the purpose of classification, we try is to find a projection which will draw the close samples closer together while simultaneously making the mutually distant samples even more distant from each other. From this point of view, a desirable projection should be the one that at the same time minimizes the local scatter $J_L(w)$ and maximizes the non-

local scatter $J_N(w)$. As it happens, we can obtain just such a projection by maximizing the following criterion:

$$J(w) = \frac{J_N(w)}{J_L(w)} = \frac{w^T S_N w}{w^T S_L w} \tag{5.28}$$

Since $J_T(w) = J_L(w) + J_N(w)$ and $S_T = S_L + S_N$, the above criterion is equivalent to

$$J_e(w) = \frac{J_T(w)}{J_L(w)} = \frac{w^T S_T w}{w^T S_L w} \tag{5.29}$$

The criterion in Eq. 5.28 indicates that we can find the projection by at the same time globally maximizing (maximizing the global scatter) and locally minimizing (minimizing the local scatter).

The criterion in Eq. 5.27 or 5.28 is formally similar to the Fisher criterion in Eq. 5.15 since they are both Rayleigh quotients. Differently, the matrices S_L and S_N in Eq. 5.27 can be constructed without knowing the class-label of samples while S_B and S_W in Eq. 5.15 cannot be so constructed. This means Fisher discriminant projection is supervised while the projection determined by $J(w)$ can be obtained in an unsupervised manner. In this chapter, then, this projection is called an Unsupervised Discriminant Projection (UDP). The criterion in Eq. 5.27 can be maximized directly by calculating the generalized eigenvectors of the following generalized eigen-equation:

$$S_N w = \lambda S_L w \tag{5.30}$$

The projection axes of UDP can be selected as the generalized eigenvectors $w_1, w_2, \cdots, w_d$ of $S_N w = \lambda S_L w$ corresponding to *d* largest positive eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$.

5.2.4 UDP Algorithm

In summary of the preceding description, the following provides the UDP algorithm:

Step 1. Construct the adjacency matrix: For the given training data set $\{X_i \mid i = 1, \cdots, M\}$, find K nearest neighbors of each data point and construct the adjacency matrix $H = (H_{ij})_{M \times M}$ using Eq. 5.22.

Step 2. Construct the local scatter kernel (LSK) matrix: Form an $M \times M$ diagonal matrix D, whose elements on the diagonal are given by $D_{ii} = \sum_{j=1}^{M} H_{ij}$, $i = 1, \cdots, M$. Then, the LSK matrix is $L = D - H$.

Step 3. Perform PCA transform of data: Calculate S_T's *m* largest positive eigenvalues $\mu_1, \cdots, \mu_m$ and the associated *m* orthonormal eigenvectors $\beta_1, \cdots, \beta_m$ using the

technique presented in [5.14, 5.15]. Let $\widetilde{S}_T = \mathrm{diag}(\mu_1, \cdots, \mu_m)$ and $P = (\beta_1, \cdots, \beta_m)$. Then we get $\widetilde{X} = P^T X$, where $X = (X_1, X_2, \cdots, X_M)$.

Step 4. Construct the two matrices $\widetilde{S}_L = \widetilde{X} L \widetilde{X}^T / M^2$ and $\widetilde{S}_N = \widetilde{S}_T - \widetilde{S}_N$. Calculate the generalized eigenvectors $v_1, v_2, \cdots, v_d$ of $\widetilde{S}_N v = \lambda \widetilde{S}_L v$ corresponding to d largest positive eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. Then, the d projection axes of UDP are $w_j = P v_j$, $j = 1, \cdots, d$.

After obtaining the projection axes, we can form the following linear transform for a given sample X_i:

$$Y_i = W^T X_i, \text{ where } W = (w_1, w_2, \cdots, w_d) \tag{5.31}$$

The feature vector Y_i is used to represent the sample X_i for recognition purposes. Concerning the UDP Algorithm, a remark should be made on the choice of *m* in Step 3. Liu and Wechsler [5.22] suggested a criterion for choosing the number of principal components in the PCA phase of their enhanced Fisher discriminant models. That is, a proper balance should be preserved between the *data energy* and the *eigenvalue magnitude* of the within-class scatter matrix [5.22]. This criterion can be borrowed here for the choice of *m*. First, to make $\widetilde{S}_L$ non-singular, an *m* should be chosen that is less than the rank of LSK matrix L. Second, to avoid overfitting, the trailing eigenvalues of $\widetilde{S}_L$ should not be too small.

5.2.5. Extension: UDP with Kernel Weighting

In this section, we will build a kernel-weighted version of UDP. We know that Laplacian Eigenmap [5.36] and LPP [5.43, 5.44] use kernel coefficients to weight the edges of the adjacency graph, where a heat kernel (Gaussian kernel) is defined by

$$k(X_i, X_j) = \exp\left(- \| X_i - X_j \|^2 / t\right) \tag{5.32}$$

Obviously, for any X_i, X_j and parameter *t*, $0 < k(X_i, X_j) \leq 1$ always holds. Further, the kernel function is a strictly monotone decreasing function with respect to the distance between two variables X_i and X_j.

The purpose of the kernel weighting is to indicate the *degree* of X_i and X_j belonging to a local δ-neighborhood. If $\| X_i - X_j \|^2 < \delta$, the smaller the distance is, the larger the *degree* would be. Otherwise, the degree is zero. The kernel weighting, like other similar weightings, may be helpful in alleviating the effect of the outliers on the projection directions of the linear models and thus makes these models more robust to outliers [5.47].

Fundamentals

Let $K_{ij}=k(X_i,X_j)$. The kernel weighted global scatter can be characterized by

$$J_T(w)=\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}K_{ij}(y_i-y_j)^2=w^T\left[\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}K_{ij}(\mathrm{x}_i-\mathrm{x}_j)(\mathrm{x}_i-\mathrm{x}_j)^T\right]w \quad (5.33)$$

Here, we can view that all training samples are within a δ-neighborhood (it is possible as long as δ is large enough). K_{ij} indicates the *degree* of X_i and X_j belonging to such a neighborhood. Let us denote

$$S_T=\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}K_{ij}(X_i-X_j)(X_i-X_j)^T, \quad (5.34)$$

Similar to the derivation of Eq. 5.21, we have

$$S_T=\frac{1}{MM}(XD_TX^T-XKX^T)=\frac{1}{MM}XL_TX^T \quad (5.35)$$

where $K=(K_{ij})_{M\times M}$, and D_T is a diagonal matrix whose elements on the diagonal are column (or row) sum of K, i.e., $(D_T)_{ii}=\sum_{j=1}^{M}K_{ij}$. $L_T=D_T-K$ is called the global scatter kernel (GSK) matrix. If the matrix S_T defined in Eq. 5.35 is used as the generation matrix and its principal eigenvectors are selected as projection axes, the *kernel weighted version of PCA* can be obtained. If we redefine the adjacency matrix as

$$H_{ij}=\begin{cases}K_{ij}, & \|\mathbf{x}_i-\mathbf{x}_j\|^2<\delta\\ 0 & \text{otherwise}\end{cases} \quad (5.36)$$

or

$$H_{ij}=\begin{cases}K_{ij}, & \textit{if } \mathrm{x}_j \textit{ is among } K \textit{ nearest nieghbors of } \mathrm{x}_i \textit{ or reverse}\\ 0 & \textit{otherwise}\end{cases} \quad (5.37)$$

the kernel-weighted local scatter can still be characterized by Eq. 5.18 or 5.19, and the kernel-weighted local scatter matrix can be expressed by Eq. 5.21. The kernel-weighted non-local scatter is characterized by

$$J_N(w)=\frac{1}{2}\frac{1}{MM}\sum_{i=1}^{M}\sum_{j=1}^{M}(K_{ij}-H_{ij})(y_i-y_j)^2 \quad (5.38)$$

$$= w^T \left[\frac{1}{2} \frac{1}{MM} \sum_{i=1}^{M} \sum_{j=1}^{M} (K_{ij} - H_{ij})(X_i - X_j)(X_i - X_j)^T \right] w$$

and the corresponding non-local scatter matrix is

$$S_N = \frac{1}{2} \frac{1}{MM} \sum_{i=1}^{M} \sum_{j=1}^{M} (K_{ij} - H_{ij})(X_i - X_j)(X_i - X_j)^T = S_T - S_L \tag{5.39}$$

Algorithm of UDP with Kernel Weighting

The UDP algorithm in Section 5.2.4 can be modified to obtain its kernel weighted version. In Step 1, the adjacency matrix $H = (H_{ij})_{M \times M}$ is constructed instead using Eq. 5.37 and, in Step 3, the PCA transform is replaced by its kernel-weighted version. For computational efficiency, the eigenvectors of S_T defined in Eq. 5.35 can be calculated in the following way.

Since L_T is a real symmetric matrix, its eigenvalues are all real. Calculate all of its eigenvalues and the corresponding eigenvectors. Suppose Λ is the diagonal matrix of eigenvalues of L_T and Q is the full matrix whose columns are the corresponding eigenvectors, L_T can be decomposed by

$$L_T = Q \Lambda Q^T = Q_L Q_L^T \tag{5.40}$$

where $Q_L = Q \Lambda^{\frac{1}{2}}$. From Eq. 5.40, it follows that $S_T = \frac{1}{MM} (XQ_L)(XQ_L)^T$. Let us define $R = \frac{1}{MM} (XQ_L)^T (XQ_L)$, which is an $M \times M$ non-negative definite matrix. Calculate R's orthonormal eigenvectors $\alpha_1, \alpha_2, \cdots, \alpha_d$ which correspond to d largest nonzero eigenvlaues $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_d$. Then, from the theorem of singular value decomposition (SVD) [5.48], the orthonormal eigenvectors $\beta_1, \beta_2, \cdots, \beta_d$ of S_T corresponding to d largest nonzero eigenvlaues $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_d$ are

$$\beta_j = \frac{1}{M \sqrt{\mu_j}} XQ_L \alpha_j, \ j = 1, \cdots, d. \tag{5.41}$$

We should make the claim that the UDP algorithm given in Section 5.2.4 is a special case of its kernel weighted version since when the kernel parameter $t = +\infty$, the weight $K_{ij} = 1$ for any i and j. For convenience, we also refer to the kernel weighted UDP version simply as UDP , and the UDP algorithm in Section 5.2.4 is denoted by UDP ($t = +\infty$).

5.2.6. Links to other Linear Projection Techniques: LDA and LPP

Comparisons with LPP

UDP and LPP are both unsupervised subspace learning techniques. Their criteria, however, are quite different. UDP maximizes the ratio of the non-local scatter (or the global scatter) to the local scatter whereas LPP minimizes the local scatter.

The local scatter criterion can be minimized in different ways subject to different constraints. One way is to assume that the projection axes are mutually orthogonal (PCA is actually based on this constraint so as to maximize the global scatter criterion). These constraint based optimization model is

$$\arg\min_{w^T w=1} f(w) = w^T \mathrm{S}_L w \tag{5.42}$$

By solving this optimization model, we get a set of orthogonal projection axes $w_1, w_2, \cdots, w_d$. The other way to minimize the local scatter criterion is to assume that the projection axes are conjugately-orthogonal. LPP is in fact based on these constraints. Let us define the matrix $S_D = XDX^T$. Then, the optimization model of LPP (based on the S_D-orthogonality constraints) is given by

$$\arg\min_{w^T \mathrm{S}_D w=1} f(w) = w^T \mathrm{S}_L w \tag{5.43}$$

which is equivalent to

$$\arg\min J_P(w) = \frac{w^T S_L w}{w^T S_D w} \;\Leftrightarrow\; \arg\max J_P(w) = \frac{w^T S_D w}{w^T S_L w} \tag{5.44}$$

Therefore, LPP essentially maximizes the ratio of $w^T S_D w$ to the local scatter. But, this criterion has no direct link to classification. Since the purpose of the constraint $w^T S_D w = w^T XDX^T w = Y^T \mathrm{S}_D Y = 1$ is just to remove an arbitrary scaling factor in the embedding and that of the matrix D is to provide a natural measure of the vertices of the adjacency graph [5.36], maximizing (or normalizing) $w^T S_D w$ does not make sense with respect to discrimination. In contrast, the criterion of UDP has a more transparent link to classification or clustering. Its physical meaning is very clear: if samples belong to a same cluster, they become closer after the projection; otherwise, they become as far as possible.

Connections to LDA

Compared with LPP, UDP has a more straightforward connection to LDA. Actually, LDA can be regarded as a special case of UDP if we assume that each class has the same number of training samples (i.e., the class *priori* probabilities are same). When the data has an ideal clustering, i.e., each local neighborhood contains exactly the same number of training samples belonging to the same class, UDP is LDA. In this case, the adjacency matrix is

$$H_{ij} = \begin{cases} 1, & \text{if } X_i \text{ and } X_j \text{ belong to the same class} \\ 0 & \text{otherwise} \end{cases} \tag{5.45}$$

And in this case, there exist *c* local neighborhoods, each of which corresponds to a cluster of samples in one pattern class. Suppose that the *k*-th neighborhood is formed by all *l* samples of Class *k*. Then, the local scatter matrix of the samples in the *k*-th neighborhood is

$$S_L^{(k)} = \frac{1}{2}\frac{1}{l^2}\sum_{i,j}(X_i^{(k)} - X_j^{(k)})(X_i^{(k)} - X_j^{(k)})^T \tag{5.46}$$

Following the derivation of S_T in Eq. 5.12, we have

$$S_L^{(k)} = \frac{1}{l}\sum_{j=1}^{l}(X_j^{(k)} - \mathrm{m}_k)(X_j^{(k)} - \mathrm{m}_k)^T \tag{5.47}$$

So, $S_L^{(k)} = S_W^{(k)}$, where $S_W^{(k)}$ is the covariance matrix of samples in Class *k*. From the above derivation and Eq. 5.20, the whole local scatter matrix is

$$\begin{aligned} S_L &= \frac{1}{2}\frac{1}{M^2}\sum_{k=1}^{c}\sum_{i,j}(X_i^{(k)} - X_j^{(k)})(X_i^{(k)} - X_j^{(k)})^T \\ &= \frac{1}{2}\frac{1}{M^2}\sum_{k=1}^{c} 2l^2 S_L^{(k)} = \frac{l}{M}\sum_{k=1}^{c}\frac{l}{M}S_W^{(k)} = \frac{l}{M}S_W \end{aligned} \tag{5.48}$$

Then, the non-local scatter matrix is

$$S_N = S_T - S_L = S_T - \frac{l}{M}S_W \tag{5.49}$$

Further, it can be shown that the following equivalent relationships hold:

$$\begin{aligned} J(w) &= \frac{w^T S_N w}{w^T S_L w} \Leftrightarrow \frac{w^T (S_T - \frac{l}{M}S_W) w}{w^T (\frac{l}{M}S_W) w} \Leftrightarrow \frac{w^T S_T w}{w^T (\frac{l}{M}S_W) w} \Leftrightarrow \frac{w^T S_T w}{w^T S_W w} \Leftrightarrow \frac{w^T S_B w}{w^T S_W w} \\ &= J_F(w) \end{aligned} \tag{5.50}$$

Therefore, UDP is LDA in the case where each local neighborhood contains exactly the same number of training samples belonging to the same class. The connection between LPP and LDA was disclosed in [5.44] provided that an Eq. (41)-like adjacency relationship is given. In addition, LPP needs another assumption, i.e., the sample mean of the data set is zero, to connect itself to LDA, while UDP does not. So, the connection between UDP and LDA is more straightforward.

5.2.7 Biometrics Applications: Experiments and Analysis

In this section, the performance of UDP is evaluated on the Yale, FERET and AR face image databases and PolyU Palmprint database and compared with the performances of PCA, LDA, and Laplacianface (LPP).

Experiment Using the Yale Database

The Yale face database contains 165 images of 15 individuals (each person providing 11 different images) under various facial expressions and lighting conditions. In our experiments, each image was manually cropped and resized to 100×80 pixels. Figure 5.7 shows sample images of one person.

Figure 5. 7 Sample images of one person in the Yale database.

Table 5. 6 The maximal recognition rates (%) of PCA, LDA, Laplacianface, and UDP on the Yale database and the corresponding dimensions (shown in parentheses) when the first 6 samples per class are used for training

Measure	PCA	LDA	Laplacianface ($t = +\infty$)	UDP ($t = +\infty$)	Laplacianface ($t^* = 800$)	UDP ($t^* = 800$)
Euclidean	90.7 (28)	86.7 (14)	90.7 (30)	96.0 (18)	90.7 (24)	97.3 (18)
Cosine	90.7 (40)	96.0 (14)	97.3 (30)	98.7 (18)	98.7 (24)	100 (18)

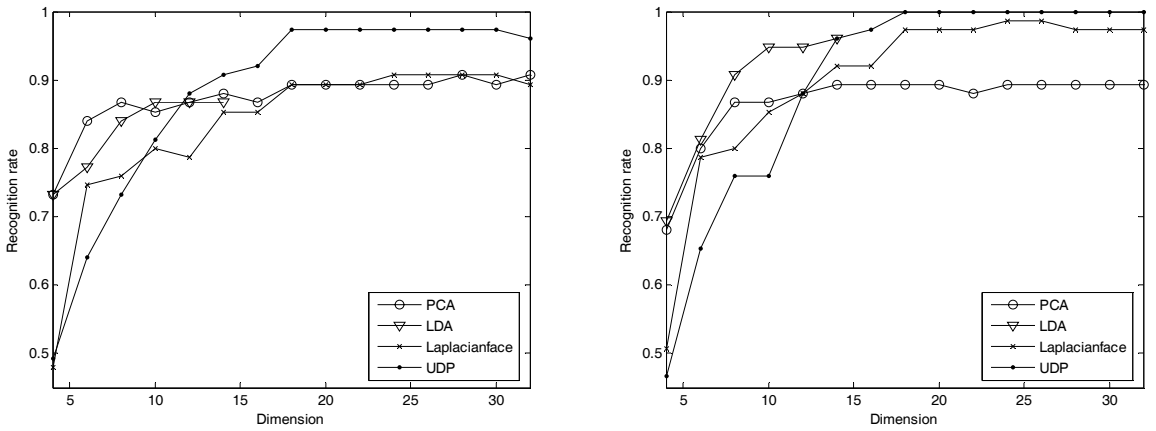
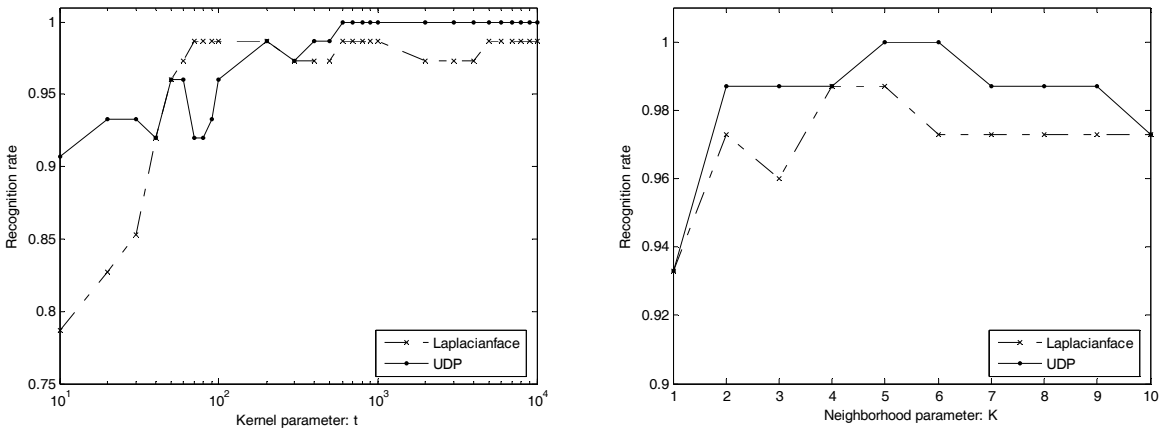


Figure 5. 8 The recognition rates of PCA, LDA, Laplacianface, and UDP versus the dimensions when (a) Euclidean distance is used and, (b) Cosine distance is used

The first experiment was performed using the first six images (i.e., center-light, with glasses, happy, left-light, without glasses, and normal) per class for training, and the remaining five images (i.e., right-light, sad, sleepy, surprised, and winking) for testing. For feature extraction, we used, respectively, PCA (eigenface), LDA (Fisherface), LPP (Laplacianface) and the proposed UDP. Note that Fisherface, Laplacianface, and UDP all involve a PCA phase. In this phase, we keep nearly 98% image energy and select the number of principal components, m, as 60 for each method. The K-nearest neighborhood parameter K in UDP and Laplacianface can be chosen as $K = l - 1 = 5$, where l denotes the number of training samples per class. The justification for this choice is that each sample should connect with the remaining $l - 1$ samples of the same class provided that within-class samples are well clustered in the observation space. There are generally two ways to select the Gaussian kernel parameter t. One way is to choose $t = +\infty$, which represents a special case of LPP (or UDP). The other way is to determine a proper parameter t^* within the interval $(0, +\infty)$ using the global-to-local strategy [5.32] to make the recognition result optimal. Finally, the nearest neighbor (NN) classifiers with Euclidean distance and cosine distance are respectively employed for classification. The maximal recognition rate of each method and the corresponding dimension are given in Table 5.6. The recognition rates versus the variation of dimensions are illustrated in Figure5.8. The recognition rates of Laplacianface and UDP versus the variation of the kernel parameter t and those versus the K-nearest neighborhood parameter K are, respectively, illustrated in Figure 5.9 (a) and (b).



(a) (b)

Figure 5. 9 (a) The maximal recognition rates of Laplacianface and UDP versus the variation of kernel parameter *t*; (b) The maximal recognition rates of Laplacianface and UDP versus the variation of K-nearest neighborhood parameter K

From Table 5.6 we can see three main points. First, UDP outperforms Laplacianface under each distance measure, whether the kernel parameter *t* is infinity or optimally chosen (t* = 800). Second, UDP and Laplacianface with cosine distances both perform better than LDA and PCA with cosine or Euclidean distances. Third, the cosine distance metric can significantly improve the performance of LDA, Laplacianface, and UDP but it has no substantial effect on the performance of PCA. Figure 5.8 shows that UDP (t* = 800) outperforms Laplacianface (t* = 800), LDA and PCA when the dimension is over 16, no matter what distance metric is used. Further, the recognition rate of UDP with cosine distance retains 100% as the dimension varies from 18 to 32. Figure 5.9 (a) indicates that the performances of UDP and Laplacianface (with cosine distance) become robust when the parameter *t* is over 200 and, UDP consistently outperforms Laplacianface when *t* is larger than 400. The recognition rate of UDP retains 100% as *t* varies from 600 to 10,000. Figure 5.9 (b) shows that the performances of UDP and Laplacianface vary with the variation of the K-nearest neighborhood parameter K. When K is chosen as $l-1=5$, both methods achieve their top recognition rates. So, we will choose $K = l-1$ for our experiments.

Why can the unsupervised method UDP (or Laplacianface) outperform the supervised method LDA? In our opinion, the possible reason is that UDP (or Laplacianface) is more robust than LDA to outliers. In the training set of this experiment, the "left-light" image of each class can be viewed as an outlier. The outlier images may cause errors in the estimate of within-class scatter and thus make LDA projection inaccurate. In contrast, UDP builds the adjacency relationship of data points using *k*-nearest neighbors and groups the data in a natural way. Most outlier images of different persons are grouped into new different clusters. By this means, the number of clusters increases, but the negative influence of outliers onto within-class scatter is eliminated. So, the resulting projection of UDP is more accurate and discriminative. Since the number of clusters increases, UDP generally needs more features than LDA to achieve its best performance. This also explains why LDA outperforms UDP using a few features, as shown in Figure

5.8.

In the second experiment, 20-fold cross-validation tests are performed to re-evaluate the performance of PCA, LDA, Laplacianface, and UDP. In each test, six images of each subject are randomly chosen for training while the remaining five images are used for testing. The parameters involved in each method are set as the same as those used in the first experiment. Table 5.7 shows the maximal average recognition rates across 20 runs of each method under nearest neighbor classifiers with two distance metrics and their corresponding standard deviations (*std*) and dimensions. From Table 5.7, it can be seen that UDP outperforms other methods and, the cosine distance metric is still helpful in improving the performance of LDA, Laplacianface, and UDP. These conclusions are on the whole consistent with those drawn from the first experiment.

Table 5. 7 The maximal average recognition rates (%) of PCA, LDA, Laplacianface, and UDP across 20 runs on the Yale database and the corresponding standard deviations (std) and dimensions (shown in parentheses)

Measure	PCA	LDA	Laplacianface ($t = +\infty$)	UDP ($t = +\infty$)	Laplacianface ($t^* = 800$)	UDP ($t^* = 800$)
Euclidean	91.7 ± 5.4 (28)	87.1 ± 9.8 (14)	89.2 ± 4.7 (44)	91.9 ± 5.1 (28)	90.3 ± 5.1 (24)	92.3 ± 5.9 (28)
Cosine	90.1 ± 6.9 (24)	92.1 ± 6.7 (14)	94.2 ± 3.3 (48)	95.1 ± 4.3 (28)	95.0 ± 2.9 (24)	95.5 ± 4.1 (28)

Since the cosine distance is more effective than the Euclidean distance for LDA, Laplacianface, and UDP, in the following experiments we use only this distance metric.

Experiment Using the FERET Database

The FERET face image database has become a standard database for testing and evaluating state-of-the-art face recognition algorithms [5.49-51]. The proposed method was tested on a subset of the FERET database. This subset includes 1000 images of 200 individuals (each one has 5 images). It is composed of the images whose names are marked with two-character strings: "*ba*", "*bj*", "*bk*", "*be*", "*bf*". This subset involves variations in facial expression, illumination, and pose. In our experiment, the facial portion of each original image was automatically cropped based on the location of eyes and mouth, and the cropped image was resized to 80×80 pixels and further pre-processed by histogram equalization. Some sample images of one person are shown in Figure 5.10.

Figure 5. 10 Samples of the cropped images in a subset of FERET database

Table 5. 8 The maximal recognition rates (%) of PCA, LDA, Laplacianface, and UDP on a subset of FERET database and the corresponding dimensions

Method	PCA	LDA	Laplacianface ($t = +\infty$)	UDP ($t = +\infty$)	Laplacianface ($t^* = 300$)	UDP ($t^* = 7000$)
Recognition rate	73.3	75.0	77.0	80.7	78.5	81.2
Dimension	85	100	105	90	90	110

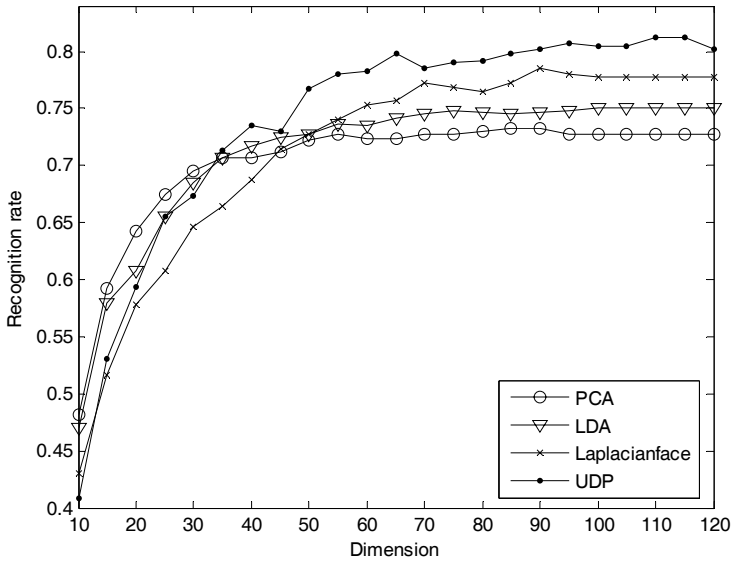


Figure 5. 11 The recognition rates of PCA, LDA, Laplacianface, and UDP versus the dimensions when cosine distance is used on a subset of FERET database

In our test, we use the first two images (i.e., "*ba*" and "*bj*") per class for training, and the remaining three images (i.e., "*bk*", "*be*" and "*bf*") for testing. PCA, LDA, Laplacianface and UDP are used for feature extraction. In the PCA phase of LDA, Laplacianface and UDP, the number of principal components, *m*, is set as 120. The K-nearest neighborhood parameter K in Laplacianface and UDP is chosen as $K = l - 1 = 1$. After feature extraction, a nearest neighbor classifier with cosine distance is employed for classification. The maximal recognition rate of each method and the corresponding dimension are given in Table 5.8. The recognition rate curve versus the variation of dimensions is shown in Figure5.11. Table 5.8 demonstrates again that UDP outperforms PCA, LDA and Laplacianface, whether the kernel parameter *t* is infinity or optimally chosen ($t^* = 7000$ for UDP and $t^* = 300$ for Laplacianface). Figure 5.11 indicates that UDP consistently performs better than other methods when the dimension is over 45.

Experiment Using the AR Database

The AR face [5.52, 5.53] contains over 4,000 color face images of 126 people (70 men and 56 women), including frontal views of faces with different facial expressions, lighting conditions and occlusions. The pictures of 120 individuals (65 men and 55 women) were taken in two sessions (separated by two weeks) and each section contains 13 color images. 20 face images (each session containing 10) of these 120 individuals are selected and used in our experiment. The face portion of each image is manually cropped

and then normalized to 50×40 pixels. The sample images of one person are shown in Figure 5.12. These images vary as follows: (a) neutral expression, (b) smiling, (c) angry, (d) screaming, (e) left light on, (f) right light on, (g) all sides light on; (h) wearing sun glasses, (i) wearing sun glasses and left light on, and (j) wearing sun glasses and right light on.

Figure 5. 12 Samples of the cropped images of one person in the AR database

Table 5. 9 The maximal average recognition rates (%) and standard deviations (std) of PCA, LDA, Laplacianface and UDP with different training sample sizes on the AR database

# /class	PCA	LDA	Laplacianface ($t = +\infty$)	UDP ($t = +\infty$)	Laplacianface ($t^* = 300$)	UDP ($t^* = 500$)
2	71.2 ± 6.0	70.7 ± 11.5	75.5 ± 8.1	76.7 ± 7.7	75.6 ± 8.1	76.6 ± 7.9
3	74.4 ± 5.7	82.1 ± 13.5	85.1 ± 7.4	86.9 ± 8.0	85.2 ± 7.5	86.8 ± 7.9
4	80.2 ± 6.0	91.2 ± 11.4	91.7 ± 4.5	93.3 ± 4.7	91.7 ± 4.5	93.2 ± 4.9
5	81.4 ± 6.2	93.9 ± 8.0	92.6 ± 4.9	93.9 ± 5.1	92.5 ± 5.0	93.9 ± 5.0
6	84.5 ± 4.3	96.7 ± 2.4	94.2 ± 2.7	95.5 ± 2.0	94.1 ± 2.9	95.6 ± 2.0

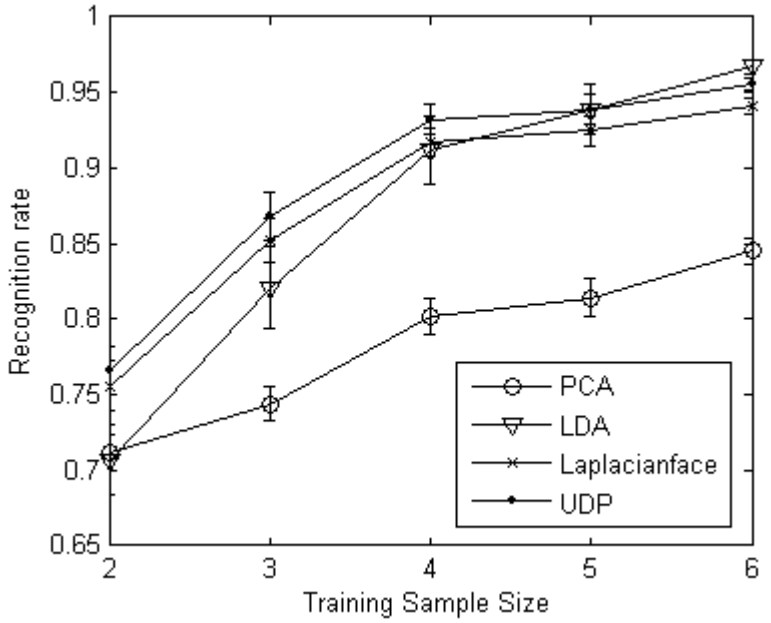


Figure 5. 13 The maximal average recognition rates of PCA, LDA, Laplacianface and UDP versus the variation of the training sample size

In our experiments, l images (l varies from 2 to 6) are randomly selected from the image gallery of each individual to form the training sample set. The remaining $20-l$ images are used for testing. For each l, we perform cross-validation tests and run the system 20 times. PCA, LDA, Laplacianface and UDP are, respectively, used for face representation. In the PCA phase of LDA, Laplacianface and UDP, the number of principal components, m, is set as 50, 120, 180, 240, and 300, respectively corresponding to l = 2, 3, 4, 5, and 6. The K-nearest neighborhood parameter K in Laplacianface and UDP is chosen as K = $l-1$. Finally, a nearest-neighbor classifier with cosine distance is employed for classification. The maximal average recognition rate and the *std* across 20 runs of tests of each method are shown in Table 5.9. The recognition rate curve versus the variation of training sample sizes is shown in Figure 5.13.

From Table 5.9 and Figure 5.13, we can see first that UDP overall outperforms Laplacianface, whether the kernel parameter is infinity or optimally chosen and second that as unsupervised methods, UDP and Laplacianface both significantly outperform PCA, irrespective of the variation in training sample size. These two points are consistent with the experimental results on the Yale and the FERET databases. In addition, we can see some inconsistent results. First, with reference to the impact of the kernel weighting on the performance of UDP and Laplacianface, in this experiment, UDP and Laplacianface both perform well without kernel weighting (i.e. $t=+\infty$). The heat-kernel (i.e. Gaussian kernel) weighting, by optimally choosing t^* = 300 for Laplacianface and t^* = 500 for UDP from the interval $(0, +\infty)$, however, does little to improve the recognition accuracy.

Another inconsistent point that is worth remarking concerns the performance comparison of UDP and LDA. UDP outperforms LDA when l is less than 5, while LDA outperforms UDP when l is over 5. This means that once the given training sample size per class becomes large, LDA may achieve better results than UDP. It is not hard to interpret this phenomenon from a statistical point of view. While there are more and more samples per class provided for training, the within-class scatter matrix can be evaluated more accurately and becomes better-conditioned, so LDA will become more robust. However, with the increase of the training sample size, more boundary points might exist between arbitrary two data clusters in input space. This makes it more difficult for UDP (or LPP) to choose a proper locality radius or the K-nearest neighborhood parameter K to characterize the "locality".

Nevertheless, UDP does have an advantage over LDA with respect to a specific biometrics problem like face recognition. Figure 8 indicates that the smaller the training sample size is, the more significant the performance difference between UDP and LDA becomes. This advantage of UDP in small sample size cases is really helpful in practice. This is because face recognition is typically a small sample size problem. There are generally few images of one person provided for training in many real-world applications.

Experiment Using the PolyU Palmprint database

The PolyU palmprint database contains 600 grayscale images of 100 different palms with six samples for each palm (*http://www4.comp.polyu.edu.hk/~biometrics/*). Six samples from each of these palms were collected in two sessions, where the first three were

captured in the first session and the other three in the second session. The average interval between the first and the second sessions is two months. In our experiments, the central part of each original image was automatically cropped using the algorithm mentioned in [5.54]. The cropped images were resized to 128×128 pixels and pre-processed using histogram equalization. Figure 5.14 shows some sample images of two palms.

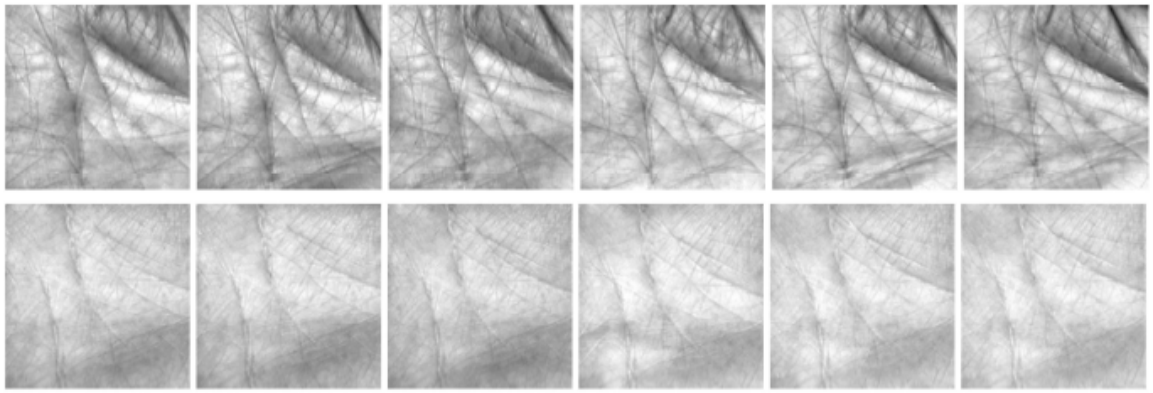


Figure 5. 14 Samples of the cropped images in PolyU Palmprint database

According to the protocol of this database, the images captured in the first session are used for training and the images captured in the second session for testing. Thus, for each palm class, there are three training samples and three testing samples. PCA, LDA, Laplacianface and UDP are used for palm feature extraction. In the PCA phase of LDA, Laplacianface and UDP, the number of principal components, m, is set as 150. The K-nearest neighborhood parameter K in Laplacianface and UDP is chosen as $K = l - 1 = 2$. After feature extraction, a nearest neighbor classifier with cosine distance is employed for classification. The maximal recognition rate of each method and the corresponding dimension are listed in Table 5.10. The recognition rate curve versus the variation of dimensions is shown in Figure 5.15

Table 5. 10 The maximal recognition rates (%) of PCA, LDA, Laplacianface, and UDP on PolyU Palmprint database and the corresponding dimensions

Method	PCA	LDA	Laplacianface ($t = +\infty$)	UDP ($t = +\infty$)	Laplacianface ($t^* = 300$)	UDP ($t^* = 200$)
Recognition rate	86.0	95.7	98.3	99.3	99.0	99.7
Dimension	90	90	140	90	100	100

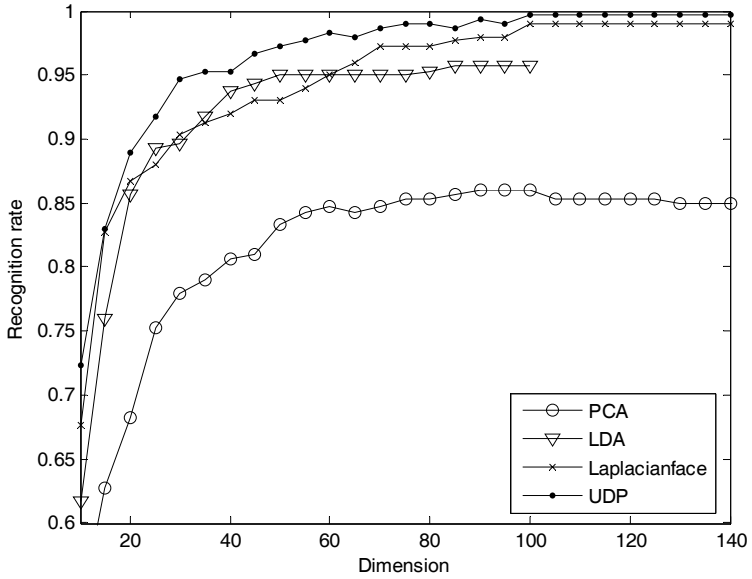


Figure 5. 15 The recognition rates of PCA, LDA, Laplacianface, and UDP versus the dimensions when cosine distance is used on PolyU Palmprint database

From Table 5.6, we can see that UDP outperforms PCA, LDA and Laplacianface. The recognition rate of UDP (when $t^* = 200$) is up to 99.7%, i.e., only one sample was missed. Figure 5.11 shows that UDP consistently performs better than other methods, irrespective of the dimensional variation. These results demonstrate that UDP is also a good tool for palm recognition.

5.2.8 Conclusions and Future Work

In this chapter, we develop an unsupervised discriminant projection (UDP) technique for dimensionality reduction of high-dimensional data in small sample size cases. The projection of UDP can be viewed as a linear approximation of the nonlinear map that uncovers and separates embeddings corresponding to different manifolds in the final embedding space. UDP considers the local and non-local scatters at the same time and seeks to find a projection maximizing the ratio of the non-local scatter to the local scatter. The consideration of the non-local quantity makes UDP more intuitive and more powerful than LPP for classification or clustering tasks. Our experimental results on three popular face image databases and one palmprint database demonstrate that UDP is more effective than LPP and PCA. In addition, UDP is more discriminative than LDA when the training sample size per class is small.

Our experimental results on the AR database, however, also reveal a drawback of UDP (LPP actually has the same problem). That is, as the training sample size per class becomes large, LDA can outperform UDP. This problem is unnoticeable in most real-world biometrics applications since the given training sample size is always very small. But, it may become prominent once UDP is applied to large sample size problems. To address this, we need a more precise characterization of the local scatter and the non-local scatter when the given training sample size per class is relatively large. A possible

way is to use the provided class label information (for example, borrowing Yan [5.45]'s and Chen [5.46]'s ideas) to facilitate this characterization and then to build a semi-supervised hybrid system.

As a generator of weighting coefficients, the Gaussian kernel (or heat kernel) is examined in this chapter. It is demonstrated to be effective in most cases. But, in some cases, it fails to improve the performance of UDP or LPP. Are there more effective kernels for weighting the proposed method? This is a problem deserving further investigation. In addition, in this chapter, we focus on developing a linear projection technique and applying it to biometrics but do not address another interesting problem, i.e., modeling multi-manifolds for classification purposes. When different classes of data lie on different manifolds, it is of central importance to uncover the embeddings corresponding to different manifolds and, at the same time, to make different embeddings as separable as possible in the final embedding space. We will address this problem and try to build a general framework for classification-oriented multi-manifolds learning in the near future. This framework may result in more effective features for biometrics tasks.

5.3 Mutual neighborhood based discriminant projections

Many machine learning and data mining applications, dimension reduction is used to transform the high dimensional data to a low dimensional space so that the retrieval [5.55], [5.64], [5.88], [5.90], [5.93], the storage [5.56], [5.63] and the analysis [5.57], [5.61], [5.62], [5.65] of data can be made more efficient. Over the last few years, the technique of dimension reduction has aroused considerable interests in the areas of data mining, machine learning, and pattern recognition [5.68], [5.71], [5.86].

In literature, linear discriminant analysis [5.66] is perhaps one of the most popular methods for supervised dimension reduction. It utilizes the variance and the center information for feature analysis. The Fisher-Raleigh criterion of LDA is theoretically optimal in the Bayes sense when the class conditional densities of the data set are Gaussian and share the same covariance. It is popular in practice due to its mathematical simplicity and computational efficiency. However, there is a problem in LDA that arises from its parametric nature. It is generally known that in real world applications the training data obtained may sometimes be under-sampled or not strictly Gaussian. In such case, the employment of the LDA criterion, which has its root from the Bayes error probability, cannot be expected to indicate accurately which features should be extracted to preserve the complex data structure for classification. The obtained projection vectors might be sub-optimal and generate large classification errors. In the experimental section, we demonstrate this fact in several application domains.

Fukunaga developed a nonparametric form of discriminant analysis (NDA) to address this problem [5.66, 5.67]. The NDA model does not rely on any assumption concerning the structure of the data density. It is more flexible and robust than LDA for practical applications. But in NDA a heuristic based weighting function has to be introduced to deemphasize the samples that are far away from the class boundaries. This weighting

function takes a free control parameter whose value may range from zero to the infinity. It is however quite difficult in practice to choose an optimal value for this parameter without the prior knowledge of the density structure of the sample space. Yan et al. [5.89] also suggested using the boundary samples for feature analysis but they did not show how to effectively find out such sample vectors for training.

Recently, the algorithms based on the locality preserving projections have provided an alternative for nonparametric discriminant analysis. They are the natural extensions of the manifold learning theory where the concept of 'locality preserving' is central [5.58], [5.79], [5.82]. The basic idea of these algorithms is that global nonlinear data structures are locally linear. By using the kNN algorithm the sample space is partitioned into a number of small size neighborhoods. The discriminant information that exists in the global nonlinear data structure can be better analyzed at the local scale and extracted for classification. Using the locality preserving projections He et al. has developed the Laplacianfaces method (LPP) [5.69] which is an unsupervised dimension reduction technique that can preserve the local affinity of the observations in the transformed low dimensional space. Yu et al. made extensions to this method by utilizing the class information for training and developed the discriminant locality preserving projection algorithm (DLPP) for supervised dimension reduction [5.94].

The locality based methods are quite effective for linear feature extraction [5.69], [5.94]. But in these methods it is required to determine not only the optimal size of the kNN neighborhood but also the parameters of the kernel functions that control the degrading rate of the affinity of the neighboring samples. This increases the difficulty for model selection and the computation cost is also high especially for large scale applications.

To overcome these problems we propose in this chapter a new method called mutual neighborhood based discriminant projection (MNDP) for nonparametric dimension reduction. The main characteristic of the new algorithm is that it constructs and utilizes the mutual k nearest neighborhoods to describe the boundary structure of the data classes for supervised dimension reduction. By constructing the between class mutual neighborhoods we can efficiently find out those samples that are close to the class boundaries and select them for training. The discriminant features can then be learned by maximizing the average distances between the marginal vectors of different classes.

The MNDP is nonparametric, so, it needs not to make any assumption on the data density. Moreover, it has the following advantages over the existing methods. First, compared with the conventional LDA the number of the discriminant components that can be obtained in MNDP is not restricted by the number of the data classes (or the rank of the between-class scatter matrice). Second, compared with NDA, the MNDP does not have to use the weighting function for training sample selection. Thus, we need not have to estimate the control parameter in the weighting function, making MNDP more convenient and efficient for real world applications. Third, unlike the methods based on the locality preserving projections the new algorithm does not involve any parameter to be specified for the kernel functions. In fact, in MNDP there is only one parameter that

needs to be controlled, i.e., the neighborhood size k, a small positive integer that can be easily configured and adjusted in practice.

Extensive experiments have been performed on a variety of benchmark databases to test the performance of MNDP. First, it is evaluated on the two well known face image databases, the Yale, and the AR. The popular Eigenfaces and the Fisherfaces methods are implemented for comparison. Then, we generalize the linear algorithm for nonlinear feature extraction by utilizing the kernel based method. The USPS handwritten digit database is used for testing and evaluation. The effectiveness of the proposed MNDP and the kernel MNDP methods have been demonstrated in the experiments.

The remaining of this paper is organized as follows. Section 2 will introduce briefly the two methods closely related to our work, LDA and NDA. In Section 3 the proposed algorithm MNDP will be described in detail. The kernel MNDP is developed for nonlinear discriminant analysis by using the kernel based method. In Section 4, the experiments on the face and the handwritten digit databases are performed for performance evaluation. Finally, conclusion will be drawn in Section 5.

5.3.1 Related works

In this section, we introduce briefly the two methods closely related to our work, LDA and NDA. LDA is a parametric method that makes Gaussian assumptions on the data density for feature analysis, whereas NDA is nonparametric and makes no density assumption. The key notations used in the remainder of this paper are listed in Table I.

Table 5. 11 Summary of key notations used in the paper

Notation	Description	Notation	Description
A	coefficients of the projection vectors of kernel MNDP	Q	k farthest neighborhood
C	number of data classes	W	projection vectors of MNDP
D	number of features after dimension reduction	S_B	between-class scatter matrix of LDA
L	number of sample pairs in k farthest neighborhood	S_W	within-class scatter matrix of LDA
M	number of sample pairs in mutual neighborhood	$\widetilde{S}_B$	between-class scatter matrix of NDA
N	number of training samples	S_N	mutual kNN scatter matrix of MNDP
P	mutual k nearest neighborhood	S_F	kFN scatter matrix of MNDP

Nonparametric discriminant analysis

Fukunaga et al. [5.66, 5.67] developed a nonparametric discriminant analysis method to address the problem in LDA. In NDA, a weighting function is introduced to assign

weights to the between-class sample pairs to deemphasize those samples that are far away from the class boundaries. For a typical two class problem, the objective function of NDA is defined as,

$$W_{opt} = \arg\max_{W} \frac{\left|W^T \widetilde{S}_B W\right|}{\left|W^T S_W W\right|}, \tag{5.52}$$

where

$$\widetilde{S}_B = \frac{1}{N_1}\sum_{u=1}^{N_1} w_u (x_u^1 - m_2(x_u^1))(x_u^l - m_2(x_u^1))^T + \frac{1}{N_2}\sum_{v=1}^{N_2} w_v (x_v^2 - m_1(x_v^2))(x_v^2 - m_1(x_v^2))^T \quad .$$

N_1 and N_2 are the number of samples of class 1 and class 2. x_u^1 and x_v^2 denote the samples of class 1 and class 2. $m_2(x_u^1)$ and $\mathrm{m_1(x_v^2)}$ are the mean centers of the k nearest neighbors of x_u^1 and x_v^2 in class 2 and class 1, respectively. w_u is the weight used to deemphasize the sample far away from the class boundary,

$$w_u = \frac{\min\{d^\alpha(x_u, x_{kNN}^1), d^\alpha(x_u, x_{kNN}^2)\}}{d^\alpha(x_u, x_{kNN}^1) + d^\alpha(x_u, x_{kNN}^2)}. \tag{5.53}$$

In formula 5.53, x_{kNN}^1 and x_{kNN}^2 are the k-th nearest neighbor of x_u in class 1 and class 2, $d(\cdot,\cdot)$ is a distance measure, $d(x_u, x_{kNN}^1)$ and $d(x_u, x_{kNN}^2)$ are the radii of the kNN neighborhoods, α is a free control parameter that ranges from zero to the infinity. This weighting function is expected to have the property that near the classification boundary it takes on large values and drops off to zero as we move away. The control parameter, α, adjusts how rapidly w_u will fall to zero. However, there are problems in using such a weighting function for NDA. First, in formula 5.53 there are two parameters, i.e., the control parameter α and the size of the neighborhood k that have to be both identified. We have to try all the possibilities of the combination of α and k to find out the optimal one leading to the best performance. This is computationally expensive and inconvenient for users in large scale applications. Second, for the data that are not strictly Gaussian it is quite possible that the data density near to the class boundary is high whereas when we move away from the boundary the density will drop. In such a case, formula 5.53 will assign large weights to the samples that are far away from the class boundary because in the regions of low density the radius of the kNN neighborhood is large, thus generating large weight using formula 5.53. The training samples that are far away from the class boundary, though providing no useful information for classification, will be mistakenly emphasized. Third, the control parameter α determines the distance measure exponentially. It is important to choose a correct value for α carefully to ensure that the weight will drop neither too slowly nor too fast. But in practice α is a real number whose value ranges from zero to the infinity, it is not easy to find out exactly the optimal value for it. It is also computationally inefficient in extending the NDA algorithm to handle the mutli-class problem by using the one-against-all strategy. The one-against-all strategy has been previously applied to the training of Support Vector Machines for classification [5.86]. When using it for NDA, however, we have to find out the proper value of α for each data class in order to fine-tune the performance of the algorithm. When the number

of the data classes is large a considerable amount of computation resources has to be allocated for the model selection.

5.3.2 Proposed algorithm

To address the problems in the existing methods, we describe in detail the proposed MNDP method in this section. The main feature of the new algorithm is that it constructs and utilizes the mutual k nearest neighborhoods to characterize the boundary structure of the data classes. The dimension of data is reduced by maximizing the average distance of the boundary sample vectors while minimizing the k farthest within class scatter.

Concept of mutual k-nearest neighborhood

First, let's consider the simple two class problem. Let X denote the space of observations, $X \subseteq R^n$. X_1, $X_2 \subset X$, $X_1 \cap X_2 = \phi$ are the two classes of training samples. We define the between-class mutual k-nearest neighborhood as the set of the neighboring sample pairs, i.e.,

$$P = \{(x_i^1, x_j^2) \mid x_i^1 \in X_1, x_j^2 \in X_2, x_i^1 \in kNN(x_j^2), x_j^2 \in kNN(x_i^1)\}, \tag{5.54}$$

where x_i^1 and x_j^2 denote the *i*-th and the *j*-th sample in classes X_1 and X_2, respectively. $kNN(x_i^1)$ represents the set of the k-nearest neighbors of the sample x_i^1 in X_2. Likewise, $kNN(x_j^2)$ is the set of the k nearest neighbors of x_j^2 in X_1. By choosing the proper neighborhood size k we can efficiently find out the samples lying close to the boundaries of the two data classes, as illustrated in Figure 5.16.

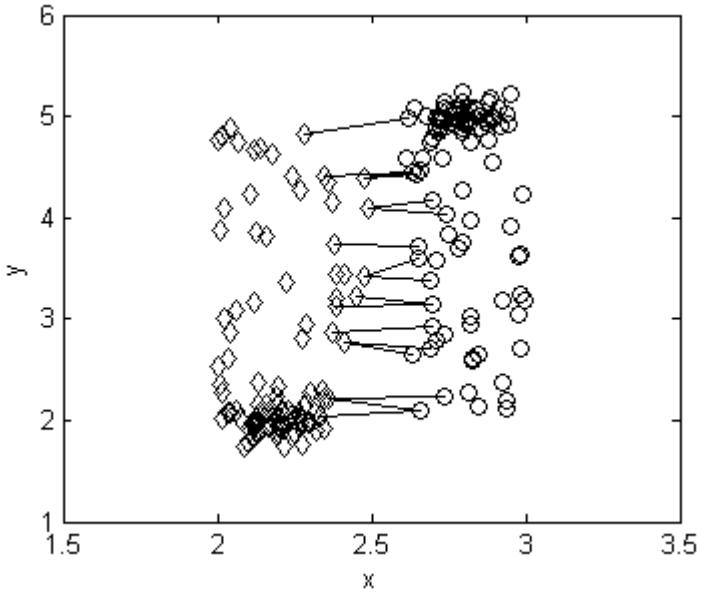


Figure 5. 16 Mutual k-nearest neighborhood of two data classes.

Figure 5.16 shows the two classes of data that are not strictly Gaussian. For each sample vector in class 1 (diamond), x, we can find out its k-nearest neighbors in class 2 (circle). On the other hand, if any of the neighbors of x, say y, has also x as one of its k-nearest neighbors in class 1, we then pair-up x with y (straight line) and add this new sample pair to the mutual neighborhood of the data set. The sample pairs in the mutual neighborhood will then be used for training.

For the multi-class problem we can construct the mutual neighborhood in two steps. First, we use the one-against-all strategy to construct the mutual neighborhood for each data class. Then, we take the union set to obtain the mutual neighborhood for the entire training set. To describe it mathematically, let there be C classes, $X_1, ..., X_C \subset X$, and let P_m be the mutual neighborhood constructed for the class X_m, i.e.,

$$P_m = \{(x_i^m, y_j) \mid x_i^m \in X_m, y_j \in X - X_m, x_i^m \in kNN(y_j), y_j \in kNN(x_i^m)\}, \quad (5.55)$$

where x_i^m represents the *i*-th sample in the *m*-th class and y_j is the *j*-th sample that belongs to $X - X_m$. The mutual neighborhood constructed for the entire data set is thus defined as,

$$P = P_1 \cup P_2 \cup ... \cup P_C. \quad (5.56)$$

Within-class Mutual k farthest neighborhood

To describe the nonparametric within-class scatter we construct and utilize the k farthest neighborhood. For the C-class problem the k farthest neighborhood for the entire training set is defined as,

$$Q = Q_1 \cup Q_2 \cup ... \cup Q_C, \quad (5.57)$$

where Q_m is the k farthest neighborhood constructed for the *m*-th class, i.e.,

$$Q_m = \{(x_i, x_j) \mid x_i \in X_m, x_j \in X_m, x_i \in kFN(x_j)\}, \quad (5.58)$$

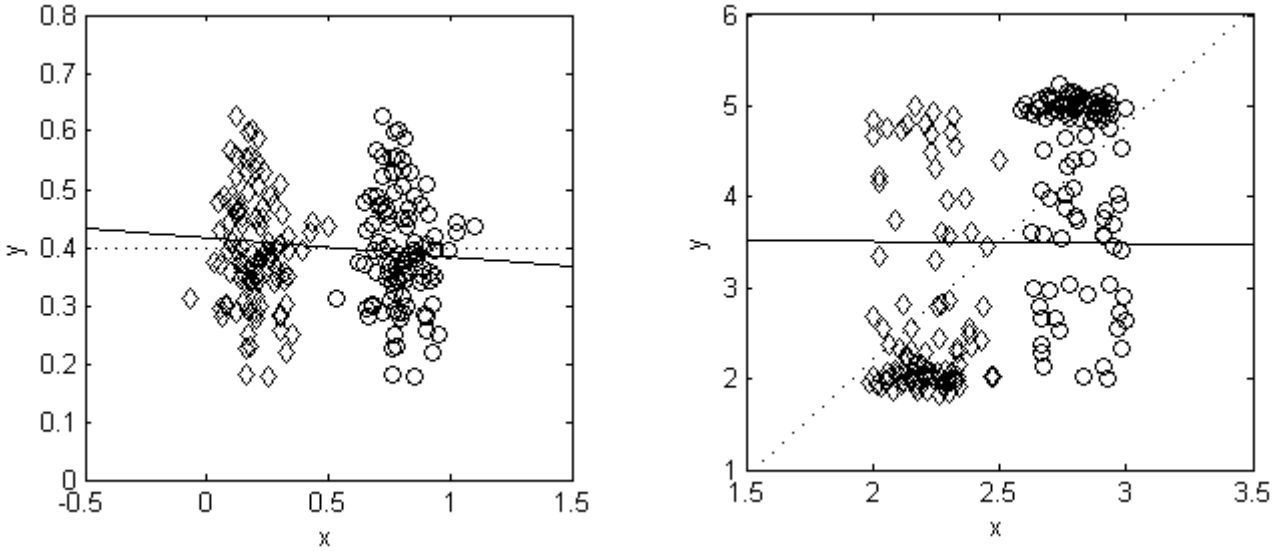
here $x_i \in kFN(x_j)$ indicates that x_i is among the k farthest neighbors of x_j, X_m represents the m-th data class.

5.3.3 Formulation of MNDP

The objective function of MNDP is to maximize the average distance between the neighboring sample pairs in the constructed mutual neighborhood while minimizing the nonparametric within-class scatter. The proposed MNDP criterion is formulated as,

$$W_{opt} = \arg\max_{W} \frac{\left|W^T S_N W\right|}{\left|W^T S_F W\right|} \tag{5.59}$$

where S_N is called the between-class mutual kNN scatter matrix, $S_N = \frac{1}{M}\sum_{i=1}^{M}\left(x_1^i - x_2^i\right)\left(x_1^i - x_2^i\right)^T$, M is the number of the sample pairs in the mutual neighborhood, and $\left(x_1^i, x_2^i\right) \in P$. S_F is called the within-class kFN scatter matrix, $\mathrm{S_F} = \frac{1}{\mathrm{L}}\sum_{\mathrm{j}=1}^{\mathrm{L}}\left(\mathrm{x}_1^{\mathrm{j}} - \mathrm{x}_2^{\mathrm{j}}\right)\left(\mathrm{x}_1^{\mathrm{j}} - \mathrm{x}_2^{\mathrm{j}}\right)^{\mathrm{T}}$, L is the number of the sample pairs in the k-furthest neighborhood, and $\left(x_1^j, x_2^j\right) \in Q$. The optimal transformation of MNDP, W_{opt}, consists of the leading eigenvectors of the matrix $S_F^{-1}S_N$. Given a testing pattern x, we can obtain its features by taking the projection $f = W_{opt}^T x$ and use this feature vector for classification. Figure 5.17 uses the simulated two dimensional data to visualize the first discriminant component (eigen vector) corresponding to the largest eigenvalue obtained in LDA and MNDP, respectively.



(a) (b)

Figure 5. 17 Illustration of the first component of LDA (dot line) and MNDP (solid line) on two dimensional data set (diamonds and circles). (a) two classes of Gaussian data, both LDA and MNDP can work effectively (b) two classes of data that do not follow the Gaussian distribution strictly, LDA fails whereas MNDP can still work effectively to separate the classes

In Figure 5.17 the two classes of data are linearly separable. Each class contains one hundred samples. In Figure5.17 (a), the two classes are both Gaussian. We can see that both the LDA and the proposed MNDP method can work effectively to find out the correct projection vectors. In Figure 5.17 (b), it illustrates the case when the two classes of data are not strictly Gaussian. The LDA fails to separate the two classes because for this set of data the estimated class centers do not reflect its true density anymore, the

LDA component obtained is biased from the optimal direction of the projection. In contrast, our proposed MNDP method is robust to the variation of the data density. It can still work well to separate the two data classes because it does not rely on the density but the boundary information of the data for feature analysis.

5.3.4 Kernel MNDP

The proposed linear MNDP algorithm is effective in separating the data classes that are linearly separable. But like other linear algorithms it fails to separate the data that may have strong nonlinearity. Recently, the kernel based method has received a lot of attentions, leading to the development of a spectrum of the kernel based methods [5.59], [5.73], [5.75], [5.76], [5.77], [5.80], [5.81], [5.84], [5.85], [5.91], [5.92], [5.95] for dimension reduction and classification, e.g., the kernel PCA , the kernel FDA, and the Support Vector Machines. In this section, we apply the kernel technique to develop the kernel MNDP algorithm for nonlinear dimension reduction.

Fundamentals of kernel method

Given a training sample $x \in R^n$ in the observation space, which is possibly nonlinearly mixed, we transform it to the feature space H via the following implicit nonlinear mapping,

$$\begin{aligned} &\Phi : R^n \to H \\ &x \mapsto \Phi(x). \end{aligned} \tag{5.60}$$

From Cover's theorem [5.84, 85] on the separability of patterns, it is known that the nonlinearly separable patterns in an input space are linearly separable with high probability if the input space is transformed nonlinearly to a high dimensional feature space. So after performing the nonlinear mapping we just need to do linear MNDP in the space H to compute the optimal projection vectors. Let us denote the MNDP transformation in the high dimensional space as $W^{\Phi} : H \to F$, $\Phi(x) \mapsto W^{\Phi} \cdot \Phi(x)$. From the theory of reproducing kernel Hilbert space [5.80, 81], we also know that any solution $W_k^{\Phi} \in H$ must lie in the span of all the training samples. Thus, there exists an expansion for W_k^{Φ} of the form,

$$W_k^{\Phi} = \sum_{i=1}^{N} \alpha_{ki} \Phi(x_i). \tag{5.61}$$

In matrix form, we have $W^{\Phi} = A\Phi$, where $A = (\alpha_1^T, ..., \alpha_D^T)^T$, $\alpha_k = (\alpha_{k1}, ..., \alpha_{kN})^T \in R^N$ and $\Phi = \left(\Phi(x_1)^T, ..., \Phi(x_N)^T\right)^T$, D is the number of the features. Now in the high dimensional space H, for a given pattern $\Phi(x)$ W^{Φ} transforms it to the L-dimensional feature vector Y,

$$Y = A\Phi \cdot \Phi(x). \tag{5.62}$$

Kernelizing the MNDP criterion

In the kernel induced high dimension space we can reformulate the MNDP in the kernel induced high dimensional space, i.e.,

$$A_{opt} = \max_{A} \frac{\left| W^{\Phi} S_N^{\Phi} W^{\Phi T} \right|}{\left| W^{\Phi} S_F^{\Phi} W^{\Phi T} \right|}, \tag{5.63}$$

where we define the between-class mutual kNN scatter matrice S_N^{Φ} as,

$$S_N^{\Phi} = \frac{1}{M} \sum [\Phi(x_i) - \Phi(x_j)][\Phi(x_i) - \Phi(x_j)]^T \ , \ (x_i, x_j) \in P \tag{5.64}$$

Similarly, the within-class kFN scatter matrice is,

$$S_F^{\Phi} = \frac{1}{L} \sum [\Phi(x_u) - \Phi(x_v)][\Phi(x_u) - \Phi(x_v)]^T \ , \ (\mathrm{x_u}, \mathrm{x_v}) \in \mathrm{Q_i} \, . \tag{5.65}$$

Kernel trick

Note that in order to find out the optimal W^{Φ} we have to evaluate the two matrices S_N^{Φ} and S_F^{Φ}. Since we do not know the explicit form of the nonlinear mapping $\Phi(\cdot)$ the dot product cannot be directly computed. Fortunately, by taking advantage of the Mercer's theorem [5.80, 5.81] we can obtain the result of the dot product by using the kernel trick. Let us denote $K(\cdot,\cdot)$ as the mercer kernel, a continuous, symmetric and semi-positive function. The inner product of the two patterns in the high dimensional space is thus equal to,

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \, . \tag{5.66}$$

In practice several types of kernel functions such as the Gaussian RBF kernel function, the Inverse multi-quadric function and the polynomial function, can be adopted [5.55], [5.76], [5.80], [5.81], [5.84], [5.85]. By applying the kernel trick, we convert the objective function of kernel MNDP into the following form,

$$A_{opt} = \max_{A} \frac{\left| A \Xi A^T \right|}{\left| A \Psi A^T \right|} \tag{5.67}$$

where

$$\begin{aligned}\Xi &= \Phi S_N^{\Phi} \Phi^T \\ &= \Phi\left\{\sum [\Phi(x_i) - \Phi(x_j)][\Phi(x_i) - \Phi(x_j)]^T\right\}\Phi^T \\ &= \sum [\Phi\Phi(x_i) - \Phi\Phi(x_j)][\Phi\Phi(x_i) - \Phi\Phi(x_j)]^T \\ &= \sum (K_i - K_j)(K_i - K_j)^T\end{aligned} \tag{5.68}$$

and $(x_i, x_j) \in P$. In a similar way, we can construct the matrice Ψ, where

$$\begin{aligned}\Psi &= \Phi S_F^{\Phi} \Phi^T \\ &= \Phi\left\{\sum [\Phi(x_u) - \Phi(x_v)][\Phi(x_u) - \Phi(x_v)]^T\right\}\Phi^T \\ &= \sum [\Phi\Phi(x_u) - \Phi\Phi(x_v)][\Phi\Phi(x_u) - \Phi\Phi(x_v)]^T \\ &= \sum (K_u - K_v)(K_u - K_v)^T\end{aligned} \tag{5.69}$$

where $(x_u, x_v) \in Q$. K_i denotes the i-th row vector of the kernel matrice whose elements are the inner products of the training samples in the high dimensional space. Finally, the optimal solution of can be found by solving the following eigen equation,

$$\Psi^{-1}\Xi A = \Gamma A . \tag{5.70}$$

where Γ is the diagonal matrice of the eigen values of the matrice $\Psi^{-1}\Xi$. After we solve Eq. 5.70 we can reduce the dimension of a presented pattern by taking the transformation,

$$s = A_{opt}^T \Phi \cdot X . \tag{5.71}$$

where A_{opt} is the matrice whose column vectors are the leading vectors of the matrice $\Psi^{-1}\Xi$.

5.3.5 Experimental result

We performed extensive experiments on three real world benchmark databases to test and evaluate the performance of the proposed MNDP and the kernel MNDP methods. The MNDP algorithm has demonstrated its effectiveness, as compared with the popular Eigenfaces and the Fisherfaces methods, in solving the face recognition problems on the AR and the Yale face image databases [5.74], [5.87], respectively. The kernel MNDP algorithm was tested by using the U.S. Postal Service handwritten numeral database [5.70]. The LDA, the kernel FDA and the MNDA were implemented for comparison. The statistics of the databases used in our experiments is listed in Table 5.12.

Table 5. 12 Statistics of databases used in experiments

Experiments		Num. of classes	Num. of training samples of each class	Num. of dimensions of data
Face recognition	AR	126	14	2000
	Yale	15	11	2000
Handwritten digit recognition	USPS digits	10	100	256

The experiments were performed on a Pentium 4 2.6GHz PC with 512MB RAM memory under Matlab 7.1 platform.

MNDP for Face Recognition: AR Database

The AR face database contains over 4,000 face images of 126 individuals taken in two time sessions under the variations of illumination, facial expression and occlusion conditions. Each person has 26 images. In our experiment we consider using a subset of 14 images of each person for training and testing. Figure 5.18 shows the selected sample images of one subject.

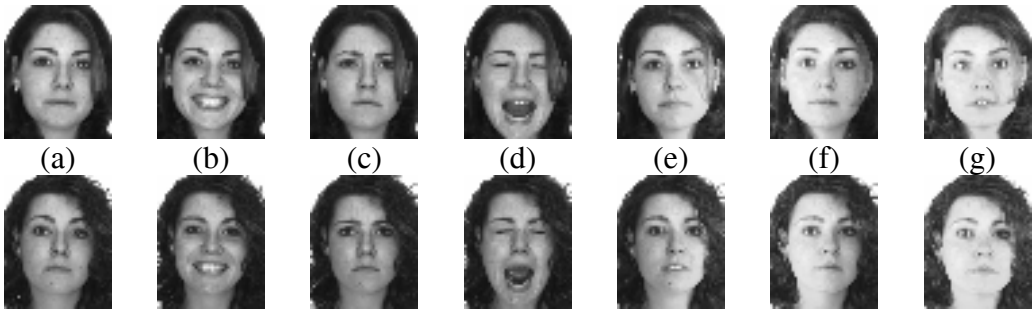


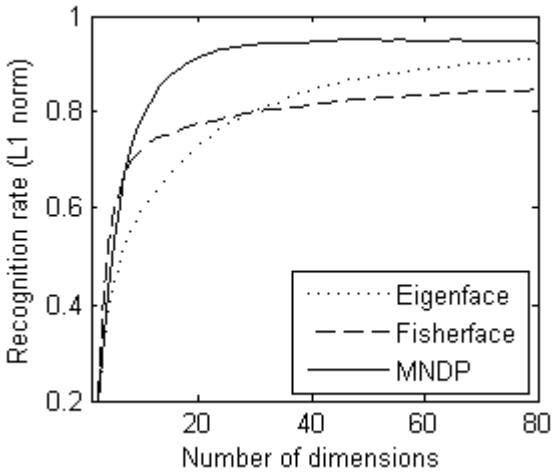
Figure 5. 18 Sample images for one subject of the AR database

In Figure 5.18, the images (a)-(g) and (n)-(t) are drawn from the first and the second time sessions respectively. For each session the first four images (a)-(d) and (n)-(q) involve the variation of facial expressions (neutral, smile, anger, scream) while the images (e)-(g) and (r)-(t) are taken under different lighting conditions (left light on, right light on, all sides light on). The images are manually cropped and normalized to 50×40 pixels. We perform twenty fold cross validations in the experiments and set the neighborhood size to be 3. For each round of testing, seven images are randomly chosen to generate the training set and the remaining samples are used for testing purpose. Therefore, there are totally 882 images in the training and the testing sets. After reducing the dimensionality of the data the three distance measures, the L_1 and the L_2 norms and the Cosine are used for pattern matching. We compute the average top recognition rate of the twenty folds of tests. The accuracy rate, the standard deviation, and the number of the discriminant components used for classification are shown in Table 5.13.

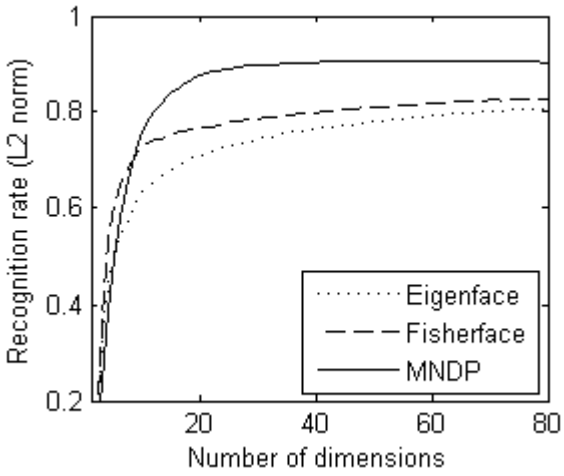
Table 5. 13 Top average recognition rate (%) on AR database

Method	Distance measures		
	L_1	L_2	Cosine
Eigenfaces	92.08 ± 1.23(80)	80.92 ± 1.23(80)	85.48 ± 1.23(80)
Fisherfaces	84.61 ± 1.32(79)	82.74 ± 1.21(80)	87.59 ± 1.28(55)
MNDP	**95.14** ± **1.12(40)**	**90.81** ± **0.81(41)**	**95.57** ± **0.81(40)**

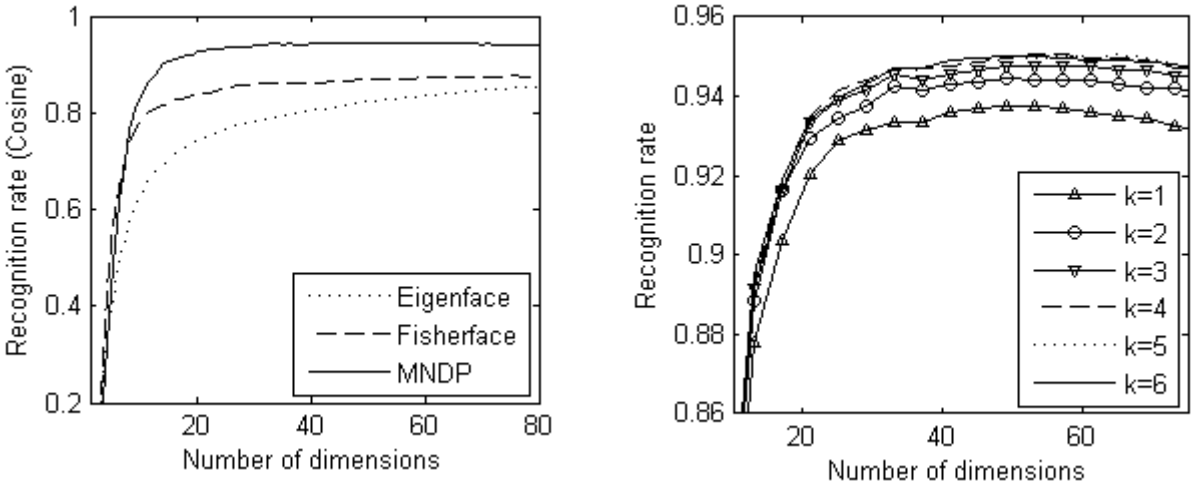
In Table 5.132, we observe that first the proposed MNDP method outperforms significantly the Eigenfaces and the Fisherfaces methods using three types of distance measures. The accuracy improvement over the Eigenfaces is 3.06%, 9.89%, and 10.09%, while for Fisherfaces it is 10.53%, 8.07% and 7.98%. Second, our method achieves the highest recognition rate with the fewest number of features. Compared with the other two methods it requires only about a half of the number of the dimensions (40/80) in achieving the top accuracy. Third, as shown in TableIII the standard deviation of the accuracy rate of MNDP is 1.12, 0.84 and 0.81 respectively. They are smaller than those of the two other methods, indicating that the performance of MNDP is more stable than those of the Fisherfaces and Eigenfaces methods in achieving the high recognition rate accuracy when the number of the features used for classification changes. In Figure 5.19 we show the statistics on the average recognition rate of the three methods,



(a)



(b)



(c) (d)

Figure 5. 19 Statistics on the recognition rate of MNDP, Eigenfaces and Fisherfaces. (a)-(c) Recognition rates with the L_1 norm, L_2 norm and the Cosine similarity. (d) Recognition rates of MNDP with varying size of the neighborhood k when the Cosine similarity is used for classification.

From Figure 5.19 (a), (b) and (c), we can see that, averagely, our proposed MNDP method shows better performance than the other two algorithms in the twenty folds tests. Especially, it outperforms the popular Fisherfaces method significantly. The reason is that face recognition is typically a small sample size problem. It is difficult to make accurate estimations on the density parameters using only a limited number of training samples. Our MNDP method can overcome this problem by dropping the assumption on the data density. It utilizes the boundary instead of the density information of data for dimension reduction, which is more suitable for small sample sized problems. In Figure 5.19 (d) we show the accuracy rate of MNDP changing over the neighborhood size k when using the Cosine similarity measure for classification. We can see that when the neighborhood size is set to be 1 the recognition rate is still low. As we increase the neighborhood size from 1 to 2 and 2 to 3 the accuracy rates jump from 93% to 94%, and 94% to 95%. However, when we further increase the size of the neighborhood to include more samples for training the recognition accuracy will not change too much any more. This indicates that there may exist a minimal neighborhood size k (in our case, $k = 3$). It determines a smallest set of the training sample vectors that are most effective in describing the class information of the data. Increasing the size of the neighborhood can only bring into the training process the samples that are far away from the class boundaries. These sample vectors however provide no further useful information to help improve the accuracy of the algorithm.

MNDP for Face Recognition:Yale face database

The Yale face database [5.87] contains 165 images of 15 individuals, each subject has 11 images of the size 100×800, manually cropped and resized to 50×40. Here, five tests are performed using different number of samples for training. More specifically, in the k-th test, we used the first k image samples per class for training and the remaining samples

for testing. The mutual neighborhood size is set to be 1. The Eigenfaces [5.72, 5.73] and the Fisherfaces methods [5.60] are implemented for performance comparison. The top recognition rate for each testing and the number of the projection vectors used for feature extraction are listed in Table 5.14.

Table 5. 14 Top recognition rate (%) and number of components used

Method	Number of training samples of each class				
	2	3	4	5	6
Eigenfaces	92.6 (28)	92.5 (32)	95.2 (34)	92.2 (72)	90.7 (24)
Fisherfaces	91.1 (11)	89.2 (10)	94.3 (12)	95.6 (8)	94.7 (10)
MNDP	**92.9 (15)**	**92.7 (11)**	**98.1 (12)**	**97.8 (7)**	**97.3 (10)**

We use the cosine similarity measure for classification. In Table IV it can be seen that in all the five tests the proposed MNDP algorithm achieves the highest recognition rates among the three algorithms. Specifically, compared with the Fisherfaces the MNDP method improves the recognition accuracy by 1.8%, 3.5%, 3.8%, 2.2% and 2.6%, respectively. It also outperforms the Eigenfaces method significantly with the increase of accuracy 2.9%, 5.6% and 6.6% when 4, 5 and 6 samples are used for training. Figure 5.20 shows the average recognition accuracy of the three algorithms changing over the number of dimension of the projection vectors.

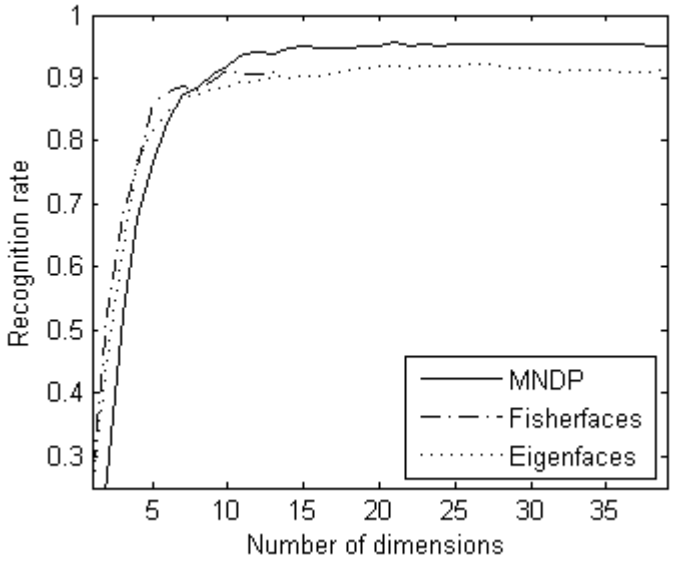


Figure 5. 20 Average accuracies of MNDP, Fisherfaces and Eigenfaces changing over the number of feature dimensions.

We can see that the accuracy rate of the MNDP method keeps the highest when more than 10 components are utilized for classification. Unlike the Fisherfaces it can obtain more than $C-1$ discriminant components for classification, where C is the number of the data classes. So, more projection vectors can be used to achieve better accuracy of classification. We also show the performance of MNDP when different types of similariy

measures, the L_1 the L_2 norms and the Cosine similarity measures are employed for classification. The result is shown in Figure 5.21.

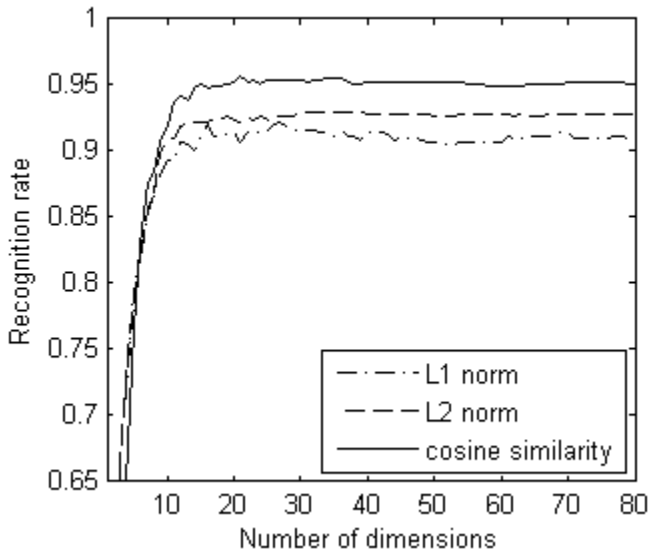


Figure 5. 21 Average accuracy of MNDP using L_1 , L_2, and Cosine similarity measures

We can see that the Cosine similarity measure generates significantly better results than the L_1 and the L_2 norms on the Yale face database, while the L_2 norm is consistently better than the L_1 norm. We conclude that the cosine similarity is more preferable in MNDP than the L_1 and L_2 norms. Figure 5.22 shows the recognition rate when the neighborhood size is set to be 1, 3, and 5 respectively. Six samples are drawn from each class for training.

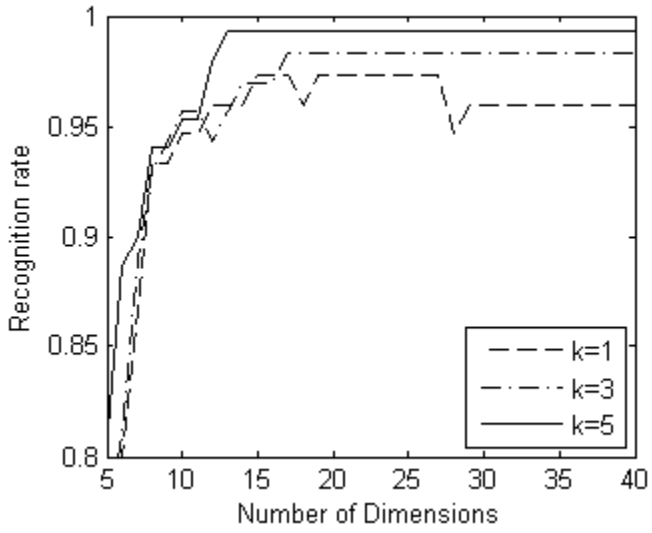


Figure 5. 22. Recognition rate of MNDP on Yale database with varying size of neighborhood

It can be seen that as we increase the size of the neighborhood from 1 to 5, more samples lying close to the boundaries of the data classes will be selected for training. The recognition rate of MNDP is also improved.

Kernel MNDP for Handwritten Digit Recognition: multi-class digit recognition

In this section, we use the U.S. Postal Service handwritten numeral database to analyze the performance of the Kernel MNDP algorithm. The LDA, the linear MNDP, and the KFDA methods are implemented for comparison.

The USPS handwritten digit database contains 7291 training observations and 2007 test observations of the digits 0 to 9. Each digit is represented as the grey scale image of 16×16 dimensions. In our experiment, all the images are converted to the one dimensional vectors and normalized to have the unit length. We design two experiments on the USPS database. In the first experiment, we perform the ten fold cross validations to evaluate the standard multi-class KMNDP method for digit recognition. In the second experiment, we test the two-class KMNDP for digit pair separation. Eight pairs of digit are selected for training and testing. The Kernel MNDP and the linear MNDP are compared with the LDA and the kernel FDA methods that have been previously used for handwritten digit recognition [5.92].

We perform the ten fold cross validation for testing. For each fold of testing, 100 samples are drawn randomly from each class to create the training set. So, we have totally 1,000 samples for training. The testing set that contains 2007 samples are used to compute the recognition accuracy. In our experiment the Cosine similarity is used for classification. The neighborhood size k is set to be 10. The average top recognition rate, the standard deviation of the accuracy rate and the number of the features used for classification are listed in Table V (with Gaussian RBF kernel).

Table 5. 15 Average top recognition rate with Gaussian kernel function

Accuracy	Method			
	KFDA	KMNDP	LDA	MNDP
equ. Num. of features	90.8 ± 0.71 (9)	**92.4 ± 0.60 (9)**	86.4 ± 0.75(9)	87.5 ± 0.66 (9)
opt. num. of features	90.8 ± 0.71 (9)	**94.1 ± 0.28 (11)**	86.4 ± 0.75(9)	89.7 ± 0.27 (10)

In Table 5.15, the first row shows the average top recognition rate of the ten fold cross validation with equal number of features being used for classification. We can see that KMNDP achieves the top accuracy rate among the four algorithms. The linear MNDP algorithm outperforms the LDA as well. The second row of the table shows the optimal number of dimensions used for each method in achieving the top accuracy. For KFDA and LDA, there are at most 9 components available for feature extraction due to the rank limit of the between class scatter matrix, whereas for KMNDP and MNDP there are no such restriction and we can obtain more features to achieve better classification accuracy. Figure 5.23 shows the average accuracy of the algorithms changing over the number of the feature dimension.

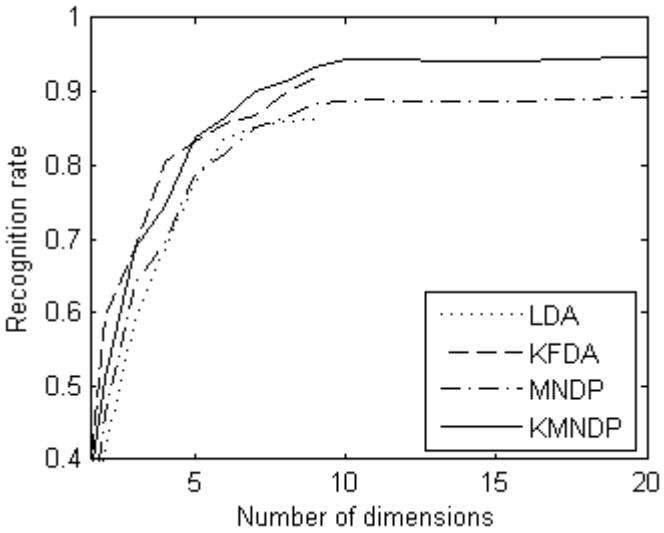


Figure 5. 23 Average accuracies of KFDA, KMNDP, LDA and MNDP changing over the number of feature dimensions

We can see that the KMNDP (with Gaussian RBF kernel) outperforms the other three methods consistently. However, the linear MNDP while being more accurate than LDA is not as competitive as the KFDA method on the USPS data set. This suggests that the handwritten digit dataset has strong nonlinearity that cannot be easily handled through using the linear projections. The kernel MNDP and the kernel FDA by taking advantage of the kernel method can address this deficiency by taking the nonlinear projection in separating the classes.

Kernel MNDP for Handwritten Digit Recognition: two-class digit recognition

In the second experiment, we test the two-class KMNDP for the recognition of digit pairs. Eight digit pairs, i.e., {0, 1}, {1, 8}, {2, 0}, {3, 4}, {4, 5}, {1, 5}, {6, 7} and {8, 6} are selected for training and testing. For each digit 100 samples are chosen randomly for training, while the samples in the testing set are used for validation. In our experiment, we set the mutual neighborhood size to be 15. The cosine similarity measure is used for the nearest neighbor classification. The top recognition rates of LDA, MNDP, KFDA and KMNDP are shown in Table 5.16.

Table 5. 16 Top recognition rate of KMNDP, KFDA, MNDP and LDA.

	Method					
Digit-pairs	KMNDP	$KMNDP_{opt}$	KFDA	MNDP	$MNDP_{opt}$	LDA
{0, 1}	99.2	**99.2 (1)**	99.2	85.2	95.3 (2)	83.9
{1, 8}	88.7	**99.1 (2)**	87.9	85.6	92.5 (2)	84.2
{2, 0}	97.3	**97.3 (1)**	96.2	93.4	93.4 (1)	92.6
{3, 4}	98.3	**98.3 (1)**	97.8	97.5	97.5 (1)	96.7
{4, 5}	98.2	**98.2 (1)**	96.9	84.4	90.5 (2)	83.1
{1, 5}	89.8	**98.1 (2)**	88.1	82.7	85.4 (2)	82.7
{6, 7}	98.7	**98.7 (1)**	97.5	94.7	94.7 (1)	93.1
{8, 6}	95.4	**97.7 (2)**	94.6	89.4	92.8 (2)	88.2

Average	95.8 (1)	**98.5 (1.4)**	94.4 (1)	89.9 (1)	93.6 (1.6)	88.1 (1)

When only one dimension of component is used, as shown in the columns 1, 3, 4, and 6 in Table 5.15, the average accuracy of KMNDP is 95.8%, higher than that of the KFDA 94.4%, while for linear MNDP it is 89.9% higher than that of the LDA, 88.1%. If we use the optimal number of features for classification, as shown in the columns 2 and 5, for the digit pairs {1, 8}, {1, 5} and {8, 6} the recognition rate can be further improved. It takes averagely 1.4 and 1.6 dimensions in achieving the accuracy rate of 98.5% and 93.6% for KMNDP and linear MNDP, significantly better than that of KFDA and LDA.

Now, taking digit pair {1, 5} as an example, we plot the scatter of 424 testing samples (264 samples from digit 1 and 160 samples from digit 5) as they are projected onto the discriminant space, as shown in Figure 5.24. For LDA and KFDA, since there is only one discriminant axis, the projected samples scatter on a line. Differently, the proposed MNDP and Kernel MNDP methods enable the data to scatter on a plane.

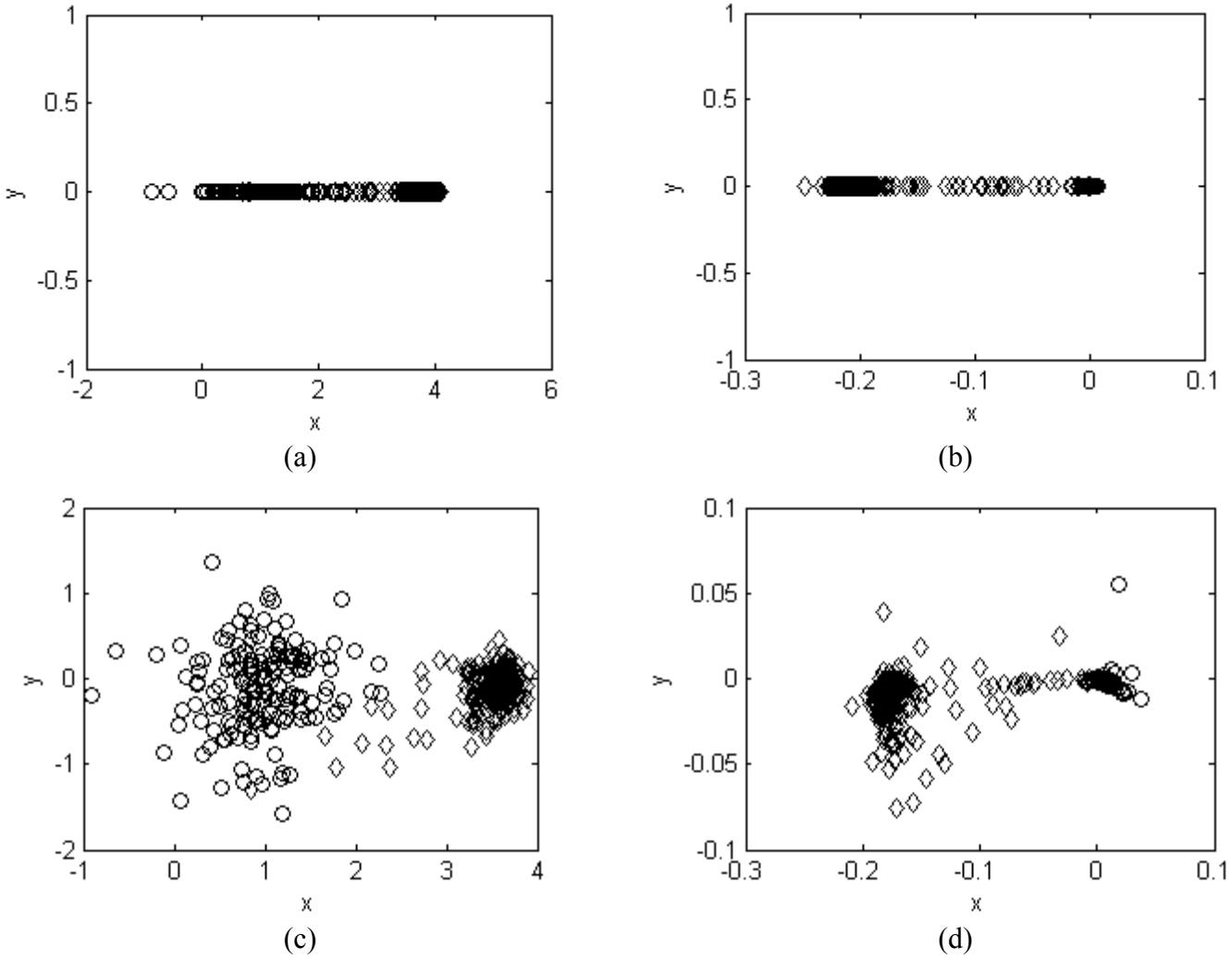


Figure 5. 24 The scatter plots of two class samples, digit 1 and digit 5 using LDA, KFDA, MNDP and KMNDP. (a) The scatter plot of LDA, where 73 out of 424 samples are misclassified. (b) The scatter plot of KFDA, where 51 samples are misclassified. (c) The scatter plot of MNDP, where 62 samples are misclassified. (d) The scatter plot of KMNDP, where 8 samples are misclassified

It can obviously be seen from Figure 5.24 that the data are more separable in the two dimensional KMNDP space than in the one dimensional LDA, KFDA and the two dimensional MNDP spaces. Actually, this separability can be simply measured by the classification errors. There are only 8 errors for KMNDP while 51, 73, and 62 samples are misclassified in KFDA, LDA, and MNDP. We also observe that compared with the linear MNDP projection the scatter of the kernel MNDP features are more tightly clustered by using the nonlinear projections. The separability of the two classes can thus be increased to reduce the error of classification.

5.3.5 Conclusion

In this chapter, we developed a new method called mutual neighborhood based discriminant projection for dimension reduction (MNDP). The proposed method has the following advantages over the existing ones. First, it is nonparametric in nature. By dropping the assumption on the data density, MNDP is reliable and suitable to be used to handle the data that are not necessarily Gaussian. Second, unlike LDA, the number of the projection vectors that can be obtained in MNDP is not restricted by the number of the data classes. More projection vectors are available for use to achieve better classification accuracy. Third, compared with the other nonparametric methods such as NDA, LPP and DLPP, there is only one parameter in MNDP, i.e., the neighborhood size k to be configured. It is convinenet for users to choose the proper parameters for model selection. Finally, we generalize the linear MNDP method for nonlinear dimension reduction by utilizing the kernel based method. It is shown to be more effective than the linear algorithm in processing the nonlinear patterns in the dataset.

Extensive experiments have been performed on a variety of benchmark databases for face and handwritten digit recognition. The experiment results on the AR and the Yale face database show that the linear MNDP method is more accurate than the popular Eigenfaces and the Fisherfaces methods. Meanwhile, it requires fewer discriminant components for the pattern matching and retrieval. For the handwritten digit recognition problem the kernel MNDP outperforms the KFDA, the LDA and the linear MNDP algorithms consistently and significantly. The effectiveness of the linear MNDP and the kernel MNDP methods has been demonstrated in our experiments.

It should be noted that the kernel MNDP algorithm, like other kernel based methods such as KPCA and KFDA, has high computational complexity when the size of the training data scales up. To reduce the computational complexity Mitra et al. [5.78] have developed a density based technique for multi-scale data condensation. By choosing only a subset of the data for training based on the kNN density estimation the computation and the memory efficiency can be improved. But their algorithm is designed mainly for

unsupervised algorithms where the class information is not utilized. The boundary information of the data classes may be lost during data reduction. It is an interesting topic for future study if we can integrate our mutual neighborhood based method with the density based data condensation technique for data reduction, so that the efficiency of the kernel MNDP algorithm can be improved.

Chapter 6. Adaptive CS4 algorithm

DNA methylation is the chemical modification of DNA that can be inherited without changing the DNA sequence. Methylation occurring at gene promoter CpG sites is one of the mechanisms for regulating the transcription of genes in human embryogenesis and tumourgenesis. Recently, it was discovered that a selected subset of 49 promoter CpG sites drawn from 40 cancer-related genes can be utilized as the biomarkers to differentiate human embryonic cells from cancer and normally differentiated cells. In this chapter, we perform a meta-level analysis on the methylation profile of three types of cells to address the two classification related questions: i) What is the minimal set of the CpG markers that we can learn to fully differentiate the three types of cells? Answers to this question can shed light into the critical subset of biomarkers useful to biologists; and ii) What is the specific pattern of gene co-methylation in human cancer and embryonic stem cells? Answers to this question can help biologists understand the nature of such cells.

In addressing the questions, we develop the Adaptive CS4 and clustering methods for rule-based cell classification and co-methylation analysis. By applying Adaptive CS4, we can reduce the number of CpG markers from 49 to only two without sacrificing the accuracy of classification. The expenses for lab array test can thus be reduced. The discovered methylation rules imply the logic of DNA methylation in programming the cells' development. On the other hand, by adaptive clustering we detect the patterns of gene co-methylation in human cancer and embryonic stem cells. The distinct patterns of co-methylation highlight the multifunctional and dominant role of DNA methylation in human embryogenesis and tumourgenesis. We show further that co-methylation depends on not only the locations of genes in the genome space but also the phenotypes of cells.

It is critical task to investigate the molecular mechanism underlying the DNA methylation rules. The clustering patterns of co-methylation identified the groups of genes orchestrated epigenetically for similar cell functions. Experiments in wet labs can

be designed to find out the co-methylation pathways in human cancer and embryonic stem cells.

6.1 Background

A genome is the program book of a life. Human genome consists of approximately 20,000~30,000 genes that encode the various types of proteins making up the human body. These genes are programmed to concert the production of proteins so that when given external stimuli, they can behave appropriately for function over time and space. One of the molecular mechanisms that are involved in such programming is DNA methylation [6.1]. Unlike gene mutations [6.2, 6.3], which regulate the transcription of genes by altering their coding sequence, DNA methylation defines the landscape of protein production by restricting the accessibility of the gene transcriptional factors to the transcription starting sites near the gene promoters. These promoter sequences, at the 5' end of the genes, are the regulatory elements for gene expression. Physically, the methylated CpG sites in the promoters will attract the methyl-binding domain proteins (MBD) such as the MBD1 to MBD4, and the MeCP1, MeCP2. The MBDs, in association with the DNA methyltransferases enzymes (DNMTs), will then recruit the histone deacetylases (HDAC) to deacetylate the histones by removing the electronegative acetyl groups from the nucleosomes [6.4]. As a result, the histone cores become positively charged so that the electronegative DNA strands will be tightly wrapped around them through the interaction of the static electromagnetic force. The transcription starting sites are thus made inaccessible to the RNA polymerase and other transcriptional factors, suppressing the transcription of the genes.

DNA methylation is considered to play important roles in the self-renewal and differentiation of human embryonic stem cells (hESC) [6.5]. When receiving the cellular signal inputs, they can transform to differentiate into any type of the adult cells with specialized functions and structures. This unlimited potency makes it an ideal solution to treat a host of the congenital, developmental, and degenerative diseases such as cancers, Parkinson's disease, spinal cord injuries and muscle damage. It is thus important for us to better understand the mechanism of DNA methylation to pave the way for the safe and effective use of hESC in medicine.

Over the past few years, the computational analysis of DNA methylation in human cells is becoming an increasingly important topic in the areas of statistical pattern recognition, data mining, and bioinformatics. These works are closely related to the classification and clustering of the methylation profiles of human cancer and normal tissue cells. Specifically, Fabian et al. [6.6] apply Support Vector Machines (SVMs) for tumour classifications using the Microarray-based methylation data. Das et al. [6.7] perform Linear Discriminant Analysis (LDA), Logistic Regression (LR) and use SVMs to predict the methylation status of the CpG sites, where SVMs achieve the best accuracy rate of classification. Bhasin et al. [6.8] also train SVMs to predict the state of CpG methylation at the genome-wide scale. Recently, Marjoram et al. [6.9] perform cluster analysis on the DNA methylation profiles of cancers. Some of the CpG sites are found to be co-

methylated. All the methods developed and used so far in DNA methylation analysis, including SVMs, LDA, and LR, are based on linear transformations.

In the previous works, the knowledge that we learn from these data is presented in the form of the weight coefficients that optimally define the decision functions. They are simply the numerical values without any semantic meanings, making it rather difficult for human experts to understand and interpret the result obtained in biological context. To address this deficiency, we propose a new algorithm, called Adaptive CS4 (ACS4) as a natural extension of the decision tree algorithm. Compared with the classical CS4 algorithm, the ACS4 can be utilized to discover and formulate the methylation rules in the form of human natural language. The knowledge learned on DNA methylation can thus be readily understood and interpreted by human experts. Moreover, by using the ACS4 algorithm we can select and use only 2 CpG biomarkers, instead of 49 markers [6.10], to predict accurately the class labels of the cells. The expenses for lab array tests and diagnosis can thus be reduced. This also allows biologists to work more efficiently by narrowing down their focus onto a much smaller set of biomarkers for analysis.

In addition, Marjoram et al. [6.9] recently study the clustering structure of the methylation profile in human tumour cells. Many CpG sites are found co-methylated. However, for hESC little is known about the clustering property of the gene CpG sites. To answer this question, we develop an adaptive clustering method and use it to find out the natural divisions of the CpG functional groups in the methylation profile of hESC. Our result shows that co-methylation exhibits different patterns for hESC and cancer cells. DNA methylation plays a multifunctional and dominant role in regulating the behaviour of these two types of cells.

6.2 Adaptive CS4 and clustering

6.2.1 ACS4 methylation rule discovery

The main idea of ACS4 is that instead of creating only a single decision tree, we can generate a committee of trees and use them for classification by voting, in a quite similar way as the boosted C4.5 algorithm [6.43]. The ACS4 can optimally find out the natural division of the attribute values and adaptively assign a set of the human linguistic variables with semantic meanings, such as ‘high’, ‘medium’, or ‘low’ to the attribute domain, so that the methylation rules in ACS4 can be learned and presented in human natural language for better utilization. To achieve this goal, we first cluster the methylation data to find out the optimal partitions of the attribute domains. We then sort the clusters according to their centres and assign the linguistic variables to the attributes for rule induction.

Given a training data set D having two classes of samples, *positive* and *negative*, the ACS4 algorithm consists of the following steps to derive iteratively k trees from D for classification.

Algorithm Adaptive CS4
***Input*:** Profile of DNA methylation

***Output*:** Association rules of methylation for classification
begin
for each CpG attribute in DNA methylation profile
Perform *c*-Means clustering with $c = 2$, 3, and 5 initial centres.
Take each cluster as one class and compute the Fisher's discriminant score, f_2, f_3, f_5, on the results of clustering.
Find out the clusters with the greatest Fisher's discriminant score, f_{k_max}.
Rank the clusters according to cluster centres.
Assign linguistic variables L_{k_max} to numerical domain of CpG attribute.
case *k_max* ***of***
2: L_{k_max} set to {'high', 'low'}
3: L_{k_max} set to {'high', 'medium', 'low'}
5: L_{k_max} set to {'very_high', 'high', 'medium', 'low', 'very_low'}
end case
end for
Compute the Shannon entropy and the information gain to rank all the features into an ordered list with the best feature at the first position.
for each feature
Use it as the root node to construct the decision tree.
Extract the DNA methylation rules by traversing the decision tree structure.
end for
end

The rules learned in ACS4 are more intuitive than those of the CS4 [6.44] in highlighting the semantics of the results obtained. For instance, one rule built in CS4 may take the form "IF the methylation level of CpG_1 ≥ 8280.25 AND the methylation level of CpG_2 ≤ 6821.75 THEN this sample is type_1". It however cannot tell us if the values '8280.25' and '6821.75' is either 'high', 'medium', or 'low' in the attribute domain of CpG_1 and CpG_2. The rules are not intuitive for human understanding and biological interpretation. In ACS4, instead, we can present the rules in human natural language to address this deficiency, hence, "IF the methylation level of CpG_1 is high AND the methylation level of CpG_2 is low THEN this sample is type_1". The utilization of the linguistic variables is based on the adaptive clustering of the attribute values, where the Fisher's discriminant scores are computed to evaluate the quality of the clustering results, so we can always choose the best one to partition the data most naturally.

6.2.2 Class prediction

By using the methylation rules, we are then able to make predictions on new testing patterns. Given a testing sample T, each of the k trees in the committee will have a specific rule to tell us its predicted class label. Denote the k rules in the tree committee as,

$$rule_1,\ rule_2,\ldots,\ rule_k. \tag{6.1}$$

For each $rule_i$ $(1 \le i \le k)$ we assign it a prediction value, p_i, hence,

$$p_i = \begin{cases} +1, & if\ rule_i\ predicts\ T\ to\ be\ in\ positve\ class, \\ -1, & otherwise. \end{cases} \tag{6.2}$$

We then evaluate the contribution of each rule to the final prediction. A weight, w_i, is assigned to the $rule_i$, according to its *Coverage* value, i.e.,

$$w_i = Coverage(rule_i). \tag{6.3}$$

The *Coverage* is the percentage of the samples in a class satisfying the rule. The greater the *Coverage*, the more reliable the rule can be used for prediction. The decision is made based on voting by computing the total score of prediction,

$$Score(T) = Sign\left(\sum_{i=1}^{k} p_i \cdot w_i\right) \tag{6.4}$$

where the signum function *Sign* extract the sign of the real number, it is defined as,

$$Sign(x) = \begin{cases} +1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0. \end{cases} \tag{6.5}$$

Thereby, if the prediction Score is +1, then the testing pattern is predicted to be in the positive class, if it is -1, then it is predicted to be in the negative class. We can also show how confident we are in making this prediction by Eq. 6.6.

$$Confidence(T) = 1 - \frac{min(\alpha^{pos}, \alpha^{neg})}{max(\alpha^{pos}, \alpha^{neg})} \tag{6.6}$$

where $\alpha^{pos} = \sum_{i=1}^{k_1} w_i^{pos}$ and $\alpha^{neg} = \sum_{i=1}^{k_2} w_i^{neg}$. w_i^{pos} and w_i^{neg} is the weight of the *i*-th rule making the positive and the negative prediction, respectively. k_1 and k_2 is the number of the rules of the positive and the negative predictions. Therefore, the larger the confidence value is the more confident we are in making the predictions. Specifically, if α^{pos} equals α^{neg}, which is the case for Sign being zero, the Confidence value of the prediction will be down to zero as well, meaning that more information has to be given to make the judgment on the class label of the testing pattern.

6.2.3 Adaptive clustering for co-methylation analysis

Suppose we have *m* CpG sites and each of them has *n* observations (sample cell lines), the adaptive clustering algorithm consists of the following steps.

Algorithm Adaptive Clustering
***Input*:** Profile of DNA Methylation
***Output*:** Clusters of gene promoter CpG sites
begin
Define a distance measure *d.*
for *c* from 2 to *N*
Perform *c*-Means clustering with *c* initial cluster centres.
Take each cluster as a single class. Compute the Fisher's discriminant score of clustering results.
end for
Find out the best result of clustering with the greatest Fisher's discriminant score.
Find out the CpG sites in the same clusters to examine the co-methylation patterns of genes.
end

We can define the distance measure *d* by using the Euclidean distance or the Pearson's correlation coefficients. The rules learned can guide biologist to design new experiments in wet-lab to find out exactly the molecular mechanism that underlies the epigenetic programming system.

Additional files:

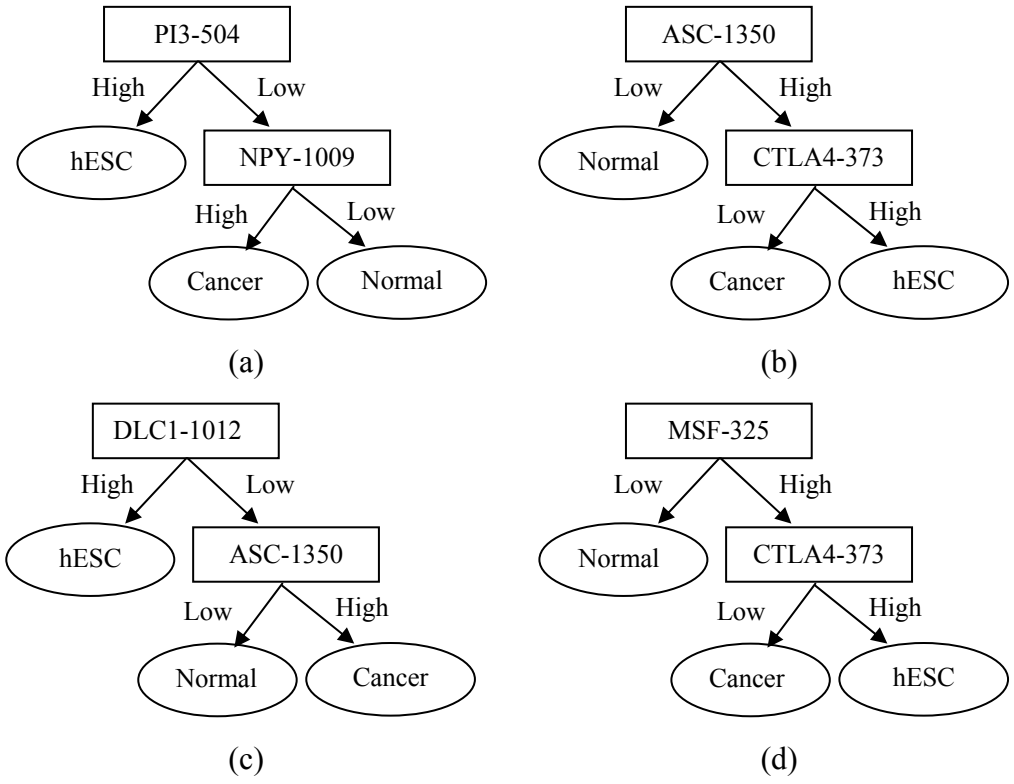


Figure 6. 1 Two committees of trees TC_1(a,b) and TC_2(c,d) (a) T_1 (b) T_2 (c) T_3 (d) T_4

Between Group Analysis (BGA):

The BGA method can be used for classification and gene selection. The idea is quite simple. For a data set with K samples, $D = \{x_1,...,x_K\}$ where $x_i \in R^N$, $i = 1,...,N$, the covariance matrix C of D which is of $N \times N$ dimensions is constructed and factorized by singular value decomposition, thus $C = U\Lambda V^T$, and U, V contain the *K-1* principal components that maximize the variations of the data. The most discriminating genes for classification are determined as those most influential to the variations of data. For the i^{th} gene, its significance is in proportional to the absolute value of the i^{th} element of the first principle component. By ranking the elements of the principle component, the genes that dominate the sample distributions and hence the prediction rates in the projected low dimension space can be identified.

6.3 Results

Dataset

The advance of the BeadArray-based technology has enabled the efficient high throughput profiling of DNA methylation at a genome-wide scale [6.11]. Recently, Bibikova et al. [6.10] performed an analysis on DNA methylation using the data acquired with this technology. They identified 49 CpG markers that can be used to classify the embryonic stem cells, the cancer and the normally differentiated cells. We show that with our proposed ACS4 algorithm the number of the biomarkers can be reduced to only two, without sacrificing the accuracy of classification.

The methylation profile used in our study consists of 37 hESC and 33 non-hESC sample cells lines (24 cancer and 9 normally differentiated), where 1,536 CpG sites in the 5' regulatory regions of 371 genes will be examined. These genes play important roles in cell proliferation, differentiation, apoptosis, DNA repair, oxidative metabolism, and other critical cellular activities. The profiling of the promoter CpG sites is based on the Qiagen DNeasy kit [6.12] for DNA extraction, the Bisulfite treatment for Cytosines-to-Uracils conversion [6.13], and the BeadArray-based GoldenGate assay [6.14] for the measurement of methylation intensities. Finally, we have 70 records with 1,536 numerical CpG attributes that range from 0 to 1.

ACS4 rule discovery

The proposed ACS4 algorithm differs from both the classical CS4 and the C4.5 methods. In CS4, the algorithm first ranks the data attributes according to their discriminative powers, then sets the leading attributes at the root nodes to grow a committee of trees. The C4.5 algorithm actually can be considered as a special case of CS4, where only the first decision tree in the committee will be trained and utilized for classification. Compared with CS4, ACS4 is different in that it can not only learn the association rules

for classification but also optimally assign a set of the linguistic variables to the numerical domains of attributes. Hence, the ACS4 rules can be presented in the form of human natural language to facilitate our understanding and interpretation on the results obtained.

In our experiments to learn the ACS4 rules, we partitioned the data set into two parts, i.e., the hESC (37 samples) and the non-hESC (33 samples). We first evaluated the performance of the algorithm through the 10-fold cross validation. We then picked out the most significant DNA methylation rules to make the biological interpretations. In the 10-fold cross validation, the data are split into 10 subsets for each category. Hence, in each round of test, nine subsets are used for training and the remaining one is utilized for testing. In Table 6.1, we present the rules discovered with the ACS4 algorithm in one of the 10 tests. The rules are all of high *Coverage* values and prediction accuracies, indicating that they are quite reliable in telling the difference between the hESC and the non-hESC with the methylation features of genes.

Table 6. 1 Significant rules of ACS4 with best *Coverage*

Tree number	Rule content	*Coverage* (%)
1	IF PI3-504 = high THEN hESC	100.0
	IF PI3-504 = low THEN non-hESC	100.0
2	IF CTSH-1148 = high THEN hESC	99.7
	IF CTSH-1148 = low THEN non-hESC	98.4
3	IF HLA-DRA-1353 = high THEN hESC	99.2
	IF HLA-DRA-1353 = low THEN non-hESC	98.7

The *Coverage* value here is the percentage of the samples in a class satisfying a rule. Suppose that a class consists of 100 positive samples and a rule is satisfied by 95 of them. The *Coverage* of this rule is then 95%. We also note that the non-hESC consists of the cancer and the normally differentiated cells. It is also convenient to find out the rules that can classify these two types of cells by applying the ACS4 algorithm. In Table 6.2, we show the most significant rule that is sufficient to achieve the full classification.

Table 6. 2 Significant rule distinguish cancer cells from normally differentiated cells

Tree number	Rule content	*Coverage* (%)
1	IF NPY-1009 = high THEN cancer	100.0
	IF NPY-1009 = low THEN normal	100.0

Finally, we are able to assemble the rules in both Tables 6.1 and 6.2 to derive the following decision tree to classify all the three types of cells.

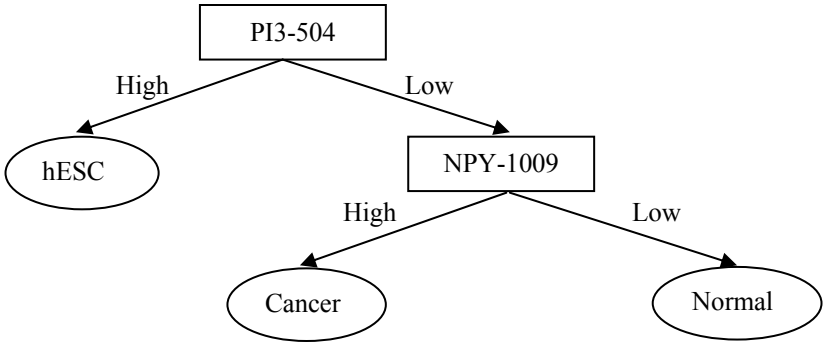


Figure 6. 2 Significant rules with two CpG sites fully separate three types of cells

Given a testing input data instance, we first measure its methylation status at the CpG sites PI3-504. If it is high at these sites, then the cell must be of the hESC type. Otherwise, we examine the CpG site NPY-1009. If it is hypermethylated, then it must be cancer type. Otherwise, it is a normally differentiated cell. In doing so, the three types of cells can be classified by using only 2 CpG markers, instead of 49 in [6.10].

There are two advantages in reducing the number of the CpG markers. First, as there are only two CpG sites to be examined, the expenses for lab array test can be reduced significantly. When the number of the sample cells is large, say, in the order of hundreds or thousands in scale, the reduction of the experiment costs will be quite remarkable. Second, as the marker set is narrowed down to only 2 CpG sites, biologists can analyze the methylation profile with less time and resource, while doing this without sacrificing the accuracy of prediction. These two advantages make our proposed ACS4 algorithm an effective method for DNA methylation analysis in hESC, cancer and normally differentiated cells. To better understand the ACS4 algorithm, in Figure 6.3, we illustrate the result of the clustering step in ACS4 on the attribute domain of CpG site PI3-504.

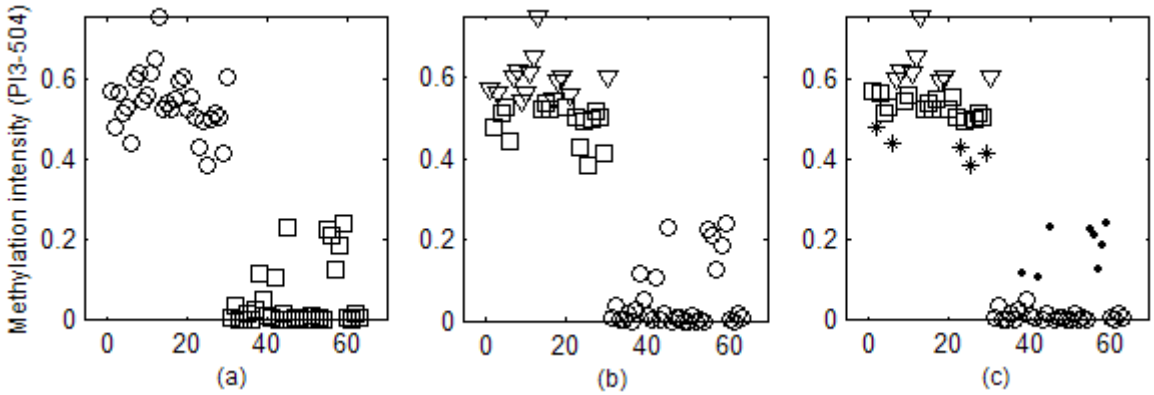


Figure 6. 3 Adaptive clustering of attribute domain of PI3-504. (a) Two clusters: fisher score 0.48. (b) Three clusters: fisher score 0.33. (c) Fiver clusters: fisher score 0.30.

The clustering is based on the *c*-means algorithm, where *c* is the number of clusters to be generated. For each observation of the attribute the *c*-means algorithm first finds its nearest neighbour in the *c* initial centres and makes a link to it. The initial centres are then

updated by taking the average of all the observations with links to them. These steps repeat until the initialized centres converge to the fixed points, forming the clusters that best describe the natural division of the attribute domain.

Figure 6.3 shows the identified clusters with the *c*-means method and the associated Fisher's discriminate score when the number of the initial cluster centres is set to 2, 3, and 5, respectively. The Fisher's score reaches to its maximum of 0.478 when the number of clusters is set to 2, corresponding to the optimal assignment of {'high', 'low'} to the attribute domain of PI3-504. For the other two results of clustering analysis, which have the Fisher's scores of 0.38 and 0.41, respectively, the results are suboptimal in the two-cluster case. Therefore, the methylation intensities of the CpG site PI3-504 take one of the two linguistic variables, either 'high' or 'low', in formulating the methylation rules.

Furthermore, let's examine closely the two CpG markers based on which the decision tree of Figure 6.2 is constructed. As illustrated in Figure 6.3, the horizontal and the vertical dimension represent the observation and the methylation intensity, respectively. Each symbol (circle or square) corresponds to one type of cell (hESC or non-hESC; cancer or normally differentiated cells). For the CpG site PI3-504, which is located at the root of the decision tree, the two clusters generated in Figure 6.3 fit very well to the two different types of cells. Therefore, if the methylation intensity is high, it must be hESC; otherwise, if it is low, it must be non-hESC. Similarly, for the CpG site NPY-1009, if the methylation intensity is high, it must be the cancer cells; otherwise, if it is low, it must be the normally differentiated cells.

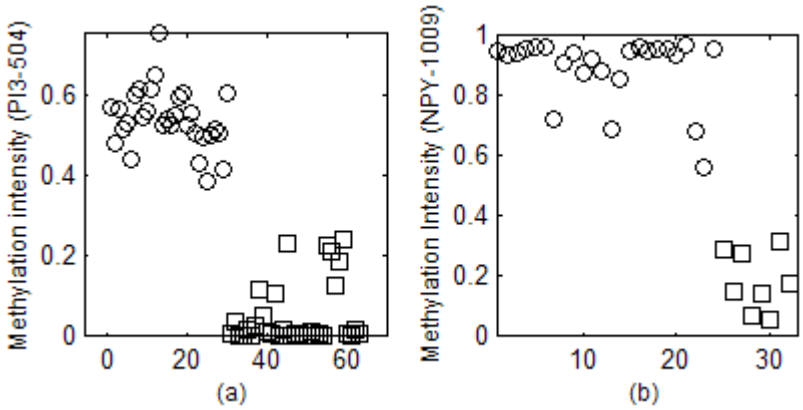


Figure 6. 4 Optimal clustering of attribute domain of CpG sites PI3-504 and NPY-1009 hESC (circle), non-hESC (square) (b) cancer (circle), normally differentiated (square).

When examining the results in biological contexts, the gene PI3 encodes the Phosphoinositide 3-kinases. This family of enzymes is recruited upon the growth factor receptor activation and produces 3' phosphoinositide lipids. The lipid products of PI3 act as the second messengers by binding to and activating diverse cellular target proteins, which ultimately mediates the cellular activities such as proliferation, differentiation, chemotaxis, survival, intracellular trafficking, and glucose homeostasis [6.15]. In hESC, the DNA is programmed in order to maintain a stable state of cellular environment, where the growth, the proliferation, and the differentiation of cells are negatively modulated. The methylation of the gene PI3 plays a critical role in down-regulating the

expression of the Phosphoinositide 3-kinases to fulfil such purpose. At the same time, the function of NPY (Neuropeptide Y) in regulating the energy balance of the cellular life is well known. Li et al. [6.16] demonstrate that the NPY expression level is low in patients with gastric and colorectal carcinomas. But the underlying mechanism is unclear. Our result indicates that DNA methylation of the gene NPY may be the leading cause of these two types of cancers.

But it is inadequate to justify the performance of ACS4 by examining the results from only 1 of the 10 tests. To determine whether the rules learned can work consistently across all the 10 folds, we need to show statistics on the tree models and the feature set. Assuming that the structures of the trees change from time to time across the 10 folds and involve even more features than 49, we can then hardly be able to apply them to achieve consistent predictions given new data. Conversely, if all the trees have the same structure, we can then be more confident that the trees capture the distribution of the dataset. It can work consistently and effectively for prediction and interpretation. To perform the statistics, we can count on the number of the distinct tree committees derived from the 10 folds data. If the leading tree members of a committee TC_i, trained in $fold_i$, share the same structure as those in TC_j built in $fold_j$, where $i, j = 1, \ldots, 10$, then TC_i and TC_j are identical, and for the common tree structure we have two counts.

In our test, we examine the rate of occurrence of the distinct tree committees by comparing the structures of the first two leading trees, which are most effective for prediction. Our result as indicated in Table 6.3 shows that there exist two distinct tree committees derived from the 10 folds training. While TC_1 includes Tree 1 and 2 (T_1 and T_2 for short), TC_2 has T_3 and T_4 (see appendix for the tree structures).

Table 6. 3 Consistency and accuracy of ACS4 across 10 fold CV

Tree Committee		Accuracy T_1 / T_3	Accuracy T_2 / T_4	Accuracy $T_{1,2} / T_{3,4}$	Occur. Rate
T_1	T_2	97.31 (0.12)	96.83 (0.06)	97.59 (0.07)	8/10
T_3	T_4	96.41 (0.06)	95.77 (0.08)	97.18 (0.14)	2/10

Table 6.3 indicates that TC_1 occurs most frequently, in 8 out of the 10 folds, as compared with the other tree committee, TC_2, which has only 2 occurrences. So, TC_1 dominates the prediction process, it is quite robust to the change of data.

In addition, as there are only two distinct tree committees, and each of them has only two member trees, the number of the distinct features involved is quite low (see appendix for features in TC_1 and TC_2). The average accuracy and the standard deviation (in bracket) of ACS4 are also listed in the table. There are three columns on the accuracy performance of ACS4. The first and the second column are the predication rates when the first and the second tree member is used for prediction separately, the third one is the accuracy when the whole committees of trees are utilized. We can observe that using more trees can improve the prediction rates. The adaptive clustering of the attributes and the voting procedure contribute most likely to the reduction of the prediction errors.

By using only one tree, T_1 encompassing two features, we can achieve even quite satisfactory result for prediction. The trained tree structure is robust to the changes of the data, reaching 97.31 prediction rates on average. It can also reduce the number of the features from 49 to 2 without distorting the result of prediction and interpretation. What is more important than simply reducing the number of the features is that we can use ACS4 to find out the dependency relations (logical rules) among the features and get to know how the rules determine the fates of the various types of cells. These rules are valuable to biologists. They can help us to understand the logic program of DNA methylation in chromatin remodelling. By analyzing the methylation circuits we can gain more controls to regulate cells' development through the methylation/ demethylation of the corresponding CpG sites presented in the methylation rules.

Besides consistency, another critical concern arises from the generalability of the ACS4 method. Can it predict well across different datasets? To evaluate, we have chosen a wide range of medical data that are publicly available, including LEUKEMIA [6.17], ALL [6.18], SRBCT [6.19], LYMPHOMA [6.20], COLON [6.21], and PROSTATE [6.22]. The detail information about the data is shown in Table. 6.4.

Table 6. 4 Statistics of the datasets utilized for 10 fold CV

Dataset	*LEUK.*	*ALL*	*SRBCT*	*LYMP.*	*COLON*	*PROS.*
Num. Instances	72	248	83	62	62	102
Num. Genes	7129	12558	2308	4026	2000	6033
Num. Classes	2	6	4	3	2	2

We run 10 folds CV and compare different algorithms, including the lazy learning method (kNN), the function based method (SVMs), and the tree based method (CS4, ACS4) based on Weka platform [6.23]. Specifically, for kNN, k is set to 1, for SVM we use the linear kernel, and for CS4 and ACS4 the number of the tree members in each committee is initialized to 20. The average accuracies and the standard deviation of each method are presented in Table 6.5.

Table 6. 5 Generalability and accuracy of ACS4 across 10 fold CV

Dataset	kNN	SVMs	CS4	ACS4
LEUK.	97.15 (3.29)	**97.68 (2.73)**	97.32 (2.25)	97.57 (1.87)*
ALL	96.57 (1.86)	**97.26 (1.52)**	96.90 (1.48)	97.18 (1.05)*
SRBCT	98.91 (2.40)	**100.0 (0.00)**	98.55 (1.23)	**100.0 (0.00)**
LYMP.	98.82 (2.47)*	**99.83 (1.61)**	97.77 (2.59)	98.36 (1.86)
COLON	82.36 (6.48)	**89.08 (6.82)**	88.12 (4.20)	88.34 (3.77)*
PROS.	89.14 (4.73)*	**92.55 (3.98)**	87.95 (3.75)	88.21 (3.59)

As shown in Table 6.5, SVMs achieves the top performance across all the 6 datasets, the best prediction accuracies are highlighted in the bold font. ACS4 ranks in the first and the

second places (with asterisks) in 1 and 3 out of the 6 datasets respectively. We can see that SVMs outperforms ACS4 in accuracy, but it is unable to generate the human understandable knowledge to assist the experiment designs in wet lab.

Compared with CS4, ACS4 cannot only give the interpretations intuitive to our understanding but also achieves better result of prediction across all the 6 datasets. Moreover, by doing clustering adaptively in the attribute domain, ACS4 can readily adjust its tree structures to the variations of data distribution. CS4 however lacks these abilities. Its outputs are numerical, thus not as intuitive as those of ACS4 to ease human understanding. Further, we can see that the kNN method ranks in the second place on two datasets, but it has to rely on all of the attributes in the data set to predict accurately.

The ACS4 algorithm, hence, can work consistently across the 10 folds cross validations. Also, it has good generalability when applied to the datasets with different distributions.

Co-methylation analysis

Marjoram et al. [6.9] recently analyzed the methylation profile of the lung and the colorectal cancers. They found that some of the CpG sites are co-methylated and form clusters. However, for hESC little is known about its clustering property of the CpG sites. To examine closely the patterns of co-methylation in hESC, we propose an adaptive clustering technique. In using this method, we initialize 2 to 60 cluster centres to cluster the 1536 CpG sites. The Fisher's discriminant scores are computed to find out the best result of clustering. In Figure 6.5, we illustrate how the Fisher's discriminant score varies over the number of initial clusters.

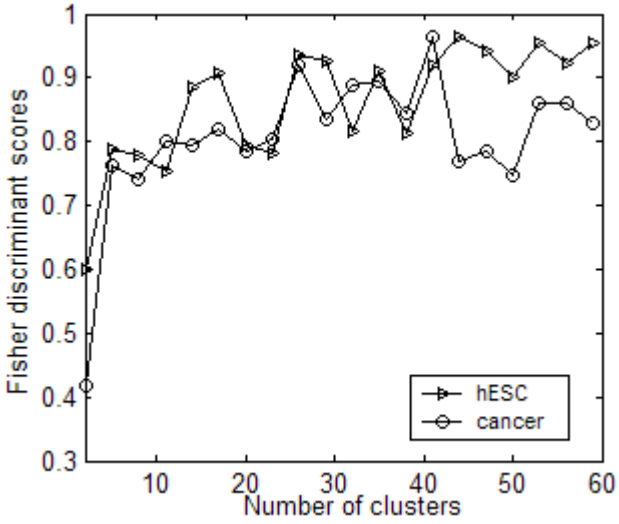
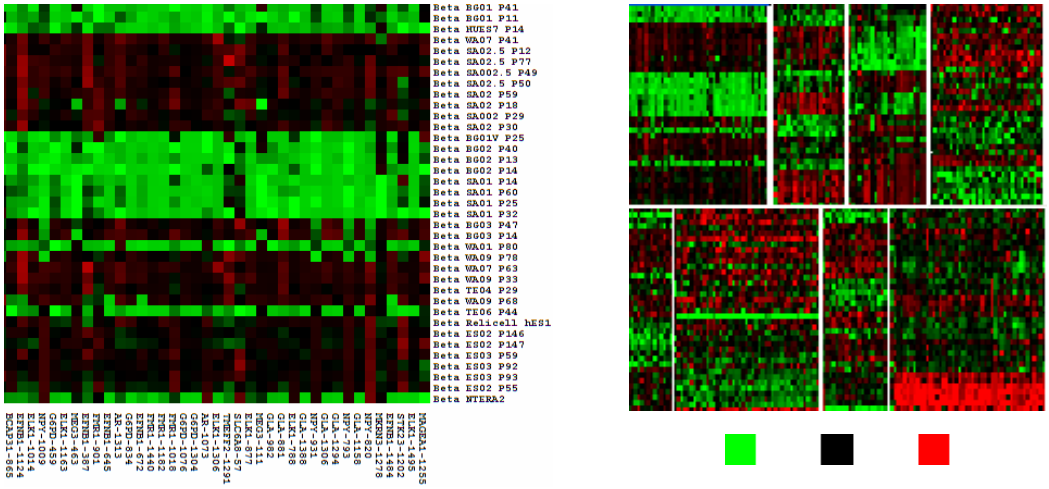


Figure 6. 5 Fisher discriminant scores vary over number of CpG clusters

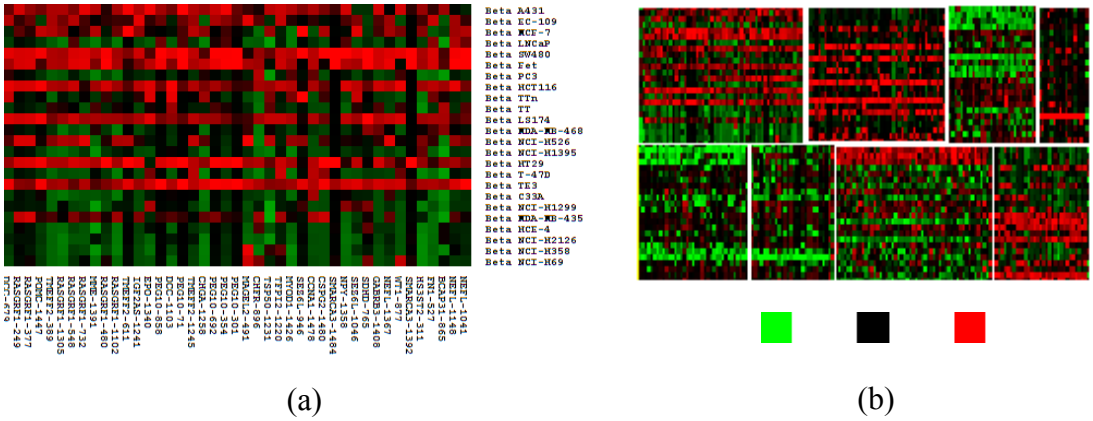
Initially, the Fisher's discriminant score is low, 0.6 and 0.4 only for the hESC and the cancer cells, when there are two clusters for each category. As we increase the number of clusters, the discriminant score gets improved gradually. However, it still fluctuates when searching for the optimal clusters that can best describe the natural division of the CpG

sample space. Finally, the best result of clustering is identified, where 59 and 41 clusters are generated for the hESC and the cancer cells, respectively. In Figs. 6.6 and 6.7, we show the clusters that exhibit strong patterns of co-methylation in hESC and cancer cells.



(a) (b)

Figure 6. 6 Clustering of hESC CpG sites. Green: negative, Black: zero, Red: positive. (a) Cluster of Genes on Chromosome X (b) Co-methylation in the first 8 clusters



(a) (b)

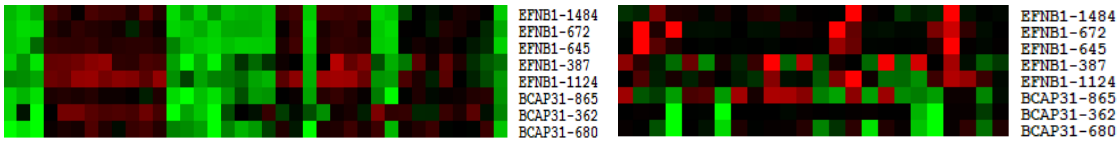
Figure 6. 7 Clustering of cancer cell CpG sites. Green: negative, Black: zero, Red: positive. (a) Cluster of genes contains RASGRF1 (b) Eight clusters exhibit strong patterns of co-methylation.

In Figure 6.6 (a), each row of the image corresponds to one sample cell line and each column represents a single CpG site. The image shows the patterns of strong co-methylations among the gene CpG sites. The full list of the names of the CpG sites and the clusters can be found in the supplemental files.

For each category of cells, we can observe that the genes are co-methylated and show the distinct patterns of clusters according to cell functions. For instance, in Figure 6.6 on hESC, we find that the genes in the same cluster, MAGEA1 [6.24], STK23 [6.25], EFNB1 [6.26], MKN3 [6.27], NPY [6.28], TMEFF2 [6.29], AR [6.30], FMR1 [6.31], are most related to differentiation, self-renewal, and migration of hESC, while in Figure 6.7

for cancer cells, the oncogenes, RASGRF1 [6.32], MYC [6.33], and CFTR [6.34] form the clusters with those highly involved in cell apoptosis (PLAGL1 [6.35], YWHAH [6.36], EP300 [6.37], BRCA1 [6.38], ERN1 [6.39]), alternative splicing of pre-mRNA (RPS26 [6.40]), DNA repair (BRCA1 [6.41], FEN1 [6.42]), tumour suppressing (TSSC4 [6.43], BCL6 [6.44], ST7 [6.45]), and ion transportation (MT1A [6.46], KCNK4 [6.47]), which characterize the immunological activities of cells in defending against DNA damages. Another example illustrating the function of co-methylation is that in cancer cells, the CFTR gene (7q31), a gene of great concern for two decades [6.34], is co-methylated with MT1A (16q13) and KCNK4 (11q13). The CFTR and KCNK4 proteins are the necessary building blocks in forming the ion channels across cell membranes, while MT1A proteins bind with the ions as the storage and transport tools. These three genes are in the same cluster. They are co-regulated epigenetically in modulating the intercellular traffics. The distinct patterns of the CpG clusters between the hESC and the cancer cells highlight the multifunctional and the dominant role of DNA methylation (demethylation) in human embryogenesis and tumourgenesis.

However, it is known that the adjacent genes in the genome space are functionally linked, and DNA methylation can spread across the DNA strand [6.48]. Our result of clustering confirms that the most highly co-methylated genes are near each other on the chromosomes. The location of the genes tends to have significant impact on the pattern of co-methylation. In Figure 6.8 (a), most of the genes highly co-methylated are located on the chromosome X. But this doesn't imply that co-methylation of genes depends simply on their locations. The epigenetic mechanism also works in a selective fashion according to cell functions. For example, we observe that the two neighbouring genes, EFNB1 (Xq12) [6.26] and BAP31 (Xq28) [6.49], both involved in the development of nervous system, located on the chromosome X, are co-methylated in hESC, but not in cancer cells, as shown in Figure 6.8. Besides the gene location there are other factors to be identified in determining which set of genes should be co-methylated, and which should not.



(a) (b)

Figure 6. 8 Co-methylation of gene EFNB1 and BCAP31 on Chromosome X. (a) EFNB1 and BCAP31 co-methylated in hESC (b) Not co-methylated in cancer cells

6.4 Discussions

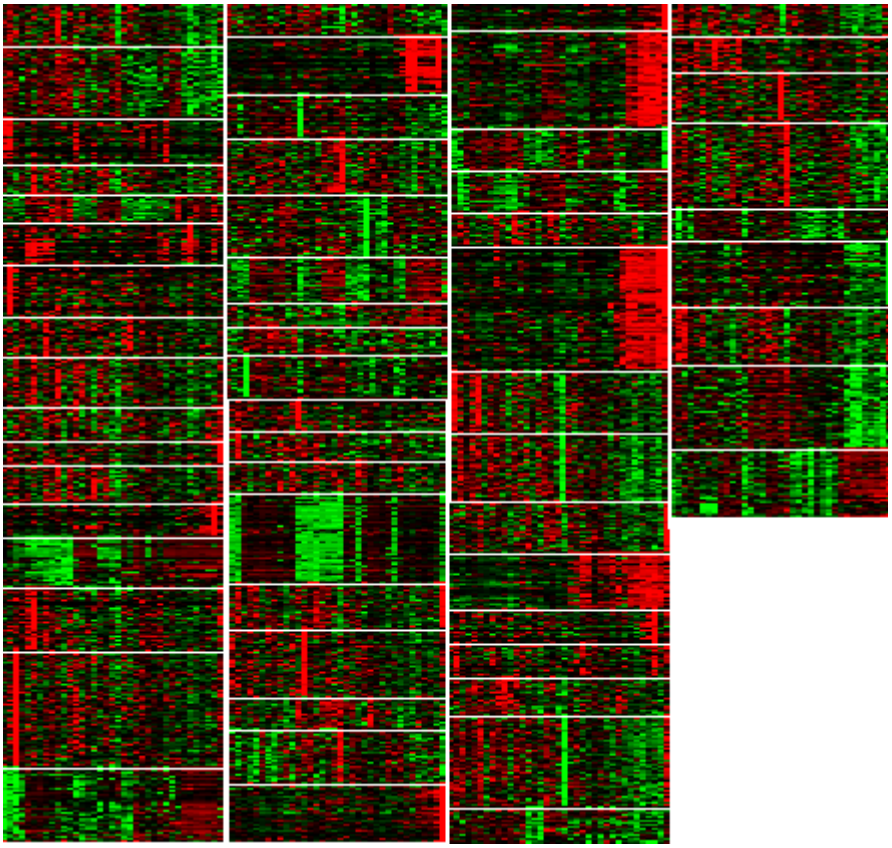
We have developed the ACS4 algorithm to discover the rules of DNA methylation in human embryogenesis and tumourgenesis. With ACS4, we can use only 2, instead of 49 CpG markers to predict accurately the class labels of the hESC, the cancer and the normally differentiated cells. The expenses for lab array test can thus be reduced significantly. Moreover, the learned ACS4 rules are formulated in human natural language. This makes it convenient for understanding and interpretation by human experts. Moreover, the distinct patterns of the CpG clusters, discovered by using our

adaptive clustering technique, highlight the multifunctional and the dominant role of DNA methylation in human embryogensis and tumourgenesis. We also confirm that the co-methylation status of the genes is largely dependent on their locations in the genome space, where adjacent genes tend to be highly co-methylated. Yet, we also find that co-methylation is a selective and function-oriented procedure. Two adjacent genes located on the same chromosme, highly co-methylated in hESC, are not co-methylated in cancer cells. The co-methylation status of genes depends not only on their locations, but also other factors to be identified in determining which set of genes should be co-methylated, and which ones should not.

Our proposed ACS4 algorithm and the adaptive clustering method are effective in detecting the patterns of DNA methylation in hESC. New evidence is presented to biologists as a potential guidance on new experiment designs in wet labs, to answer the following questions: i) How do we explain the logical structure of the decision tree in Figure 1 at the molecular level? What are the epigenetic pathways underlying such decision process? ii) How does the methylation mechanism decide which set of genes should be co-methylated, and which set of genes should not? How can we explain that two genes adjacent to each other are co-methylated in hESC, but not in cancer cells? iii) What is the relation between the epigenetic events (e.g., methylation, histone deacetylation) and the genetic events (e.g., gene mutation and transposition), and the role of DNA methylation in evolution? All these questions are to be answered to fully understand the epigenetic nature of cellular life.

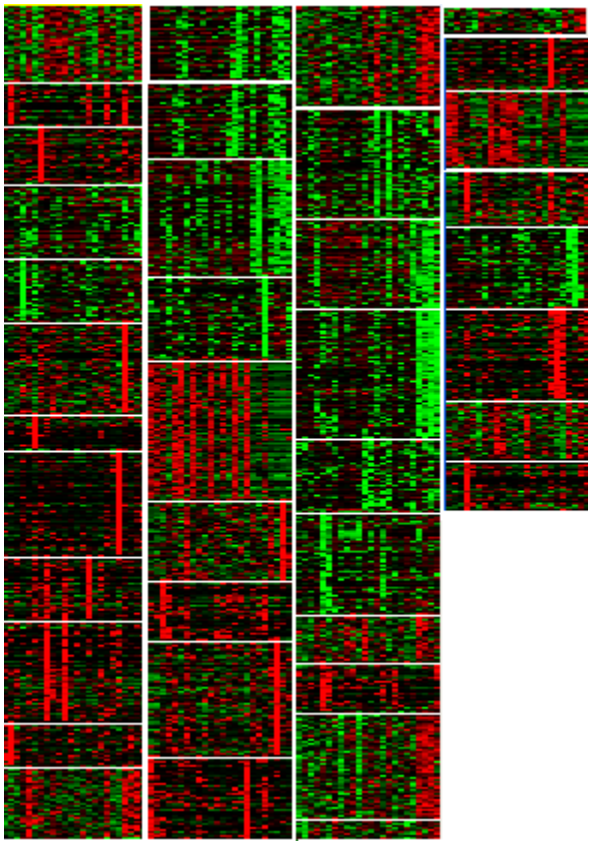
Additional file 1 – Name list of CpG clusters for hESC cells
http://www.cse.ust.hk/~csniuben/embryonic_stem_cell_cluster.txt

Additional file 2 – Array image of CpG clusters for hESC cells
http://www.cse.ust.hk/~csniuben/additional_image_stem_cell_CpG_clusters.bmp



Additional file 3 – Name list and Array image of CpG clusters for cancer cells:
http://www.cse.ust.hk/~csniuben/cancer_cell_cluster.txt

Additional file 4 – Array image of CpG clusters for cancer cells:
http://www.cse.ust.hk/~csniuben/additional_image_cancer_cell_CpG_clusters.



Chapter 7. Conclusions and future works

Advanced database applications, XML databases, Internet File Sharing, Biometrics and Bioinformatics, requires intelligent software to be developed for high-dimensional data indexing and searching. In this work, we presented 4 cmoputer algorithms to address the problem of dimensionality reduction. They are the two-dimensional Laplacianfaces, the Unsupervised Discriminant Projection, the Mutual Neighborhood based Discriminant Projection, and the Adaptive CS4 ensemble rules.

The 2D Laplacianfaces method is based on the two techniques, locality preserving and image-based projection. We improve the computational efficiency. The training time and the memory complexity is reduced from $O(m^2 \times n^2)$ to only $O(m \times n)$, where m and n are the number of rows and columns of the sample image. It is also more accurate than 2DPCA and 2DLDA by using the local information to help with recognition. In our experiment we also add a regulatory term to avoid the singularity problem in Fisherfaces, so, it is in fact regularized Fisherfaces in our implementations.

Unsupervised discriminant projection (UDP) considers not only the local but also the global information for optimization. It preserves the local density of the data while maximizing the non-local global scatter. UDP outperforms consistently the Locality Preserving Projections (LPP) and PCA, and outperforms LDA when the training sample size per class is relatively small.

We also developed Mutual Neighborhood based Discrimiant Projection (MNDP). As the errors of classification derive mainly from the samples on the class boundaries, we establish MNDP to best preserve the data geometry near the class boundary. We construct the mutual neighborhoods to highlight those samples on the boundaries which most likely contribute to the prediction errors. The extracted features from the facial and handwritten data indicate that MNDP is significantly and csonsistantly better than PCA and LDA. Moreover, the problem of singularity in LDA was successfully addressed.

Adaptive CS4 (ACS4) was designed for feature selection and prediction. It can be used to identify the most discriminant features and generates a committee of trees for prediction. We evaluate ACS4 on the biology database for DNA methylation analysis. The number of the index features for cell line classification is reduced significantly from 49 to only 2. The computational and the wet-lab costs for cancer diagnosis can thus be reduced by about 20 times in contrast to the previously reports. Meanwhile, we also propose a strategy for adaptive clustering on the gene methylation database. The results of clustering confirm that DNA methylation plays the dominant role in the process of tumourgenesis and embryogenesis in human cells.

These 4 methods are developed to deal with different kinds of problems. 2DLaplacianfaces, by employing image-based projection, is computationally more efficient than others. The traditional 1DLaplacianfaces method incurs high computation and storage costs on large images. 2DLaplacianfaces address this problem by reducing significantly the size of the matrix in the associated eigen problems.

Unsupervised Discriminant Projection is capable of feature extraction without using the class information. It transforms the data points to a low dimensional space where the neighboring samples remain still close and those faraway are mapped even farther. The local features help the classifiers, such as kNN, to retrieve data points more accurately by examining the local neighborhoods. The non-local features, on the other hand, help to achieve better separability of data class to reduce the Bayesian error.

MNDP is based on the geometry intuition that recognition errors derive mainly from the samples nearby the class boundaries. So, we focus on the sample pairs within and between the class boundaries in formulating MNDP. Also, MNDP can work effectively on the under-sampled dataset. In LDA, we can obtain at most N-1 projections on N class problem. But in MNDP the between-class scatter is constructed by using the boundary samples rather than the class mean vectors. Thereby, the rank of the between-class scatter is not restricted to N-1 anymore. We can have more useful projection vectors for feature extraction in MNDP.

The adaptive CS4 algorithm is developed to uncover the rules regulating the transcription of genes. The rules should be made human understandable. To achieve this goal, we perform adaptive clustering to discretize the numerical attributes into the human linguistic variables. The DNA methylation rules we discovered tell us the difference between the human embryonic stem cells, the cancer cells, and the somatic cells in the epigenetic context. They advance our knowledge about embryogensis and tumorgenesis. Moreover, by using ACS4 we have reduced the number of CpG markers from 39 to only 3 without degrading the prediction accuracy. ACS4 is a useful tool to help us understand the epigenetic events in cell development.

To summarize, these 4 methods are designed to solve different kinds of problems. 2DLaplacianfaces is computationally more efficient than others on large images. UDP is unsupervised, thus requiring no class information to be provided for feature extraction. It analyzes both the local and the non-local features of data to help with classification. MNDP focus on the boundary samples to improve the recognition rate. Also, we can use it to achieve more projections than LDA when the number of data classes is quite limited. With ACS4 we can uncover the DNA methylation rules that are human understandable. The rules can explain how the transcriptions of the genes are epigenetically regulated in cells' development, informative to the biologists for wet-lab design to discover the underlying pathways.

In ACS4, choosing the correct number of clusters is important. Clustering is performed for 2 purposes. First, it is used for discretization to declare the linguistic variables. With these variables, we are able to build the ACS4 rules in human natural language. These rules explain how the epigenetic mechanism regulates the transcription of genes in cells' development. We define five linguistic variables: "Very High", "High", "Medium","Low", "Very Low", in a same manner as in fuzzy logic and controls. We then create three sets of linguistic variables, {"High", "Low"}, { "High", "Medium", "Low"} and {"Very High", "High", "Medium", "Low", "Very Low"} for rule

construction. Given a numerical attribute, we cluster with 2, 3, and 5 initial centers, respectively. We then take each cluster as a class. The clustering result with the best fisher's discriminant score is the reference for discretization. We assign then the linguistic variables to the clusters. For example, if there are 2 clusters, we choose {"High", "Low"} while if there are 5 clusters, we use {"Very High", "High", "Medium", "Low", "Very Low"} instead.

The recognition rate can be related with the number of clusters. It is possible that the accuracy of classification can be improved if we increase the number of clusters. But in doing so the number of rules may also be increased, and some of them may be redundant due to the loss of generalability. There can be a tradeoff between specificity and generalability. We intend to achieve the satisfactory accuracy with as few rules as possible.

Also, by using genetic algorithms we may find out the best result of clustering without specifying the number of clusters beforehand. The adjacent matrix consists of 0-1 elements indicating whether 2 samples belong to the same cluster. The matrix can be encoded as a chromosome of binary string. For each chromosome/binary string we can compute its fisher's discriminant score to evaluate its fitness. All the adjacent matrices form a population of chromosomes. After a number of iterations of selection, reproduction, crossover, and mutation, the optimal/near-optimal solutions, the best result of clustering can be achieved.

Second, we perform clustering to analyze the epigenetic interactions among the gene promoters. The epigenetic process such as DNA methylation, histone modification and RNAi, dominates the process of gene transcription in cells' development. By clustering we can find out which groups of genes are functionally related for cell life. The patterns of co-methylation/co-regulation of genes can help us to detect the underlying pathways in tumorgenesis and embryogensis. To identify the gene clusters, we search exhaustively for the highest fisher's discriminant score. The computational cost in doing so is high. We can consider using heuristic based search to accelerate. For example, we can use the above mentioned GA method.

In MNDP, it does not perform very well for smaller dimensions. The possibility is that, First, the leading components of MNDP may be less energy compact than those of Fisherfaces. Since Fisher's criterion seeks to maximize the between-class variance its leading eigenvectors are quite capable of capturing the global pattern of the conditional densities. Thus, they provide a better accuracy than those of MNDP at the initial stage. But MNDP can examine more closely and make full use of the boundary information for classification. Its trailing components are more effective than those of Fisherfaces in characterizing the local density patterns near around each data point, thus, it can outperform Fisherfaces as more components are included. The distance metrics for classification affect the accuracy as well. When L1 and L2 norms are utilized for matching, we can observe the difference of accuracy between Fisherfaces and MNDP. But when we use the Cosine measure, this difference is quite small. Hence, if we choose different metrics the error rates can also be different. It is not impossible that MNDP can

outperform Fisherfaces with the leading vectors by using other distance metrics. Another issue is the dataset we use. We do experiments on the human facial image and handwritten digit databases and observe the above phenomenon. Whether or not this occurs on other types of data is to be investigated.

Using kernel functions may improve the robustness of the MNDP method. It is possible that two samples in a mutual neighborhood can be far away from each other as mutuality doesn't necessarily imply locality. In such special case a minority of the 'faraway neighbors', the outliers, can dominate the discriminant function, making the classifier too generic and be pruned to outliers. This problem hopefully can be solved by using the kernel functions. With the Gaussian kernel, we can measure the distance between 2 samples in the kernel induced space. Hence, the samples in the same mutual neighborhood but farway from each other will be assigned a small weight to alleviate its effect on the discriminant function. This can help to minimize the distortion of the local densities around the data points. Actually, Loog et al. have proposed a method [7.1] "Multiclass linear dimension reduction by weighted pairwise fisher criteria" based this idea. They weight the sample pairs based on their distances to handle the noises.

Also a question is why do the performances of 2D Laplacianfaces, Fisherfaces, and Eigenfaces deteriorate with dimension after 5-dimensions? I searched the literature and found 3 papers [7.2-7.4] describing this as the "peak phenomenon". Hughes [7.2] first confirmed the existence of the relationship between the dimensionality, the number of the training samples and the accuracy of recognition. He found that an optimal number of dimension exist and can lead to the best performance of recognition. With the number of training samples m fixed, the accuracy first begins to rise with the increase of dimension n, as shown in Figure 7.1. It must then fall back as $n/m \to \infty$ as the precision of the probability estimates monotonically degrades.

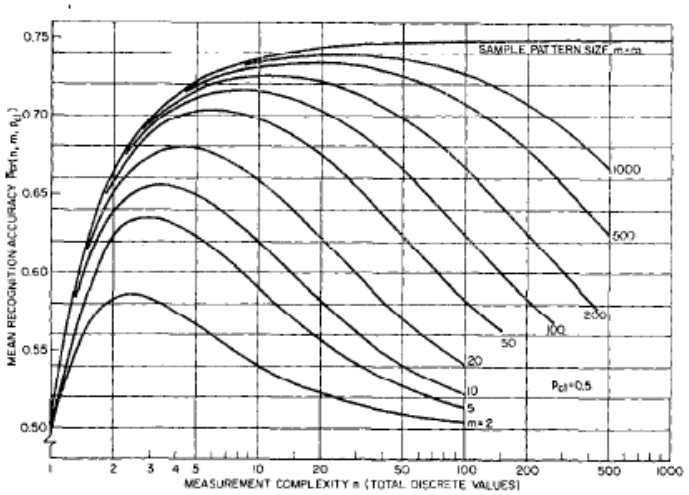


Figure 7. 1 Relation between *m*, *n* and recognition accuracy. (Taken from [7.2])

In Figure 7.1, horizontally it is the number of dimensions. Vertically, it is the recognition accuracy, the number of training samples ranges from 2 to 1000. Classification is based

on the Bayesian method. Also, Hamamoto et al. [7.3] showed that no increase in the generalization error of multilayer ANN classifiers is observed if the number of training samples increases linearly with the dimensionality. But for other methods, QDF, 1-NN and Parzen classifier, there is the increase of error when the number of training samples is fixed but the dimension increases, as shown in Figure 7.2.

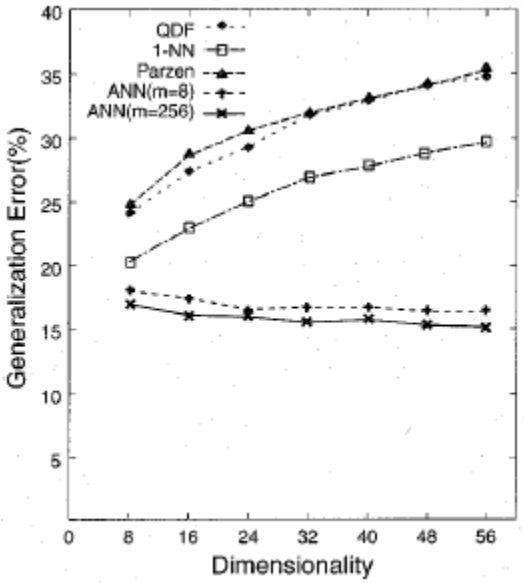


Figure 7. 2 Error and dimensionality for different types of classifiers (Taken from [7.3])

Figure 7.2. indicates that the generalization error of 1-NN, QDF, and Parzen classifiers increases after 8 dimensions. ANN turns out to be more robust than others with more neurons (m) being employed in the hidden layer. Hua et al. [7.4] also showed this peak phenomenon on QDA, as shown in Figure 7.3.

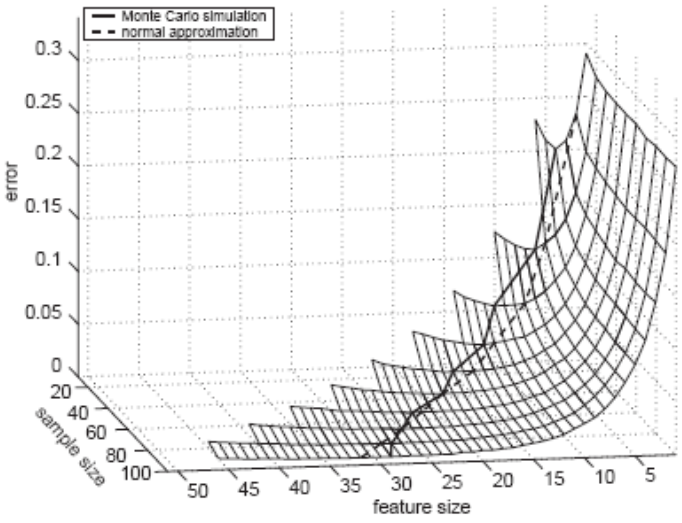


Figure 7. 3 Relation between generalization error, dimensionality and number of training samples for QDA classifier. (Taken from [7.4])

Figure 7.3. shows that when the sample size is fixed, the error rate grows along with feature size. When the feature size is fixed the error rates decreases if the sample size increases. So the error rate depends on both the number of dimensions and the sample sizes.

In the thesis, it might be a little bit misleading as we should have used “the number of components” instead of “the number of dimensions” in the Figure caption. In fact, a component of the 2D method is a multi-dimensional vector, while that of the 1-D method is a scalar. As each component in 2D version is multi-dimensional we can achieve the peak performance with fewer components, but not necessarily fewer dimensions. For clarity, we show the first 40 dimensions of the 1-D method. The curves arrive at their peak performance. They then stabilize and drop slightly as the dimension increases.

Chapter 8. Appendix – Database supports on Indexing: applications, computing architectures, and algorithms

Index structures help users to access data efficiently. We review in this Chapter the techniques for indexing in 2 types of information systems, Relational DBMS and Internet Information Retrieval. Methods for dimensionality reduction are essential to give support for such types of systems to achieve better performance in data indexing and retrieval. In Section 8.1 we discuss about RDBMS, its advanced applications, data models, computing architectures, and the indexing methods. In Section 8.2, we investigate the architecture of indexing and query processing regarding Information Retrieval. We study 2 of its major applications in Web Search Engine and Internet File Sharing.

8.1 Relational DBMS

8.1.1 Fundamentals of RDBMS

A database reflects the logic relations among the real world entities. It is populated with many data instances to be consumed by users. A database management system (DBMS) is a collection of the Computer softwares to facilitate the design, creation, modification, and communication of databases. To interpret the complexity of the real world various types of data models have been developed. The hierarchical, the network, the relational, and the object oriented models are most popular, among which the relational model is in the dominant place due to its flexibility to handle the complex queries.

The relation model is based on the set-theoretic *relation*, which is defined as a subset of the cross product of a list of domains $D_1, D_2, ..., D_k$, i.e., $C = D_1 \times D_2 \times ... \times D_k$. In other words, it is the set of all k-tuples $v_1, v_2, ..., v_k$, where $v_1 \in D_1, v_2 \in D_2, ..., v_k \in D_k$. A *relation* is any subset of the cross product of one or more domains, i.e., $T \subseteq C$. In addition, a relation scheme R is a finite set of attributes $A_1, A_2, ..., A_k$, where for each of them A_i, i = 1,…,k, there is a corresponding domain D_i containing all its possible values.

To handle user queries the Relational Algebra, a set of operations on relations, was proposed by E.F. Codd in 1972, including mainly the operations *SELECT*, *PROJECT*, *PRODUCT*, *UNION*, *INTERSECT*, and *JOIN* .

SELECT: σ
Let T and R be the Relation and its scheme, respectively, A is an attribute, $A \in R$. Hence, $\sigma_{A=a}(T) = \{t \in T \mid t(A) = a\}$, where $t(A)$ denotes the value of attribute A of tuple t.

PROJECT: π
Let T and R be the Relation and its scheme, respectively, A is an attribute, $A \in R$.
Hence, $\pi_X(T) = \{t(A) \mid t \in T\}$, where $t(A)$ denotes the value of attribute A of tuple t

INTERSECT: $\cap$
Given the two relations T and S having the same scheme, $T \cap S = \{t \mid t \in T, T \notin S\}$.

UNION: $\cup$
Given the two relations T and S having the same scheme, $T \cup S = \{t \mid t \in T \text{ or } S\}$.

DIFFERENCE : $-$
Given the two relations T and S having the same scheme, $T - S = \{t \mid t \in T, t \notin S\}$.

PRODUCT: $\times$
Let T and S be two relations of arity k_1 and k_2 respectively, where arity is the cardinality of the relation scheme. $T \times S$ is the set of all $k_1 + k_2$ - tuples, whose first k_1 components form a tuple in T and whose last k_2 components form a tuple in S.

JOIN: Π
Let T be a relation with the attributes A, B and C. Let S be a table with the attributes C, D and E. $T \Pi S = \pi_{T.A,T.B,T.C,S.D,S.E}(\sigma_{T.C=S.C}(T \times S))$.

The operations retrieve data based on the primary-foreign keys defined in the relation model. A primary key is defined as an attribute or a set of attributes uniquely identifying a specific tuple, while the foreign key establishes the logic relations among the entities in the model. There are three types of logic relations, many-to-many, one-to-many, and one-to-one.

To define the Many-to-Many relation, where many tuples from one relation can match many tuples in the other, a junction relation, whose primary key is composed of the 2 foreign keys from the two tables can be created. For the simpler yet more common case of one-to-many relation, it only adds the primary key of the relation on the "one" side to the "many" side as the foreign key for reference, the junction relation is not required. For the simplest one-to-one relation, one needn't to define the primary-foreign key relations because the data can be put into one table without information redundancy.

The operations for data modeling and manipulations are implemented by the Structured Query Language (SQL). The main SQL commands fall into three categories. The Data Definition Language (DDL) contains the commands used to create and modify the data model. After the data model is established database administrators and users can then use the Data Manipulation Language (DML) to insert, retrieve and change the data contained in relations. There are also other types of commands for data administration and transactional controls, as listed in Table 8.1.

Table 8.1 Core SQL commands

Data Definition Language	*CREATE TABLE*; *ALTER TABLE*; *DROP TABLE*; *CREATE INDEX*; *ALTER INDEX*; *DROP INDEX*; *CREATE VIEW*; *DROP VIEW*
Data Manipulation Language	*SELECT*; *INSERT*; *DELETE*; *UPDATE*
Administration and transactional control	*ALTER PASSWORD*; *GRANT*; *REVOKE*; *CREATE SYNONYM*; *START AUDIT*; *STOP AUDIT*

Recently, the Web Service Technology has aroused considerable interests as a protocal for Enterprise Application Integration (EAI) and Business to Business (B2B) computing. Large enterprises and business organizations usually maintain applications that are heterogeneous in languages, data structures, and platforms. This makes it difficult to exchange the data and services to share the resources. The vendor dependent SQL standard turns out to be inadequate to address this difficulty.

In the Client/Server model, web services expose the RPC-like programming interfaces to the application developers. For example, StockInfor is a web service that allows the programmers and financial analysts to retrieve the information about the stock names, the stock prices in an exchange center. It publishes the operations of query, insert, and delete to the users. Each operation can take input parameters and produces an output to the remote user end.

By web service, the client application doesn't have to be aware of the implementation details of the service providers, but only the specifications of the programming interfaces. Built on top of the eXtensible Markup Language (XML), the three fundamental technologies, Web Service Description Language (WSDL), Simple Object Access Protocol (SOAP), Universal Description Discovery and Integration (UDDI) provide the core functions of Web services. Their relationship is shown in Figure 8.1.

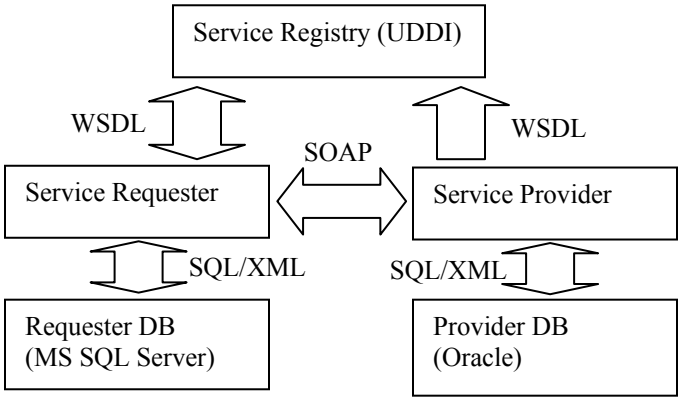


Figure 8.1 Architecture of Web services for Database applications.

In Figure 8.1, we demonstrate a typical scenario for database applications, where the service user wants to retrieve data from the service provider, and store the retrieved information into its own database for analysis. To get the data they needn't have to be aware of which type of RDBS is running at the remote end.

To achieve this goal, the service providers will use the WSDL protocol to describe the interfaces of their services and then publish them by registering at the UDDI servers. The users can then search the UDDI server to find out the services they want. By reading the WSDL document it can learn the interface of the remote function and make calls to the service provider with a SOAP request. The service providers in response search their local DB by the SQL/XML operations. The data are returned in XML format to the users in the SOAP protocal. The users can then perform the SQL/XML operations to check out the XML data and insert them into their own RDBMS.

The SQL/XML operations can convert the data from the XML format to the relation tables, and reverse. For instance, Oracle implement its API called XML SQL (XSU) to enable java to convert XML to SQL. Consider a simple example where we have the following relation table employee,

```
CREATE TABLE employee
(
  EMPNO NUMBER,
  ENAME VARCHAR2(20),
  JOB VARCHAR2(20),
  MGR  NUMBER,
  HIREDATE DATE,
  SAL NUMBER,
  DEPTNO NUMBER
);
```

After mapping the result of SQL query "select * from employee" to XML document, the XSU generates the following output.

```
<?xml version='1.0'?>
<ROWSET>
    <ROW num="1">
    <EMPNO>7369</EMPNO>
    <ENAME>Smith</ENAME>
    <JOB>CLERK</JOB>
    <MGR>7902</MGR>
    <HIREDATE>12/17/1980 0:0:0</HIREDATE>
    <SAL>800</SAL>
    <DEPTNO>20</DEPTNO>
    </ROW>
 <!-- additional rows ... -->
</ROWSET>
```

The tag *ROWSET* corresponds to the name of the relation table, each *ROW* element represents one tuple. The tag names of the *ROW* elements refer to the attributes of the relation scheme. The SQL-XML conversion is made by the OracleXMLQuery java class. Its object takes in the SQL query on a relation table as the input parameter, and returns the result directly in the format of XML document.

```
Import oracle.jdbc.driver.*;
import oracle.xml.sql.query.OracleXMLQuery;
import java.lang.*;
```

```
import java.sql.*;

class testXMLSQL {
  public static void main(String[] argv)
  {
    Connection conn  = getConnection(,);
    DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
    String username = "scott"; String password ="tiger";
    Connection conn =
    DriverManager.getConnection("jdbc:oracle:oci8:@",username,password);
    OracleXMLQuery qry = new OracleXMLQuery(conn, "select * from employee");
    String str = qry.getXMLString();
    System.out.println(" The XML output is:\n"+str);
    qry.close();
   }
}
```

The java statements as highlighted in the bold font illustrate the operations for SQL-XML mapping, where a new object of the java class OracleXMLQuery is initialized to convert the SQL query result to the XML format. Conversely, if we have the XML document in the first place, we can transform it to the RDBMS table with the following statements.

```
import oracle.xml.sql.dml.OracleXMLSave;
    ……
    OracleXMLSave sav = new OracleXMLSave(conn, "scott.employee");
    sav.insertXML(argv[0]);
    sav.close();
```

The XML-SQL mapping is implemented in the java class OracleXMLSave. Its instance can take the name of the XML document as the input parameter, then extract and insert the XML data into the RDBMS table. The Web services architecture, by taking advantage of the XML technology, enhances the interoperability of the RDBMS and relational data model, making them the highly effective solutions for real world applications.

To summarize, we present here an example to illustrate the logic design of RDBMS by using the Unified Modeling Language (UML). As shown in Figure 8.2, our sample database, SaleDB, contains four relation tables, Customer, Employee, Department, and Transaction. There are three relationships established among these four tables, as shown by the primary key (PK) - foreign key (FK) connections. A customer may ask for many Employees for service and one Employee can serve different customers, the relationship between the Customer and the Employee is thus many-to-many. Hence, a junction table, Transaction, containing the CustomerID and EmployeeID as foreign keys, is created to define this relation. On the other hand, the Department relation has the one-to-many relationship to Employee because one Department can have many employees, yet, one employee can belong to only one department.

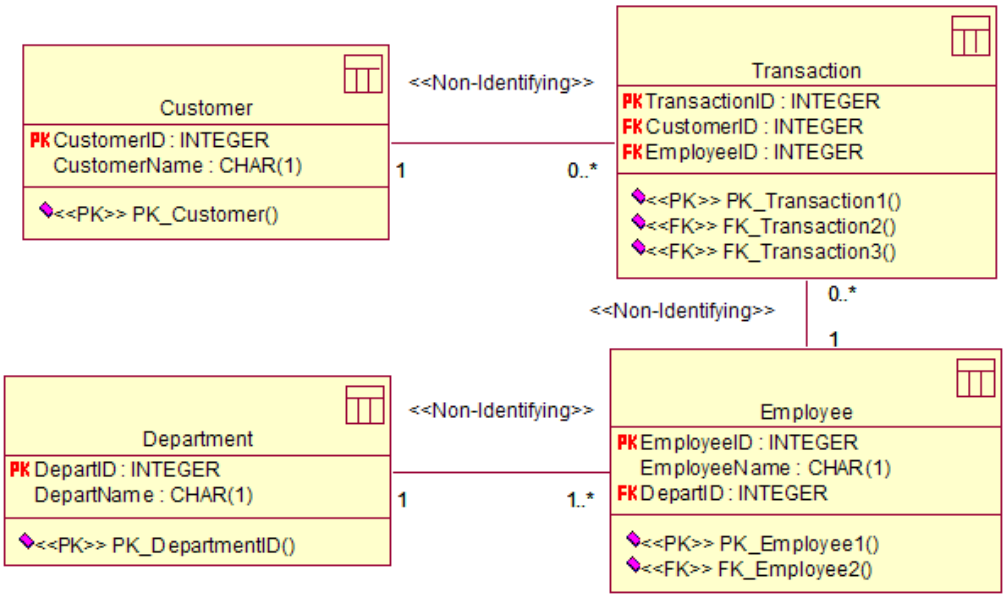


Figure 8.2 Relational data model diagram in UML

8.1.2 Advanced applications of RDBMS

Advanced applications of RDMS can involve the processing of data of high dimensionality. Real-time requirement can be the important indicator of the system performance. We discuss three major applications of database in bioinformatics, multimedia processing, and Geographic Information Retrieval.

Biological Databases.

Over the past few years, the advances of the high-throughput biotechnologies, e.g., micro-array, have made Bioinformatics an active research area. Bioinformatics addresses the needs of the modeling, storage, retrieval, and management of the biological information. Useful patterns discovered in the biological databases can highlight the connections between the genes and the human diseases. It can also help to find out the drug targets to design new drugs for treatments.

By 2006, there are over 1,000 public and commercial biological databases. Usually, these databases contain the genomics and proteomics data, some of them also maintain the information of protein-to-protein interactions and gene expressions. Basically, the data are kept in the form of genome sequence or the protein sequence of amino acids. Information about the sequences including their biological functions, locis, SNP patterns, and the three dimensional structures can be annoted for reference. We list in Table 8.2 some of the most highly cited biological databases.

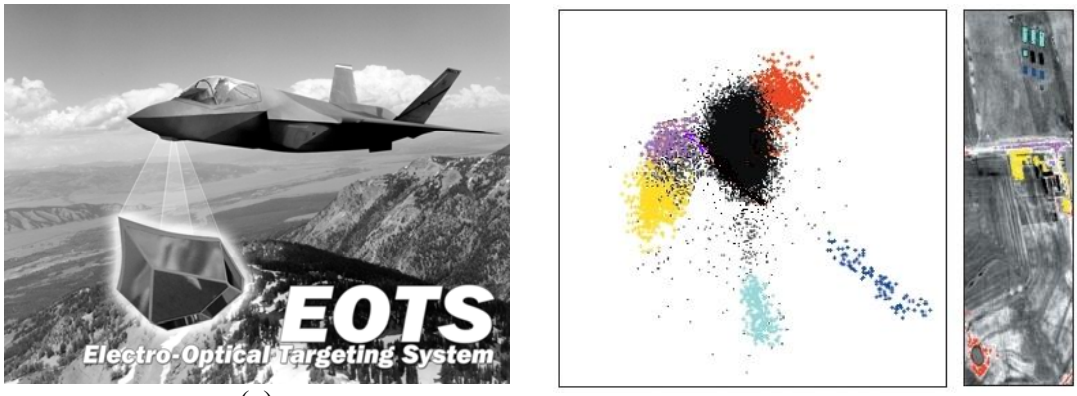
Table 8.2 Highly cited biological databases in bioinformatics study

Database type	Name	Major content
DNA sequence	GeneBank	Contains the current knowledge about the sequence information of all organisms.
Protein sequence	Uniprot	Comprehensive resource on protein sequence information, the combination of other three DBs, Swiss-Prot, TrEMBL and PIR.
Protein 3D structure	PDB	Contains the 3D structural information of 43,633 Proteins, Nucleic Acids, and protein/NA complex as of May 22 2007.
Protein and gene interactions	BioGrid	Includes more than 116,000 protein/ gene interactions of four species.
Micro-array	Gene Expression Omnibus	A comprehensive repository for gene expression data browsing, query and retrieval.

With the biological data are being generated everyday, how to manage the data efficiently is a challenge. For example, as of May 22 2007, the PDB database contains the 3D structural information of 43,633 types of macromolecules. This number still grows by more than 20% every year, and will reach to 2 millions in the near future. The present queries to the protein database are based on text keywords, such as the name of the protein or its function. It is insufficient for more advanced applications, e.g., drug design and protein interaction analysis, where the queries are based on the 3D structures of the proteins. New indexing methods have to be developed to enable such applications for efficient retrieval. The processing of high-throughput micro-array data is another case. A Micro-array record can contain 500,000 testing spot, or attributes, showing the gene expression/protein profiles. How to build the index to speed up the accurate retrieval of the array data is also important.

Multimedia Databases

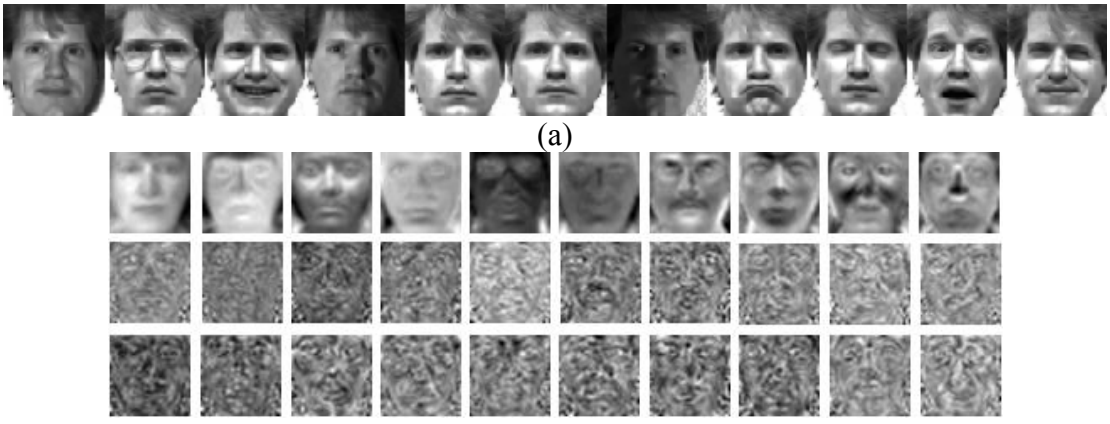
The rapid growth of the World Wide Web saw the explosion of the multimedia data, video and audio clips, texts and images. These data need to be stored, indexed and retrieved efficiently for users. Multimedia information processing is important to a number of the real world applications, such as digital entertainments (Computer games, movies, and animations), face and palmprint recognition, medical imaging and diagnosis, and military. In Figure 8.3 and 8.4, we illustrate two cases where the still images are utilized to support the hyperspectral airborne target and the human face recognitions.



(a) (b)

Figure 8.3 Hyperspectral dimensionality reduction for affordable spectral systems. (a) Electro-optical targeting system (b) Left. Hyperspectral Fisher discriminant subspace scatterplots showing color-coded target class clusters; Right. Hyperspectral image overlayed with color-coded target classes

The electro-optical targeting system (EOTS) (Figure 8.3 (a)) is deployed to detect and recognize the potential air/ ground targets to aid pilot decision. Real time analysis and response to the EOTS sensor inputs is of high priority for military tasks. A database containing the hyperspectral images of various types of the manmade and the natural objects, such as bridges, trees, soils, enemy vehicles, etc., is constructed. The labeled data (Figure 8.3(b) Right) are used as the training samples to learn the Fisher discriminant subspaces for dimensionality reduction (Figure 8.3 (b) Left), thereby, given some new testing images the recognition time can be reduced significantly in the lower dimensional space. The goal for real time recognition on decision support can be achieved.



(a)

(b)

Figure 8.4 Face recognition with Yale facial image database (a) Sample face images of one subject captured with changing facial expressions and illuminations. (b) From top to down. PCA, Fisher discriminant and Laplacian subspaces for dimensionality reduction.

Face recognition is one of the most fundamental and challenging problems to solve in computer science. Human can quickly recognize the human faces under various conditions, but it is difficult for computers to reach such level of competence. The current state-of-the-art face recognition technology such as the eigenfaces and the fisherfaces can

be effective when applied to the 2D/ 3D facial image databases in experiments. For the Yale face image database containing a modest number of training sampes (in Figure 8.4 (a) shown several of the sample images), Figure 8.4 (b) illustrates the PCA, the LDA and the Locality Preserving Subspaces constructed for dimensionality reduction. For small database the accuracy performance can be quite satisfactory, but more advanced techniques for feature extraction has to be developed to handle the large facial image databases.

Besides the 2D/ 3D images, the wealth of the video and audio information in World Wide Web also poses the challenges for dimensionality reduction.

Geographic Information System (GIS)

A GIS can answer the geographic questions, like 'What is the average house-price in district A over the last three years?' or 'Which is the nearest hospital to the central bus stop?' To do so, the GIS builds links between the map features and the attribute tables as depicted in Figure 8.5. A map feature represents an object on the map, such as a tree, a house, or a bridge, etc. It has a location, a shape, and a symbol describing its characteristics. There are three major types of features, point, line and polygon in a typical GIS, defining the geometries of the various kinds of objects. A set of the features and tables make up a theme, and a couple of the themes form a geographic database.

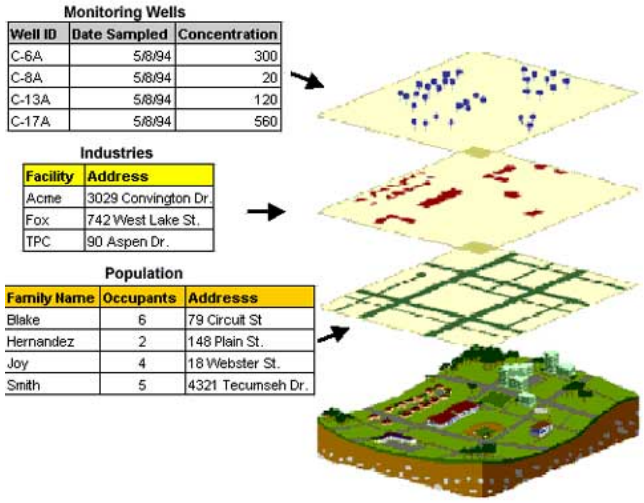


Figure 8.5 Attribute tables are linked to the map features to form a geographic database

To query the spatial information in GIS, extensions have been made to the spatial SQL language. In spatial SQL, a set of the spatial operators are designed and implemented to add the spatial constraints to SQL queries. For example, the following statements will return the information on the hospitals that are less than 2 miles away from the central bus stop.

SELECT *
FROM bus_stop, hospital

WHERE
DISTANCE(hospital.location, bus_stop.location) < 2
AND bus_stop.name = 'central'

For GIS, indexing the map features can help to speedup query processing. Minimal bound rectangle method has been implemented in spatial SQL for such purpose. Also, the machine learning methods, such as the clustering algorithms and the dimensionality reduction techniques, can potentially be employed to give additional supports.

8.1.3 Computing architecture of DBMS

The computing architecture of DBMS follows a trend of decentralization. There exist 3 types of architecturs, centralized, Client/Server, and Peer to Peer.

Centralized DBMS maintains the data in a single database server (server farm) to support user queries. The centralized architecture is highly reliable and secure, preferred by the Banks and hospitals where everything has to be put tightly under the control. The data integrity can also be well maintained in the centralized structure.

A distributed database is a network of the database servers that may distribute across a wide area. Users can simultaneously access and manipulate the data in several databases, and each database in the system is managed by its local database server, as depicted in Figure 8.6, an illustration of the Oracle DBMS architecture.

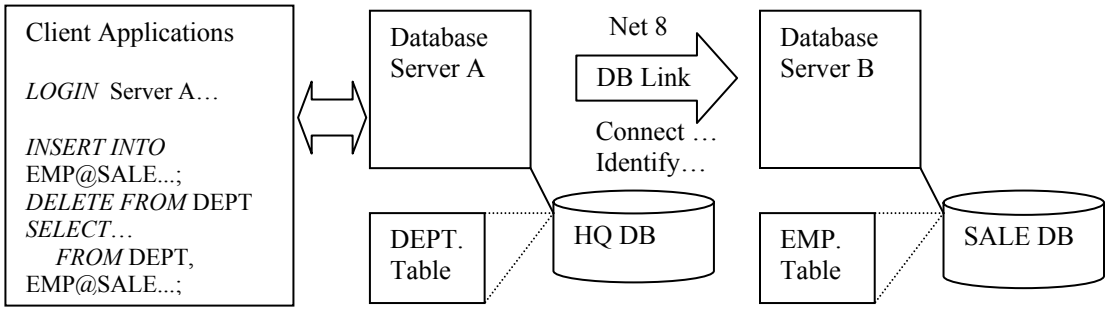


Figure 8.6 Computing architecture of distributed DBMS

Two database servers A and B in the Oracle system are connected through the Net8 protocol for communication. They host the database HQ and SALE, which contains the table DEPT and EMP respectively. The user applications that login into the local Server A can access the data in Server B as well. All the database servers in the Oracle network are addressed by their global names defined in a hierarchical structure maintained by the name server. Figure 8.7 depicts the hierarchical namespace of two sample databases 'STOCK' hosted by two different database servers.

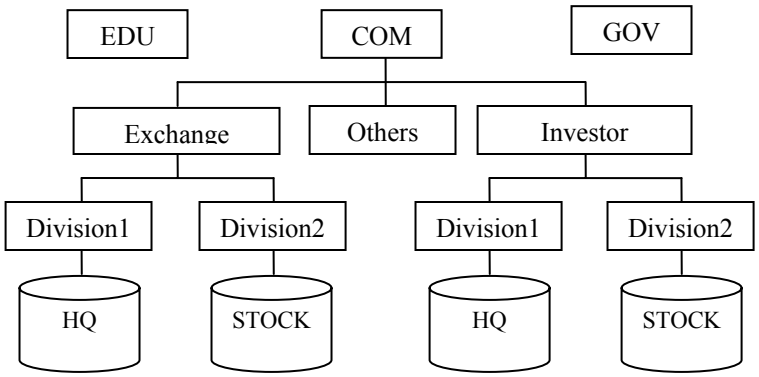


Figure 8.7 Hierarchical namespace of distributed DBMS in Oracle system

Oracle creates database links based on the global names to define a one-way communication path from an Oracle database to another. For example, the global names of the two STOCK databases are stock.division2.exchange.com and stock.division2.investor.com. The following SQL statements creates a database link to the remote site,

CREATE DATABSE LINK stock.division2.investor.com;

After creating the database link, user applications on the local database, stock.division2.exchange.com, can retrieve data from the remote database stock.division2.investor.com. In the case we need to check out all the stock accounts created by the same customers we can launch the query,

*SELECT **
FROM ACCOUNT a1, ACCOUNT@stock.division2.investor.com a2;
WHERE a1.customerID = a2.customerID;

Recently, Peer-to-Peer (P2P) has appeared as a new architecture of distributed computing. In P2P, all the peers are born equal. There are two generations of P2P networks so far. In the first generation P2P, e.g., Gnutella, query and routing are done through broadcast. The flooding of message causes the network overhead. In the second generation, structured P2P, such as Kademila and Chord, querying and routing are based on the distributed hash tables, where each peer can reach its target for data exchange within a number of hops. The indexing of the data on each peer is locally managed to facilitate the query processing.

8.1.4 Indexing algorithm in RDBMS

DBMS index addresses the physical locations for direct data access. The tree based structures, Hash tables, and Linked list are the popular index models widely in use.

In describing the indexing mechanism, a search key is an attribute or a set of attributes used to look up records in the database. An index file consists of the records (index

entries) in the form <search-key, pointer>, where search-key is the indexing attribute(s) and pointer indicates the physical locations of the data for direct access. There are two major types of indexes, the Ordered and the Hash indexes. The ordered indexes sort out the values of the search-keys in the numerical or alphabetical order, whereas the hash indexes are distributed uniformly across the physical storage space of data by using a 'hash function.'

The Balanced trees such as the Binary and the Binary plus trees (B tree and B+ trees) are ordered indexes.

B tree

Formally, a B tree of order p (i.e., maximally allowed number of pointers to the child nodes), is defined to satisfy the following conditions,

i) Each internal node in the B-tree is of the form $\langle P_1, < K_1, Pr_1 >, P_2, < K_2, Pr_2 >, ..., < K_{q-1}, Pr_{q-1} >, P_q \rangle$, where $q \le p$. Each P_i is a tree pointer to the child node in the B tree. Each Pr_i is a data pointer, a pointer to the record whose search key field value is equal to K_i (or to the data file block containing that record).
ii) Within each node, $K_1 < K_2 < ... < K_{q-1}$ and for all search key field values X in the subtree pointed by P_i, we have, $K_{i-1} < X < K_i$ for $1 < i < q$; $X < K_i$ for $i = 1$; and $K_{i-1} < X$ for $i = q$.
iii) Each node, except the root and leaf nodes, has at least Roundup(p/2) tree pointers. The root node has at least two tree pointers and all leaf nodes are at the same level.

Figure 8 illustrates a B tree of order $p = 3$, containing eight elements in two levels.

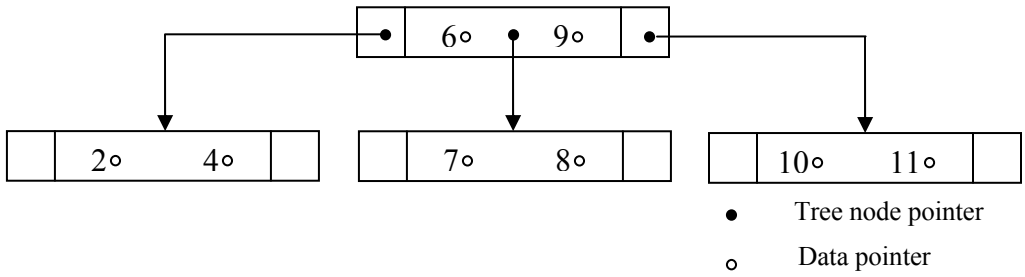


Figure 8.8 B tree of two levels order 3 containing 8 elements

There are two types of pointers, the subtree and the data pointers, in each node of the B tree. The data pointer shows the storage location of the record(s) whose key values equal to those of the tree node, and the subtree pointer segments the data range into a number of the intervals to reduce the time complexity of search.

B+ tree

Most implementations of the tree structure indexes in real world applications are based on the improved B+ method, a variant of the B tree. In a B+ tree, data pointers are stored only at the leaf nodes of the tree, which are linked to provide ordered access on the search field to the records. The internal nodes do not contain the data pointers and some of them have repeated values of the search field to guide the search. Since the internal nodes and the leaf nodes are different in structure, a B+ tree of order p is defined as follows,

i) Each internal node is of the form $\langle P_1, K_1, P_2, K_2, ..., P_{q-1}, K_{q-1}, P_q \rangle$, where $q \le p$ and each P_i is a tree pointer.

ii) Within each internal node, $K_1 < K_2 < ... < K_{q-1}$, and for all search field values X in the subtree pointed by P_i, we have $K_{i-1} < X \le K_i$, for $1 < i < q$; $X \le K_i$ for $i = 1$; and $K_{i-1} < X$ for $i = q$.

iii) Each internal node, except the root, has at least Roundup(p/2) tree pointers, the root node has at least two tree pointers if it is an internal node.

The structure of the leaf nodes of a B+ tree of order p is as follows,

i) Each leaf node is of the form $\langle < K_1, Pr_1 >, < K_2, Pr_2 >, ..., < K_{q-1}, Pr_{q-1} >, P_{next} \rangle$, where $q \le p$, each Pr_i is a data pointer, and P_{next} points to the next leaf node of the B+ tree.

ii) Within each leaf node, $K_1 \le K_2, ..., K_{q-1}$, $q \le p$. Each Pr_i is a data pointer that points to the record(s) whose search field value is K_i.

iii) Each leaf node has at least Roundup(p/2) values, and all leaf nodes are at the same level.

Figure 8.9 illustrates a B+ tree of order $p = 3$ containing nine elements in two levels.

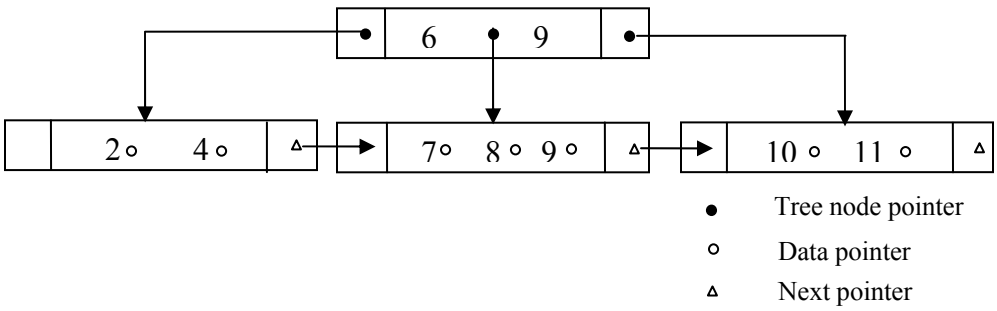


Figure 8.9 B+ tree of two levels containing 9 elements, internal and leaf node order is 3 and 3, respectively.

The order of a B+ tree is determined by the allocated memory size of the tree nodes, the size of the key variables, and the pointers. Suppose the size of the tree nodes is limited to

B = 512 Bytes, a data pointer Pr = 7 Bytes, a search key V = 9 Bytes, a subtree pointer P = 6 Byptes, and a next pointer in the leaf node P_{next} = 6 Byptes. The orders of the tree with respect to the internal and the leaf nodes can be computed by evaluating the two inequalities,

$$(p_{internal} \cdot P) + ((p_{internal} - 1) \cdot V) \leq B \quad (8.1)$$

$$(p_{leaf} \cdot (Pr + V) + P_{next}) \leq B \quad (8.2)$$

Thus, we have $p_{internal}$ = 34, and p_{leaf} = 31. Because the internal nodes of B+ tree do not contain the data pointers, the memory space can be saved to fan out more subtrees linking more data.

In processing a query, a path is traversed in the tree from the root to the leaf node. If there are K search-key values in the file the path is no longer than $\log_{p/2}(K)$. A node is generally the same size as a disk block, typically 4 Kilobytes, and p is typically around 100. With 1 million search key values and p = 100, at most $\log_{50}(10^6) = 4$ nodes are accessed in a look up. The insertion and deletion operations are recursive in nature and can cascade up or down the B+ tree to reshape its structure.

The insertion operation is carried out by examining the state of the tree nodes, either overflow or still having empty space to hold new entries. The algorithm for insertion is summarized as follows,

Algorithm Insertion
Begin
Insertion()
  Find the leaf node in which the search-key value is matched;
  If there is room in the leaf node L
    Insert <search-key value, record/ bucket pointer> pair into leaf node at the appropriate position;
  Else Split(L)
  Endif

Split(L)
  Sort the <search-key value, pointer> pairs in L and the one to be inserted;
  Place the first Roundup(p/2) in the original node, and the rest in a new node;
  Let the new node be L_{new}, and let K_{least} be the least key value in L_{new}, insert <K_{least}, L_{new}> in the parent of the node being split;
  If the parent L_{parent} is full
    If L_{parent} is not root
      Split(L_{parent});
    Else
      Split(L_{parent});
      Increase tree level by one;
    Endif
  Endif
End

The deletion operation of B+ tree seems to be more complicated than insertion. It involves rebalancing the nodes in case of the underflow of the parent of the node being deleted. The underflow of the internodes means that it has only one fan out.

Algorithm Deletion
Begin
Deletion()
  Find the leaf node in which the search-key value is matched;
  Remove <search-key value, record/ bucket pointer> from the leaf at the appropriate position;
  If the parent L_{parent} is overflow
    Rebalance(L_{parent});
  Endif

Rebalance(L_{parent})
If L_{parent} is root
  Collapse root and decrease tree level by one;
 Else
    Check number of entries in immediate neighbors (siblings) of L_{parent}'s only child node.
    If current node is minimal sized
      If sibling node has enough rooms
        Merge with sibling and remove parent of current node;
       Else
        Redistribute the pointers between the current node and the sibling such that both have more than the minimum number of entries;
        Adjust the parent nodes by propagating to upper tree levels;
      End if
     Else
       Split current node and remove duplicate entries in parent;
    Endif
Endif
End

Figure 8.10-13 illustrate an example of the deletion operation in B+ tree, where the tree structure is rebalanced after deletion by merging or splitting the nodes of underflow.

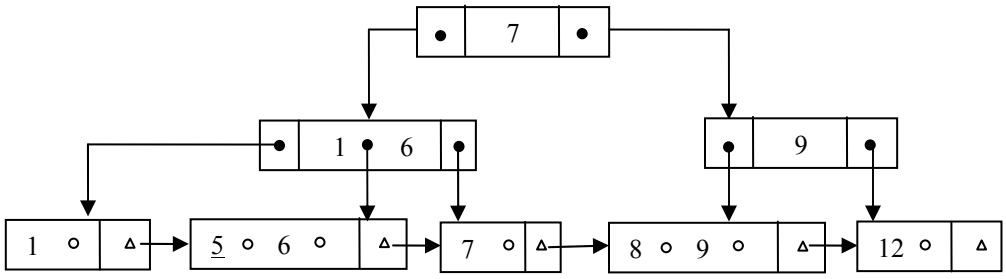


Figure 8.10 Simple deletion in leaf node with search-key value equal 5

In Figure 8.10, the data with the key value equal 5 is to be deleted. The procedure first searches all the entries by traversing the tree. As after deletion the parent node is not underflow, the tree keep still the balance, so, there is no need for rebalancing. Next, consider deleting the data entries whose key values equal 12, shown in Figure 8.11.

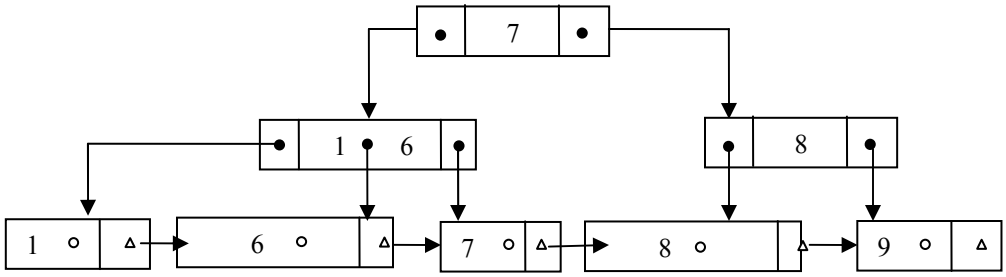


Figure 8.11 Split to rebalance the node containing search-key value equal 8 and 9 after deletion of node with key value 12.

Because the parent has only one fan out to the node containing the key values 8 and 9 after deletion, the parent node is underflow and needs rebalancing. Since its child node has more than one key value, we can do simply split to redistribute the data, similar to the insertion procedure. The parent node is also modified by removing the duplicated keys and the associated pointers. Nevertheless, if there are not enough keys in the node for splitting, merging is performed for rebalancing. There are two possible ways of merging the sibling tree nodes. In the first case, if the sibling nodes still leave empty rooms to hold new data, then the two sibling nodes can be merged directly, accordingly the parent node will be removed, as shown in Figure 8.12 the result of merging after deleting the node with the key value 9.

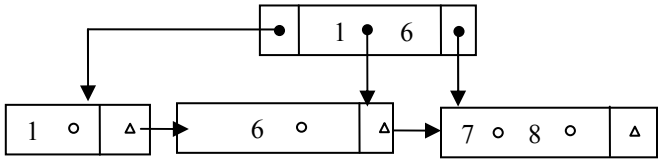


Figure 8.12. Merge to rebalance the node containing search-key value equal 7 and 8 after deletion of node with key value 9. Tree level also decreases by one.

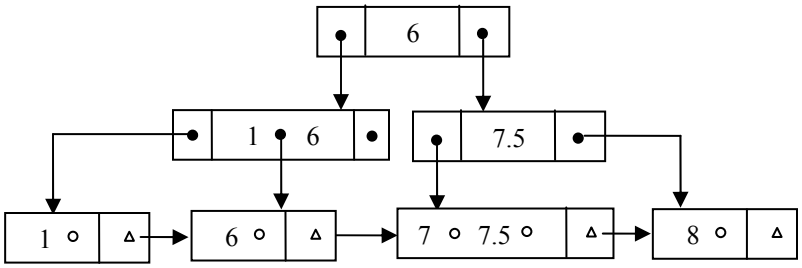


Figure 8.13. Merge with propagation to rebalance the node containing search-key value equal 7, 7.5 and 8 after deletion of node with key value 9. Tree level increases by one.

Also, it is possible that the sibling nodes do not have the enough rooms to merge with new data. As shown in Figure 8.13, the order of the leaf node is 2 and the sibling of the

node with 8 contains 7 and 7.5, which has no more rooms to merge. So we split the node into two, like doing insertion, to rebalance the tree structure.

Hash Index

Hash indexing is highly efficient in processing the point queries based on the equality of key value. It is suitable to be utilized to handle the database operations such as the join of tables, where the data in multiple tables with equal key values are to be checked out. In point queries, it can outperform the balanced trees, which are well designed for range queries. A hash function h(key) is a function from the set of all search-key values K to the set of all data-block (bucket) addresses B. Records with different search-key values may be mapped to the same bucket for localization. An ideal hash function should be uniform, i.e., each bucket is assigned the same number of search-key values from the set of all possible values. In addition, it should be random so each bucket will have the same number of records assigned to it irrespective of the distribution of the search-key values.

There are mainly two types of hashing algorithm, the static and the dynamic hashing. In static hashing, function h(key) maps search-key values to a fixed set B of bucket addresses. However, as the databases grow with time, if the initial number of buckets is too small, performance will degrade due to too much overflows. Moreover, if the index size at some point in the future is anticipated and the number of buckets is predefined, or the database shrinks in number of records, significant amount of space will be wasted. The dynamic hashing algorithms were proposed to address these deficiencies. In dynamic hashing, the size of the index structure can grow and shrink according to the number of records actually stored in the database. Thus, the problem of bucket overflow can be solved to enable the quick searches, with the minimal tradeoff of memory space. We will discuss more on hashing in the section.

Inverted index

Inverted index is in fact the sum of all possible searches stored in advance to speed up query processing. It is especially useful to index text data in database, such as emails, web documents, and DNA/Protein sequences. By establishing the inverted index, we can not only gain rapid access to the searched entries, but also learn useful statistic features of the document, such as the pattern of distribution, the frequency, and the relations among the text terms in the document. For instance, a gene sequence may contain hundreds of thousands of nucleotides, A, T, C, and G. Every three associated base pairs, encodes a type of the amino acid. There are totally twenty types of protein amino acid, and approximately 300 types of non-protein amino acid. These encoding sequences together with other features such as the CG pairs may play the dominant role in defining the biological functions of the genes. Taking these brilliant sequences as feature terms, the inverted index can be used to highlight the statistical differences among the different types of genes, to reveal the hidden mechanism behind the true biological process. Also, inverted index is the core component underlying the fast advancing technology of information retrieval, which we will examine closely in the next section.

8.2 Information retrieval

Web search engines and Internet File Sharing are the 2 major applications of Information Retrievial. Dimensionality reduction can help to reduce the bandwidth and the storage overheads in retrieving the high-dimensional data over Internet.

8.2.1 Web search engines

The World Wide Web is growing fast. By 2005, there are more than 19.2 billion web pages and 70 million websites in the world. By February 2007, the number of the web pages has reached to at least 29.7 billion. More than 100 million distinct web sites exist, each hosting 273 pages on average. The number still grows at the pace of about 2-3 million sites, and 0.54 to 0.81 billion pages each month according to the survey of Netcraft web server. Due to the scale of the problem, web users need to have the index to help browse the web.

Web search engines build index for the texts, the images, the videos in web pages. They can be centralized or de-centralized. The client/server model is the industry standard at present, while the distributed model becomes increasingly popular as the Internet is growing fast. We will investigate the client/server model and use 'Google' as a case for illustration. We then discuss the distributed models of search engines proposed recently.

Search engines in Client/Server architecture

The company Google was started in 1996 by Larry Page and Sergey Brin, two PhD students then at Stanford. It is now the number one search engine widely used around the world. Google maintains a Client/Server model. The engine consists of millions of commodity-class PCs orignaized as clusters of servers. It can support a peak request stream of thousands of queries per second by reading hundreds of megabytes of data in tens of billions of CPU cycles.

When user queries Google, his/her web browser look up the domain name system (DNS) to map www.google.com to a particular IP address. As the Google services are installed world wide, a DNS-based load-balancing system can choose a cluster with the highest geographic proximity to the user to perform the task. After receive the query, a cluster load balancer that monitor the workload of the Google Web servers (GWSs) will forward the request to the most appropriate one. The GWS then coordinate to search for the query and merge the results to be returned in Hypertext Markup Language (HTML) to the browsers.

The procedure of searching consists of two steps. First, the index servers in the cluster look up the inverted index that maps each query keyword to a list of the document IDs. Then they merge the ID lists, and rank the matched documents to find the matched web pages. Because the data can be tens of terabytes in the raw format and the index file can be several terabytes in itself, the query has to be performed in parallel by dividing the index into small parts called index shards. Each shard is handled by a number of the

index servers. Each PC has its own CPU, and hard disk, and the servers are interconnected through the high speed Ethernet switches for data transfer. Figure 8.14 shows a type server of PCs.

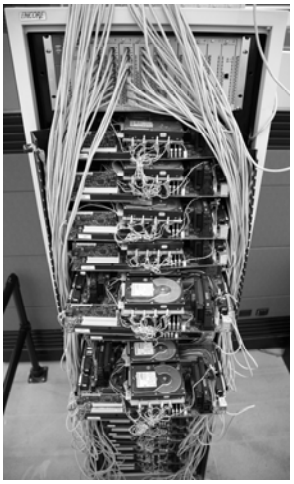


Figure 8.14. A Google server hosting multiple PCs. Clusters of server machines coordinate for searching to answer the user queries.

There are mainly 5 types of servers playing different roles, the web crawlers, the URL servers, the store servers, the indexer and the sorter servers. Their functions are listed in Tables.

Table 8.3 Names and functions of servers in Google architecture

Server Name	Server Function
Web crawler	Download web pages
URL server	Send lists of URLs to be fetched to the crawlers
Store server	Take the web pages fetched by web crawler, compress and store the pages into the repository.
Indexer server	Read the repository, uncompress and parse the documents to build the forward index and the anchor file that contain the information on the link structure of the web pages.
Sorter server	Build the inverted index of the web pages to support the keyword searches.

The web pages are downloaded by the crawler servers. The store servers retrieve the web pages then, compress and store them into the repository. The indexers in turn read the repository, uncompress and parse the documents to build the forward indexes to be stored into the barrels. It also generates the anchor files containing the information on the link structure of the web pages. The new URLs parsed out by the indexer are collected by the URL servers to instruct a new round of crawling of webpages. The anchor files are analyzed with PageRank algorithm to compute the rank of the webpages. Finally, the sorter servers work on the Barrels data to generate the inverted index of all the web pages to speed up the searching process.

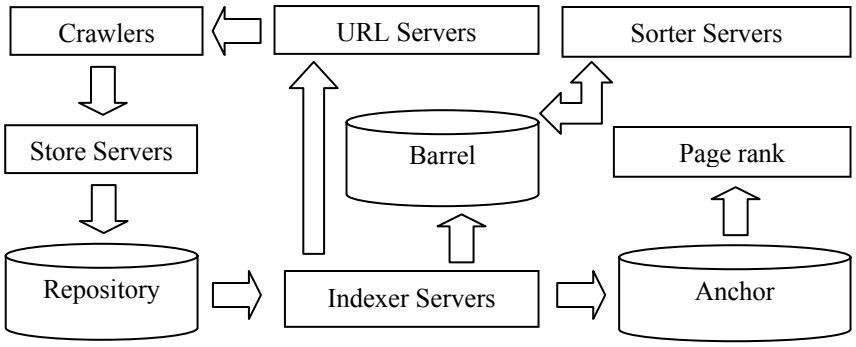


Figure 8.15. Diagram of the main components and the working flow of Google computing architecture in webpage crawling, indexing and ranking to facilitate user retrieval.

The crawlers work in parallel to download the web pages. The URL servers dispatch lists of URLs to a number of crawlers each keeping about 300 connections open to download webpages. At the peak performance, the crawlers can download over 100 web pages per second. The bottleneck for download is DNS lookup. To reduce the lookup latency each crawler maintains a DNS cache so that it doesn't have to look up the DNS servers before download. After the web pages are downloaded, the indexers parse the document to extract the text, the images, the URLs and build the forward index. The forward index consists of the document IDs assigned to each webpage and the associated text contents in form of word IDs. The forward index is stored into the barrels and then retrieved by the sorter servers to create the inverted indexes, which are searched to answer queries. The structures of the forward and the inverted index are illustrated in Figure 8.16 and 17.

docID	wordID:24	nhits:8	hit hit hit hit…
	wordID:24	nhits:8	
	Null wordID		
docID	wordID:24	nhits:8	hit hit hit hit…
	wordID:24	nhits:8	hit hit hit hit…
	wordID:24	nhits:8	hit hit hit hit…
	Null wordID		

Figure 8.16 Forward index containing document ID, word ID, occurrences, and positions

The forward index contains the document and the word IDs. The word ID and its occurrences are 24 and 8 bits integers respectively. The hit records the position of the word occurring in the document. The inverted index creates the data structure in the reverse order to the forward index.

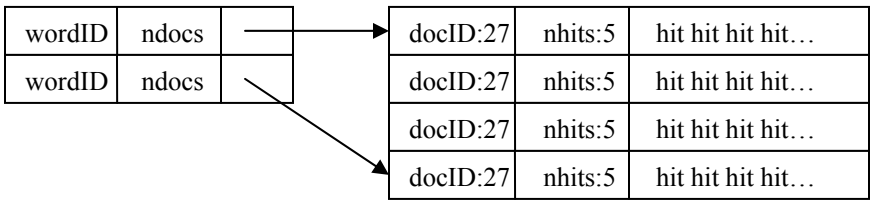


Figure 8.17 Inverted index containing word ID, document ID, occurrences, and positions

Given a keyword in query the search engine can quickly check out the relevant documents by looking up the inverted index. The inverted and the forwarded index are approximately equal in size, while the lexicon, the list of the distinct words in the inverted index is much smaller. The lexicon, the forwarded and the inverted indexes, have the ratio of size about 1:140:140 in Google.

The query is processed by using the Boolean model and the page rank algorithm to achieve the best recall and precision. Typically, Google has about 40,000 documents matched on average for each query. Only those with high page ranks are returned to users. The PageRank algorithm is actually a Markov chain model on the web graph. It was developed by Sergey Brin and Larry Page to rank the quality of the web pages based on the link analysis.

Mathematically, let u be a web page. The let F_u be the set of pages u points to and B_u be the set of pages that point to u. Let $N_u = |F_u|$ be the number of links from u and let c be a factor used for normalization (so that the total rank of all web pages is constant). A simplified version of PageRank is defined in Eq. 8.3.

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v} \tag{8.3}$$

Starting from an initial rank the equation iterates the computation till converge. In the form of matrix, let A be a square matrix with the rows and column corresponding to web pages. Let $A_{u,v} = 1/N$ if there is an edge from u to v and $A_{u,v} = 0$ if not. R is the initial page rank, then Eq. 8.3 can be rewritten as $R = cAR$. By converge, R is the eigenvector of A. It can be worked out by using the power method in linear algebra.

In practice, there is the so called problem of Rank Sink. Consider the two web pages that point to each other but to no other page. And suppose there is some web page which points one of them. Then, during iteration, this loop will accumulate rank but never distribute any rank because there is no outedge. To overcome this problem, Eq. 8.3 is modified by adding a factor E.

$$R = cAR + (1-c)E \tag{8.4}$$

The additional factor *E* modeling the behavior of the web surfer in periodically 'getting bored' of the looping walks and jumping to a random page chosen based on the distribution in *E*.

The PageRank algorithm is fairly straightforward mathematically.

Algorithm PageRank
Begin
Initialize a non-degenerative vector R_0, δ and ε;
While $\delta > \varepsilon$
$R_{i+1} \leftarrow AR_i$
$d \leftarrow \|R_i\|_1 - \|R_{i+1}\|_1$
$R_{i+1} \leftarrow R_{i+1} + dE$
$\delta \leftarrow \|R_{i+1} - R_i\|_1$
Endwhile
End

Here $\|\cdot\|_1$ is the L_1 norm. the factor d increases the rate of convergence and maintain $\|R\|_1$ to be normalized. In practice Google chooses *E* to be a uniform vector with all its elements equal to 0.15. Beside the problem of the sink-links, the dangle-link can also be encoutered. Dangling links are the pages without outgoing links. Because they do not contribute to the ranks of other pages, they can be removed from the computation. After all the PageRanks are calculated, they are added back for ranking.

To deal with the scale of Web, Google builds a custom file system to manage the files of the word-document indexes, the URLs, and the Anchors. It uses large size files to minimize the workload of the file servers. The queries are processed locally for quick responses to users. The architecture of the Google file system is shown in Figure 8.18.

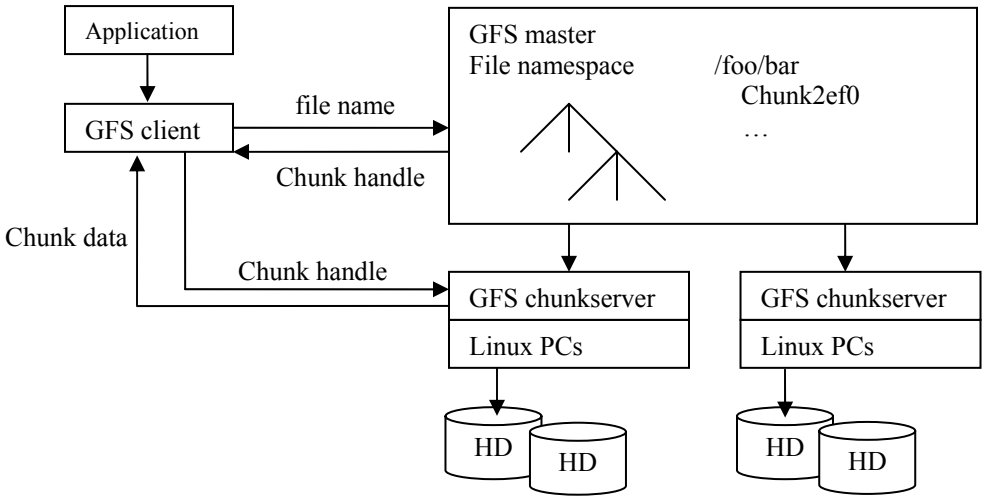


Figure 8.18. Architecture of Google file system managing index, URL, and anchor files

Centralized search engines, however, lack the scalability to deal with the fast growing Web. The entire web holds about 91,000 terabytes of information, including the dynamically generated webpages, the private websites with access limits, and the multimedia information including images, audios and videos. The surface web that can be easily reached by search engines is only about 167 terabytes by contrast. In other words, what we can see by using the current search engines is only a tiny piece of the web. As it is still growing (20 billion indexed pages and several millions of pages everyday) the centralized systems turns out to be inadequate.

Distributed web search engines

Distributed web search engines index the web by sharing the computing resources (bandwidth, storage, and CPU cycles) distributed over the Internet. There are several efforts to develop the distributed web search engines, *Grub*, *Majestic-12*, and *Minerva*.

Grub – Grub is an open source distributed search crawler started in 2000 by Kord Campbell et al. in US.. It was recently acquired by Wikia Inc., July, 2007. Grub users download the grub client software and run it during the computer idle time. The client works in parallel indexing the URLs and sending them back to the main grub server in compressed form. The grub server then assembles the indexes to process the user queries. Grub, nevertheless, is not purely distributed because the indexing and searching is performed on the centralized server. Though it can achieve better coverage over the whole web, it still has the same problem in centralized systems.

Majestic-12 – MJ12 was founded in late 2004 by Alex Chundnovsky in UK.. It works in a similar way as Grub. The MJ12 client software runs on the idle PCs around the world. It crawls, collates, and feeds back the information to the master server. The crawled data will be indexed and searched by MJ12 to answer the queries.

Minerva – The Minerva project was launched by Matthias Bender et al. at the Max-Planck-Institute of Informatics, Germany. In Minerva, each idle PC manages its local search engine to build the index from the crawled web pages. In routing the query, the distributed hash tables (DHT) are utilized to show which PC contains the index entry for the given keyword. By looking up the index, all the PCs that contain the keyword can be identified. Minerva can find out the web pages rapidly by employing DHT, but it still unable to rank and merge the web pages efficiently to answer the users.

8.2.2 Internet file sharing

Indexing and searching for distributed web information retrieval is still challenging. In contrast, for Internet file sharing, Peer-to-Peer computing has gained widespread popularities among the PC users. It is particularly useful for sharing large audio and video files over the Internet. There are 2 types of P2P models, the unstructured and the structured, according to the different protocols for indexing and query routing.

Unstructured Peer to Peer

Unstructured P2P does not utilize distributed hash table for indexing and query routing. Typical unstructured P2P systems include Napster, Gnutella, Kazaa, and BitTorrent. Napster is an early example of P2P networks, but it still has a master server in central. Gnutella by contrast is fully decentralized without any master servers. Kazaa establishes an overlay network ontop of the supernodes of high processing powers to manage the worker peers. BitTorrent is designed to distribute large files by querying the tracker servers and trading with fairs among the peers.

Napster – Napster was initially designed for sharing music, particularly, mp3 files. Users first publish a list of the mp3 that they have to the central Napster server. The server maintains an index to indicate which peer contains which mp3. It returns the matched IP addresses of the peers that host the file. The peer-to-peer communication can then be started and users can connect to each other for download, as illustrated in Figure 8.19.

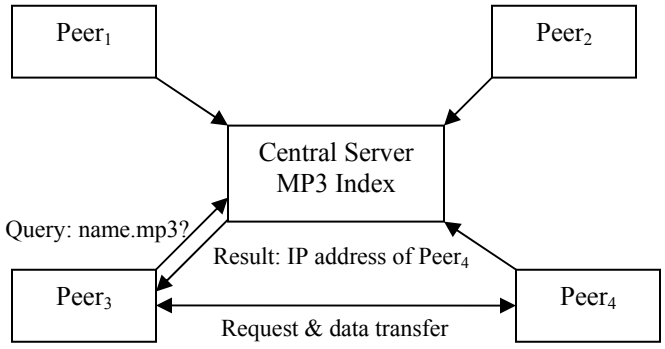


Figure 8.19. Napster. A centralized server maintains the index of MP3 files to coordinate the communications among peers

As Napster relies on the central server to operate the whole network, however it can suffer the single point of failure on the denial-of-service attacks and censorships.

Gnutella – Gnutella, unlike Napster, is a fully decentralized P2P file sharing software with many successful implementations like BearShare and Shareaza. The Gnutella protocol is quite simple yet very powerful. Its client software hosts a list of the audio/video files and the local index for searching. There are basically four types of messages operating the Gnutella network. When users start the Gnutella clients, they send the 'Ping' message to discover their neighboring nodes in the network. All the nodes receiving such a ping message will respond to it with a 'Pong' message. The pong messages are routed back to the node sending the message to establish the connections. A 'Query' message is issued when users want to search for a file. Each node, if having the matching file, is supposed to respond to this message with a 'QueryHit' message, and forwards in turn the query message to its neighbor nodes. For firewalled nodes, a 'Push' message is delivered to make file downloads possible.

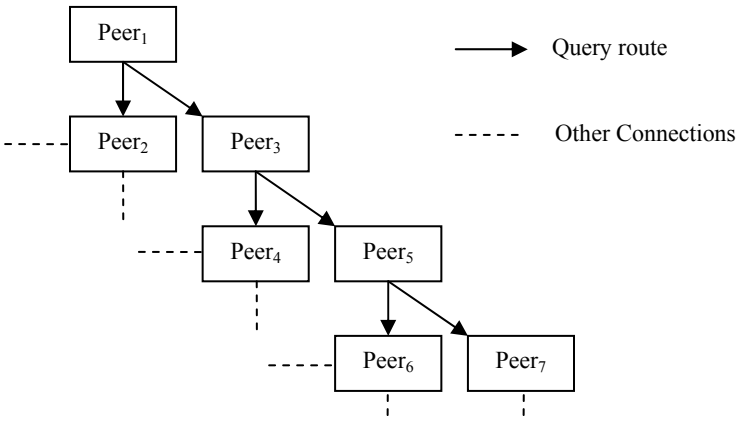


Figure 8.20. Architecture of Gnutella network.

As illustrated in Figure 8.20, a query is sent from $Peer_1$ to its neighbors $Peer_{2,3}$. These two peers then check their local index and forward the query message to the neighbors. The query, in this way, is flooding through the Gnutella network and all the matched results with the IP addresses will be reported to $Peer_1$ for download. To stop messages, a Time-To-Live (TTL) tag is calculated at each hop. On every node the TTL value is decreased by one. When it is zero the message will be discarded to terminate the searching process. There are two main problems for Gnutella network. First, the response latency to return the query result is relatively high as usually it takes long time for the messages to be transferred across a huge number of peers. Second, every node in the network has to sustain the flooding search. When the number of users is large the denial of services can be triggered due to the message overhead. This leads to the development of the Kazza protocol to work out.

Kazza – Different from Gnutella, Kazza incorporate the concept of supernodes. Supernodes are those peers having the stronger processing powers (bandwidth, memory, hard disk). They act as central nodes in a segment of the network. Kazza reduces the network traffic and sets up an overlay of the supernodes for indexing and searching.

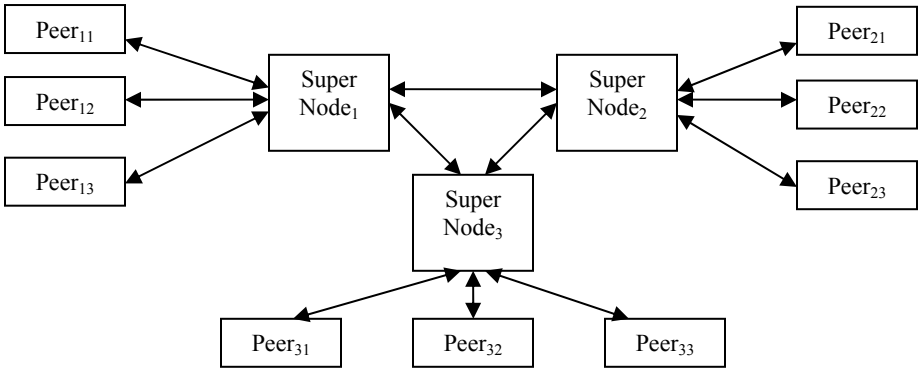


Figure 8.21. Architecture of Kazza network

Kazza flooding searches only the supernodes. If the entry matches to the supernode index then it checks further its member peers and returns the result. It can thus avoid the problems in Napster and Gnutella.

BitTorrent – BitTorrent (BT) is ideal for sharing large size files. For example, many Linux distributions are distributed by BitTorrent. BT can save the file servers a lot of the bandwidth by sharing the files among the users requesting on the same file. It works like Napster except that BT involves a population of the volunteered servers called Trackers to provide the indexing services to users. The Tracker software chops the file to be uploaded into small pieces typically of the size 256kB. It then creates a .torrent file announcing the URL of the Tracker. Users download and read the .torrent file to connect to the Tracker server and begin the download. The Tracker records and updates an index from time to time to show which peer has downloaded which part of the file. So, other users can read the index to find the existing owners of the request content. They can then contact the owners and trade fairly with them for exchange. The more the users downloading a file, the more the partners they can trade with, and the faster the file can be downloaded.

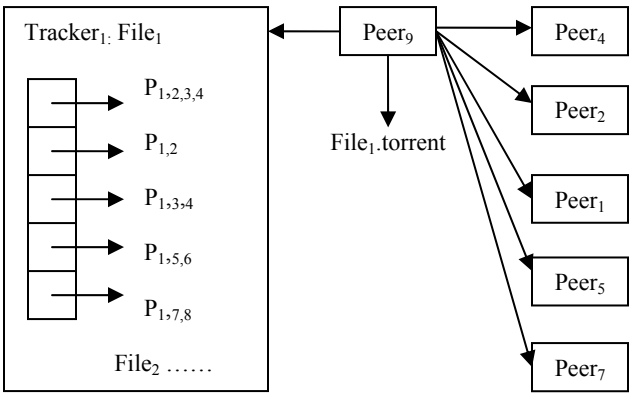


Figure 8.22. Architecture of BitTorrent

In Figure 8.22, consider a simplified case where a file is chopped into five pieces and uploaded to the server $Tracker_1$. A .torrent file containing the URL of $Tracker_1$ is then published and downloaded by users who want to download $file_1$. The users connect to the Tracker server and read the index to know which other peers have downloaded any part of the file before. Suppose $Peer_9$ possesses $segment_1$ which $Peer_7$ hasn't downloaded, and $Peer_7$ owns $segment_5$ that $Peer_9$ doesn't have. $Peer_9$ can then negotiate with $Peer_7$ to trade the stuff. After trading, P_7 will have $segment_1$ from $Peer_9$, while $Peer_9$ will have $segment_5$ from $Peer_7$. In doing so, by trading with the current holders of the file segments, $Peer_9$ can have all the parts to assemble and reconstruct the whole file. Accordingly, the index file in the tracker server will also be updated dynamically to notify the newcomers that $Peer_9$ is also a source where they can ask for download. There are many BitTorrent Trackers to guide the download of files by peers.

The unstructured P2P file sharing is quite straitforward for design and implementation. But they suffer the problem of the single point of failure and the overhead of message flooding. Distributed Hash Tables (DHTs) can potentially be utilized to address these problems.

Structured Peer to Peer

Structured Peer to Peer file sharing is based on the DHT. Basically, the peers in a DHT network can self-organize an overlay on top of the Internet, and each peer needs to know only a few other peers in the system. The network can work efficiently with thousands or millions of nodes. A hash function is employed to address the nodes and the resources to the same ID space. A distance metric is then defined and utilized to allocate the resource to the nodes with matched IDs. The time complexity to find a resource is factorized.

The general steps of using DHT for storage and retrieval are as follows. To publish the IP address of a file with a given filename, the SHA1 hash function hashes the filename to produce a 160-bit key *k*. A message put(*k*, IP address) is sent to an existing node connected to the DHT network. The message is then forwarded from node to node through the overlay until it reaches the closest node(s), where the message will be stored. Any other client to retrieve the message can hash the filename to produce *k*, and ask any DHT node to find out the node containing *k* with the message get(*k*). The message is again routed through the overlay to the node responsible for *k* to reply to the users.

The hash function SHA1 can uniformly and randomly partition the keyword space and the node space to avoid the problem of identity collision. There are different types of the DHT overlay networks for file sharing, depending on how the hash indices and the distance metric are defined. Two popular DHT based networks are the Kadmelia and the Chord.

Kadmelia — Kadmelia now widely used is designed by Petar Maymounkov and David Mazieres. It uses a *XOR* metric to define distance. Two node IDs or a node ID and a key, which are usually 160 bits of binary numbers, are *XOR*ed and. The result is converted to integer used as the distance between them. The hash index for routing consists of a list for each bit of the node ID. For instance, if a node ID has 160 bits, a node will keep 128 lists. A list has many entries which are typically the node IDs of other nodes and their IP addresses plus port numbers. Nodes in the n-th list must have the IDs whose n-th bit is different from that of the hosting node, while the first n-1 bits are the same. Therefore, its always easy to fill the first list by ping-pong other nodes as statistically 1/2 of the nodes in the network have the different value in the first bit, and 1/4 for the second list, so on and so forth. Thus, every list corresponds to a specific distance from the node. With an ID of 160 bits, every node in the network will classify other nodes in one of 128 different distance ranges, one specific distance range per bit. As nodes are encountered on the network, they are added to the lists to update the routing table, while the failure nodes will be removed. Figure 8.23 illustrates an example of the Kademila network.

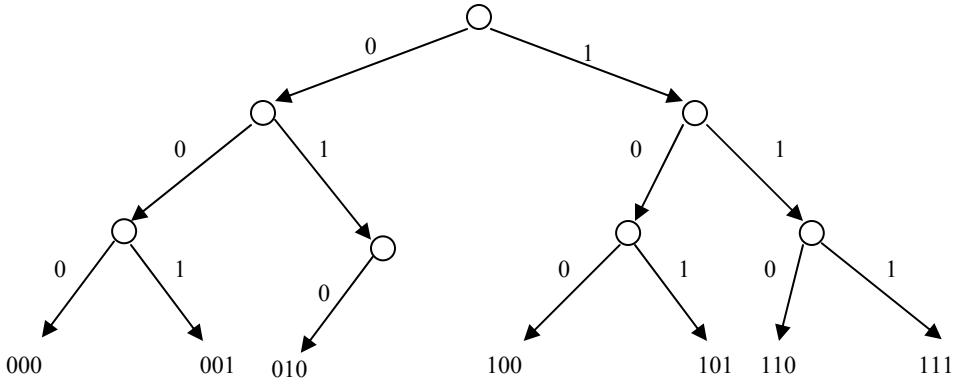


Figure 8.23. Illustration of Kademila network with 3 bits hash outputs

In Figure 8.23, there are seven peers in the Kademila network, forming a B-tree structure. Each level of the tree corresponds to a bit of the Hash index. Now, consider the node whose ID is '110'. For the first bit, the three nodes '000', '001', and '010' will be the candidates to be put into its list because they all have '0' rather than '1' in the first place. For the second bit, nodes '100' and '101' will be the candidates because they all have the same value in the first bit but different in the second. Finally, for the third bit, '111' will be listed because the first two bits are all equal but different in the third. After generating the routing table users can efficiently locate the nodes and the resources by checking out their IP addresses for the next hop.

Given a keyword of the data to upload, the Kademila client software initiates a FIND_NODE request by querying to the closest ones to the key value. The recipients that receive the query in turn will lookup their routing tables to find the neighbors. Because every node has a better knowledge of its own neighbors than any other node, the query will be forwarded closer and closer to the searched key. The search continues until node nodes are returned that are closer than the best previous results. And the IP addresses of the best nodes matched in the result list will be returned to the requesters. The data can thus be sent to the target node for storage using the STORE message. To provide redundancy in case of node failures, the data are duplicated and stored in multiple nodes that are the close to each other. This also increases the speed of searching.

To retrieve the uploaded data given some searching keyword, Kademila first maps the input to a key string. The key string is then used for routing to locate the node(s) that contain the resource. Because the nodes are hierarchically organized, the complexity of searching is only $O(logN)$ for an overlay containing N nodes. Recently, the Kademila algorithm has been implemented in a number of the widely used file sharing software, such as BitTorrent, BitComet, and eMule, where its effectiveness has been demonstrated.

Another DHT based file sharing system that has drawn considerable academic attentions is the Chord algorithm, a result of the IRIS project (Infrastructure for Resilient Internet Systems) by MIT, UC Berkeley, Rice U. & Microsoft, and New York U.

Chord — The key idea of Chord is similar to that of the Kademila except that the distance metric and the organization of the routing table are different. The IP addresses of the nodes and the keywords are mapped to the same ID space by using the SHA1 hash function. All the nodes are sorted according to the values of the node IDs (ascending order), and virtually arranged into a ring structure. Given a query keyword *k*, the hash function first computes its hash output, usually a 160 bits string. The binary string is then converted to an integer, say, *a*. The storage node can be found according to the rule of Chord at the node whose ID is greater than or equal to *a*. This node is called the successor of *k.* The routing table of a node maintains a list of the nodes in various distance ranges taking the factorized step length of 2^n for the n-th entry. Therefore, for a ID space of m bits it requires only m entries to cover all the nodes. Figure 26 depicts the topology of a sample Chord ring having 10 nodes.

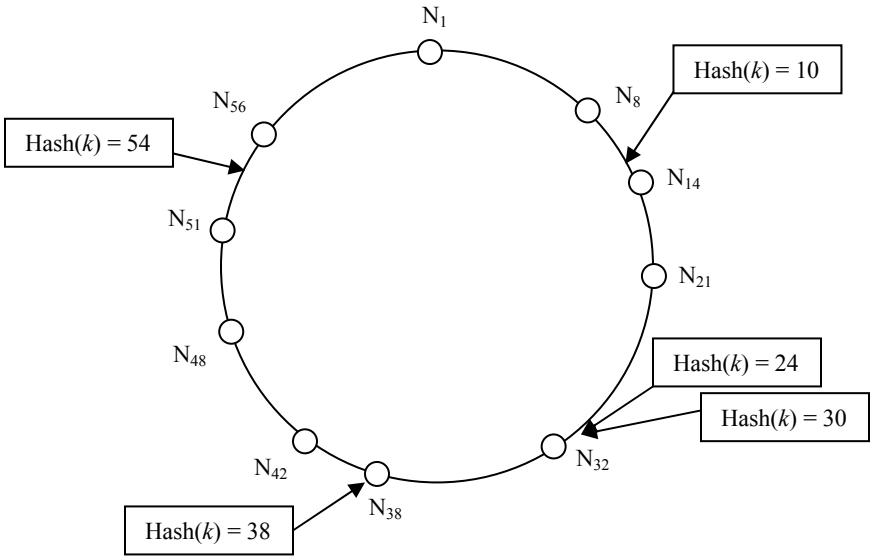


Figure 8.24. Illustration of a sample Chord ring having 10 nodes

As shown in Figure 8.24, the queries with the keywords whose hash value is 10 will be matched to its successor, the node N_{14}, whose ID value is 14. Similarly, hash values 24 and 30 will be matched to N_{32}, and 38 matched N_{38}, 54 to N_{56}. The step length for routing is set to 2^{i-1} where *i* smaller than *m*, the number of bits of the ID space, correspond to the i-th entry in the routing table. The searching process proceeds as follows. When a node gets a query, it first checks whether it has the matched record. If not, it looks up the routing table to find the successor of the key and forward the query to it. The recipient node repeats the above steps until the records have been found, as shown in Figure 8.25.

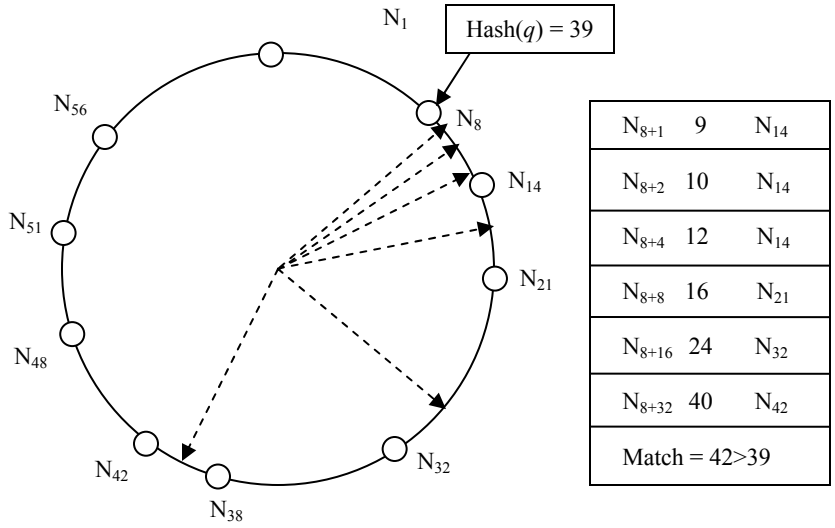


Figure 8.25. Node$_8$ receives the query ID 39 and find the result at its successor N_{42}

In Figure 8.25, given the hash value of the query keyword 39, node$_8$ finds that it has no matched result locally. It thus looks up the routing table. As it is found that the sixth entry with the node ID 42 is the direct successor of 39, the query will be forward to this node to check further.

The DHT algorithms, typically Kademila and Chord, establish a robust and scalable framework to realize the large scale distributed indexing and searching for Internet file sharing and applications. The indexing and routing mechanism improves the wide area storage networks, and protects the widely distributed applications from malicious attacks. Though still the challenge, the aggregation of the storage, bandwidth, and computing resources upon a secure layer of indexing and routing mechanisms will revolutionize our concept about the new generation of Internet computing.

8.3 Gabor wavelets

Gabor wavelet transform, also known as windowed Fourier transform, is localized in space. This property makes many functions and operators using wavelets "sparse" when transformed into the wavelet domain. The sparseness is an advantage for a number of useful applications such as data compression, feature detection and de-noise in images. Gabor wavelet has aroused considerable interests in vision research. There is the evidence that the Gabor wavelets are of similar shape as the receptive fields of orientation-selective simple cells in the primary visual cortex (V1). Lee [8.1] modeled such cells by the following Gabor wavelet function:

$$\psi(x,y,\omega_0,\theta)=\frac{\omega_0}{\sqrt{2\pi}K}e^{-\frac{\omega_0^2}{8K^2}\left(4(x\cos\theta+y\sin\theta)^2+(-x\sin\theta+y\cos\theta)^2\right)}\left[e^{i(\omega_0 x\cos\theta+\omega_0 y\sin\theta)}-e^{-\frac{K^2}{2}}\right] \quad (8.5)$$

The wavelet is zero centered. θ is the orientation of the wavelet. ω_0 is the size of the Gaussian window and the frequency of sinusoid basis. K is a constant. A Gabor wavelet hence is the product of the Gaussian window function and the sinusoid basis at various orientations, scales and frequiencies, as illustrated in Figure 8.26.

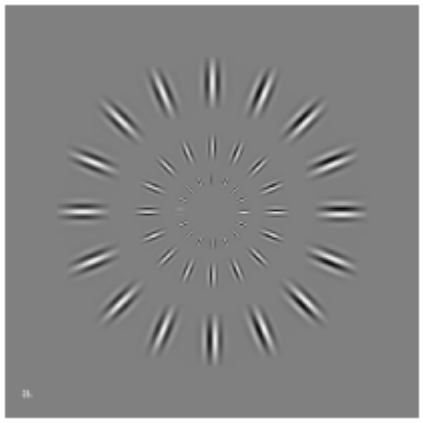


Figure 8.26. Gabor wavelets at various orientations, scales and frequencies (Taken from [8.1])

Each member of this family of Gabor wavelets models the spatial receptive field structure of a simple cell. The response of a simple cell is the projection of an image onto a Gabor wavelet, which is the inner product of the image f with the receptive field centered at (x_0, y_0),

$$\begin{aligned} R_s &= \psi * f \\ &= \int_x \int_y \Psi(x_0 - x, y_0 - y : \omega_0, \theta_0) f(x, y) dx dy \end{aligned} \tag{8.6}$$

References

[3.1] K.I. Kim, M.O. Franz, and B. Scholkopf. "Iterative Kernel Principal Component Analysis for Image Modeling," IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 27, No.9, pp.1351–1366, 2005.

[3.2] S. Mika, B. Schölkopf, A.J. Smola, K. Müller, M. Scholz, G. Rätsch. "Kernel PCA and De-Noising in Feature Spaces," NIPS, pp. 536-542, 1998.

[3.3] X. He, http://people.cs.uchicago.edu/~xiaofei/research1.html.

[3.4] J. Yang, A.F. Frangi, J.Y. Yang, D. Zhang, Z. Jin. "KPCA Plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 2, pp. 230-244, 2005.

[3.5] N. Belhumeur, J. Hespanha and D. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition Using Class-Specific Linear Projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19, No. 7, pp. 711-720, 1997.

[4.1] Ruhr Univ., http://www.radiologie.ruhr-uni-bochum.de/.

[4.2] Univ. Queensland, http://study.itee.uq.edu.au/.

[4.3] Encylopaedia Britannica, Inc., http://hominidae-cranial-capacity.

[4.4] C. Liu. "Gabor-based Kernel PCA with Fraction Power Polynomial Model for Face Recognition," IEEE Transactions on Patten Analysis and Machine Intelligence, Vol. 6, No. 5, pp.572-581, 2004.

[4.5] xantox. http://strangepaths.com/.

[4.6] NASA. http://aemc.jpl.nasa.gov/activities/bio_regen.cfm

[4.7] I. Guyon, V. Vapnik, B.E. Boser, L. Bottou, S.A. Solla. "Structural Risk Minimization for Character Recognition," NIPS, pp. 471-479, 1991.

[4.8] D. Zhang, Hui Peng, Jie Zhou, S.K. Pal, "A Novel Face Recognition System Using Hybrid Neural and Dual Eigenspaces Methods," IEEE Transactions on Systems, Man and Cybernetics, Part A, Volume 32, Issue 6, Nov. 2002, pp. 787 – 793

[4.9] M. Vose. The Simple Genetic Algorithm: Foundations and Theory, MIT Press, Cambridge, MA, 1999.

[4.10] S.K. Pal, S. Mitra, P. Mitra. "Rough-Fuzzy MLP: Modular Evolution, Rule Generation, and Evaluation," IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 1, pp. 14-25, 2003.

[4.11] Wolfram Mathematica. http://www.wolfram.com/products/mathematica.

[4.12] S.T. Roweis, and L.K. Saul. "Nonlinear Dimensionality Reduction by Locally Linear

Embedding," Science, Vol. 22, pp. 2323-2326, 2000.

[4.13] J.B. Tenenbaum, V. Silva, and J.C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, pp. 2319-2323, 2000.

[4.14] M. Belkin, P. Niyogi. "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," Neural Computation, Vol. 15, Vol. No. 6, pp.1373-1396, 2003.

[4.15] J. Yang, D. Zhang, A.F. Frangi, J.Y Yang. "Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 26, No.1, pp. 131-137, 2004.

[5.1] X. He, S. Y, P. Niyogi, and H.J. Zhang. "Face Recognition Using Laplacianfaces," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 3, pp. 328- 340, 2005.

[5.2] S.T. Roweis and L.K. Saul. "Nonlinear Dimensionality Reduction by Locally Linear Embedding," Science, Vol. 290, pp. 2323-2326, 2000.

[5.3] K. Liu, Y. Q. Cheng, and J. Y. Yang. "Algebraic Feature Extraction for Image Recognition based on an Optimal Discrimination Criterion," Pattern Recognition, Vol. 26, No. 6, pp. 903-911, 1993.

[5.4] J. Yang, D. Zhang, A.F. Frangi, and J.Y. Yang. "Two Dimensional PCA: A New Approach to Appearance Based Face Representation and Recongition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 1, pp. 131-137 2004.

[5.5] J. Yang and J. Y. Yang. "From Image Vector to Matrix: a Straightforward Image Projection Technique – IMPCA vs. PCA," Pattern Recognition, Vol. 35, No. 9, pp. 1997-1999, 2002.

[5.6] J. Yang, D. Zhang, Y. Xu, and J.Y. Yang. "Two Dimensional Discriminant Transform for Face Recognition," Pattern Recognition, Vol. 38, No. 7, pp. 1125-1129, 2005.

[5.7] H. Xiong, M.N.S. Swamy, M.O. Ahmad. "Two Dimensional FLD for Face Recognition," Pattern Recognition, Vol. 38, No. 7, pp. 1121-1124, 2005.

[5.8] X. Jing, H.S. Wong, and D. Zhang. "Face Recognition based on 2D Fisherface Approach." Pattern Recognition, Vol. 39, No. 4, pp. 707-710, 2006.

[5.9] The Facial Recognition Technology (FERET) Database, http://www.itl.nist.gov/iad/humanid/feret/feret_master.html, 2002.

[5.10] Matlab Central File Exchange, http://www.mathworks.com/matlabcentral/.

[5.11] A.M. Martinez and R. Benavente. "The AR Face Database," CVC Technical Report, No. 24, June 1998.

[5.12] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu. "An Optimal Algorithm for Approximate Nearest Neighbor Searching," J. ACM, Vol. 45, pp. 891-923, 1998.

[5.13] A.K. Jain, R.P.W. Duin and J. Mao. "Statistical Pattern Recognition: a Review," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 4-37, 2000.

[5.14] M. Kirby and L. Sirovich. "Application of the KL Procedure for the Characterization of Human Faces," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No.1, pp. 103-108, 1990.

[5.15] M. Turk and A. Pentland. "Eigenfaces for Recognition," J. Cognitive Neuroscience, Vol. 3, No. 1, pp.71-86, 1991.

[5.16] K. Liu, Y.Q. Cheng, J.Y. Yang, X. Liu. "An Efficient Algorithm for Foley-Sammon Optimal Set of Discriminant Vectors by Algebraic Method," International Journal of Pattern Recognition and Artificial Intelligence, Vol. 6, No.5, pp. 817-829, 1992.

[5.17] D.L. Swets and J. Weng. "Using Discriminant Eigenfeatures for Image Retrieval," IEEE Transasctions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 8, pp. 831-836, 1996.

[5.18] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriengman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No.7, pp. 711-720, 1997.

[5.19] L.F. Chen, H.Y.M. Liao, J.C. Lin, M.D. Kao, and G.J. Yu. "A New LDA-based Face Recognition System Which Can Solve the Small Sample Size Problem," Pattern Recognition, Vol. 33, No.10, pp. 1713-1726, 2000.

[5.20] H. Yu, J. Yang. "A Direct LDA Algorithm for High-dimensional Data—with Application to Face Recognition," Pattern Recognition, Vol. 34, No. 10, pp. 2067-2070, 2001.

[5.21] J. Yang, J.Y. Yang. "Why Can LDA be Performed in PCA Transformed Space?" Pattern Recognition, Vol. 36, No.2, pp. 563-566, 2003.

[5.22] C.J. Liu and H. Wechsler. "A Shape- and Texture-based Enhanced Fisher Classifier for Face Recognition", IEEE Transactions on Image Processing, Vol. 10, No. 4, 598-608, 2001.

[5.23] W. Zhao, A. Krishnaswamy, R. Chellappa, D. Swets, and J. Weng. "Discriminant Analysis of Principal Components for Face Recognition", in Face Recognition: From Theory to Applications, Edited by H. Wechsler, P.J. Phillips, V. Bruce, F.F. Soulie and T.S. Huang, Springer-Verlag, pp. 73-85, 1998.

[5.24] J. Yang, D. Zhang, A.F. Frangi, J.Y. Yang. "Two-Dimensional PCA: a New Approach to Face Representation and Recognition," IEEE Transactions Pattern Analysis Machine Intelligence, Vol. 26, No.1, pp. 131-137, 2004.

[5.25] J. Yang, D. Zhang, X. Yong, and J.Y. Yang. "Two-dimensional Discriminant Transform for Face Recognition," Pattern Recognition, Vol. 38, No. 7, 1125-1129, 2005.

[5.26] J. Ye and Q. Li. "A Two-stage Linear Discriminant Analysis via QR Decomposition." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 6, pp. 929-941, 2005.

[5.27] B. Schölkopf, A. Smola, and K.R. Muller. "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," Neural Computation, Vol. 10, No.5, pp. 1299-1319, 1998.

[5.28] S. Mika, G. Rätsch, J Weston, B. Schölkopf, A. Smola, and K.R. Müller. "Constructing Descriptive and Discriminative Non-linear Features: Rayleigh Coefficients in Kernel Feature Spaces," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No.5, pp. 623-628, 2003.

[5.29] G. Baudat and F. Anouar. "Generalized Discriminant Analysis Using a Kernel Approach," Neural Computation, Vol.12, No. 10, pp. 2385-2404, 2000.

[5.30] M.H. Yang. "Kernel Eigenfaces vs. kernel Fisherfaces: Face Recognition Using Kernel Methods," Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (RGR'02), pp. 215-220, 2002.

[5.31] J. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. "Face Recognition Using Kernel Direct Discriminant Analysis Algorithms,"IEEE Transactions on Neural Networks, Vol. 14, No. 1, pp. 117-126, 2003.

[5.32] J. Yang, A.F. Frangi, D. Zhang, J.Y. Yang and J. Zhong. "KPCA plus LDA: a Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition," IEEE Transactions on Pattern Analysis Machine Intelligence, Vol. 27, No. 2, 230-244, 2005.

[5.33] H.S. Seung and D.D. Lee. "The Manifold Ways of Perception". Science, Vol. 290. pp. 2268 – 2269, 2000.

[5.34] J.B. Tenenbaum, V. de Silva and J. C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, Vol. 290, pp. 2319-2323, 2000.

[5.35] S.T. Roweis and L.K. Saul. "Nonlinear Dimensionality Reduction by Locally Linear Embedding", Science, Vol. 290, pp. 2323-2326, 2000.

[5.36] M. Belkin, P. Niyogi. "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," Neural Computation, Vol. 15, No. 6, pp.1373-1396, 2003.

[5.37] O. Kouropteva, O. Okun, and M. Pietikainen. "Supervised Locally Linear Embedding Algorithm for Pattern Recognition," Lecture Notes in Computer Science, Vol. 2652, pp. 386–394, 2003.

[5.38] D. Ridder, M. Loog and M. Reinders. "Local Fisher Embedding," Proc. 17th International Conference on Pattern Recognition (ICPR2004), 2004.

[5.39] N. Vlassis, Y. Motomura, and B. Krose. "Supervised Dimension Reduction of Intrinsically Low Dimensional Data", Neural Computation, Vol. 14, No.1, pp.191–215, 2002.

[5.40] L.K. Saul and S.T. Roweis. "Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds", Journal of Machine Learning Research, Vol. 4, pp. 119-155, 2003.

[5.41] M. Brand. "Charting a Manifold," Neural Information Processing Systems 15, NIPS, 2002.

[5.42] Y. Bengio, J.F. Paiement, and P. Vincent. “Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering”, NIPS,2003.

[5.43] X. He, P. Niyogi, “Locality Preserving Projections,” NIPS, 2003.

[5.44] X. He, S. Yan, Y. Hu, P. Niyogi, H.J. Zhang. “Face Recognition Using Laplacianfaces,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No.3, pp. 328-340, 2005.

[5.45] S. Yan, D. Xu, B. Zhang and H.J. Zhang. “Graph Embedding: A General Framework for Dimensionality Reduction,” CVPR, pp. 830 – 837, 2005.

[5.46] H.T. Chen, H.W. Chang, and T.L. Liu. “Local Discriminant Embedding and Its Variants,” CVPR, pp. 846 – 853, 2005.

[5.47] Y. Koren and L. Carmel. “Robust Linear Dimensionality Reduction,” IEEE Transactions on Visualization and Computer Graphics, Vol.10, No.4, pp. 459-470, 2004.

[5.48] G.H. Golub and C.F. Van Loan. Matrix Computations, The Johns Hopkins University Press, Baltimore and London, Third edition, 1996.

[5.49] P. J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. “The FERET Evaluation Methodology for Face-Recognition Algorithms,” IEEE Transactions on Pattern Anal. Machine Intelligence, Vol. 22, No. 10, pp.1090-1104, 2000.

[5.50] P. J. Phillips. “The Facial Recognition Technology (FERET) Database,” http://www.itl.nist.gov/iad/humanid/feret/feret_master.html.

[5.51] P.J. Phillips, H. Wechsler, J. Huang, P. Rauss. “The FERET Database and Evaluation Procedure for Face Recognition Algorithms,” Image and Vision Computing, Vol. 16, No.5, pp. 295-306, 1998.

[5.52] A.M. Martinez and R. Benavente. “The AR Face Database,” http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html

[5.53] A.M. Martinez and R. Benavente. “The AR Face Database,” CVC Technical Report #24, 1998.

[5.54] D. Zhang. Palmprint Authentication, Kluwer Academic Publishers, USA, 2004.

[5.55] C. Aggarwal. “An Efficient Subspace Sampling Framework for High-dimensional Data Reduction, Selectivity Estimation, and Nearest-neighbor Search,” IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 10, pp. 1247-1262, 2004.

[5.56] C. Aggarwal. “On the Effects of Dimensionality Reduction on High Dimensional Similarity Search,” Proc. of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 256-266, 2001.

[5.57] C. Aggarwal, J. Han, J. Wang, P.S. Yu. “On High Dimensional Projected Clustering of Data Streams,” Data Min. Knowl. Discov, Vol. 10, No. 3, pp. 251-273, 2005.

[5.58] M. Balasubramanian, E.L. Schwartz, J.B. Tenenbaum, V. Silva and J.C. Langford. "The Isomap Algorithm and Topological Stability," Science, Vol. 295, pp. 7, 2002.

[5.59] G. Baudat and F. Anouar. "Generalized Discriminant Analysis Using a Kernel Approach," Neural Computation, Vol. 12, No. 10, pp. 2385-2404, 2000.

[5.60] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriengman. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 711-720, 1997.

[5.61] D. Cai, X. He, J. Han. "Document Clustering Using Locality Preserving Indexing," IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 12, pp. 1624-1637, 2005.

[5.62] V. Castelli, A. Thomasian, C. Li. "CSVD: Clustering and Singular Value Decomposition for Approximate Similarity Search in High-dimensional Spaces," IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 3, pp. 671-685, 2003.

[5.63] B. Cui, B. C. Ooi, J. Su, K. Tan. "Indexing High-dimensional Data for Efficient In-memory Similarity Search," IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 3, pp. 339-353, 2005.

[5.64] O. Egecioglu, H. Ferhatosmanoglu, U. Ogras. "Dimensionality Reduction and Similarity Computation by Inner-product Approximations," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 6, pp. 714-726, 2004.

[5.65] X. Fu and L. Wang. "Data Dimensionality Reduction with Application to Simplifying RBF Network Structure and Improving Classification Performance", IEEE Transactions on System, Man, Cybern, Part B - Cybernetics, Vol.33, No.3, pp. 399-409, 2003.

[5.66] K. Fukunaga. Introduction to Statistical Pattern Recognition, Second Edition, Academic Press, Inc, 1990.

[5.67] K. Fukunaga and J. Mantock. "Nonparametric discriminant analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 5, No. 6, pp. 671-678, 1983.

[5.68] D. Hand, H. Mannila, and P. Smyth. Principles of Data Mining, 2001.

[5.69] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang. "Face Recognition Using Laplacianfaces," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 3, pp 328-340, 2005.

[5.70] J.J. Hull. "A Database for Handwritten Text Recognition Research," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, No. 5, pp. 550-554, 1994.

[5.71] A.K. Jain, R.P.W. Duin, and J. Mao. "Statistical Pattern Recognition: a Review," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, pp. 4-37, 2000.

[5.72] M. Kirby and L. Sirovich. "Application of the KL Procedure for the Characterization of Human Faces," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, pp. 103-108, 1990.

[5.73] Z. Liang and P. Shi. "Uncorrelated Discriminant Vectors Using a Kernel Method," Pattern

Recognition, Vol. 38, pp. 307-310, 2005.

[5.74] A. Martinez, R. Benavente. "The AR Face Database," CVC Technical Report, No.24, 1998.

[5.75] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.R. Müller. "Fisher Discriminant Analysis with Kernels,'' Proc. IEEE Int'l Workshop Neural Networks for Signal Processing IX, pp. 41-48, 1999.

[5.76] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K.R. Müller. "Constructing Descriptive and Discriminative Nonlinear Features: Rayleigh Coefficients in Kernel Feature Spaces," IEEE Transactions Pattern Analysis and Machine Intelligence, Vol. 25, No. 5, pp. 623-628, 2003.

[5.77] S. Mika, G. Rätsch, B. Schölkopf, A. Smola, J. Weston, and K.R. Müller. "Invariant Feature Extraction and Classification in Kernel Spaces," Advances in Neural Information Processing Systems 12, Cambridge, Mass: MIT Press, 1999.

[5.78] P. Mitra, C.A. Murthy, C.A. and S.K. Pal. "Density based Multiscale Data Condensation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 6, pp. 734-747, 2002.

[5.79] S.T. Roweis and L. K. Saul. "Nonlinear Dimensionality Reduction by Locally Linear Embedding," Science, Vol. 290, pp. 2323-2326, 2000.

[5.80] B. Schölkopf and A. Smola. Learning with Kernels. Cambridge, Mass.: MIT Press, 2002.

[5.81] B. Schölkopf, A. Smola, and K.R. Müller. " Nonlinear Component Analysis as a Kernel Eigenvalue Problem,' Neural Computation, Vol. 10, No. 5, pp. 1299-1319, 1998.

[5.82] J.B. Tenenbaum, V. Silva and J. C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction," Science, Vol. 290, pp. 2319-2323, 2000.

[5.83] M. Turk and A. Pendland. "Eigenfaces for Recognition," J. Cognitive Neuroscience, Vol. 3, No.1, pp.71-86, 1991.

[5.84] V. Vapnik. The Nature of Statistical Learning Theory. New York: Springer, 1995.

[5.85] V. Vapnik. Statistical Learning Theory, Wiley, 1998.

[5.86] W. Wang, J. Yang. "Mining High Dimensional Data," Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers, Kluwer Academic Publishers, 2005.

[5.87] Yale Univ. Face Database. http://cvc.Yale.edu.

[5.88] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang. "Discriminant Analysis with Tensor Representation," Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, pp. 526-532, 2005.

[5.89] S. Yan, D. Xu, B. Zhang, H. Zhang. "Graph Embedding: a General Framework for

Dimensionality Reduction," Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, pp. 830-837, 2005.

[5.90] J. Yan, B. Zhang, N. Liu, S. Yan, Q. Cheng, W. Fan, Q. Yang, W. Xi, and Z. Chen. "Effective and Efficient Dimensionality Reduction for Large-scale and Streaming Data Preprocessing," IEEE Transactions on Data and Knowledge Engineering, Vol. 18, No. 3, pp. 320-333, 2006.

[5.91] J. Yang, X. Gao, D. Zhang, and J. Yang. "Kernel ICA: an Alternative Formulation and its Application to Face Recognition," Pattern Recognition, Vol. 38, pp. 1784-1787, 2005.

[5.92] J. Yang, A.F. Frangi, J.Y. Yang, D. Zhang, Z. Jin. "KPCA Plus LDA: a Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 2, pp 230-244, 2004.

[5.93] J. Ye, Q. Li, H. Xiong, H. Park, R. Janardan, and V. Kumar. "IDR/QR: an Incremental Dimension Reduction Algorithm via QR Decomposition," IEEE Transactions on Data and Knowledge Engineering, Vol. 17, No. 9, pp. 1208-1222, 2005.

[5.94] W. Yu, X. Teng, and C. Liu. "Discriminant Locality Preserving Projections: a New Method to Face Representation and Recognition," Proc. of Second Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp. 201-207, 2005.

[5.95] W. Zheng, L. Zhao, and C. Zou. "Foley-Sammon Optimal Discriminant Vectors Using Kernel Approach," IEEE Transactions on Neural Networks, Vol. 16, No.1, pp. 1-9, 2005.

[6.1] R.G. Gosden, A.P. Feinber. "Genetics and Epigenetics-nature's Pen-and-pencil Set," N. Engl. J. Med., Vol. 356, No. 7, pp. 697-705, 2007.

[6.2] R. Taggart, C. Starr. Biology: the Unity and Diversity of Life: Mutated Genes and Their Protein Products. Thompson Brooks, 2006.

[6.3] J.P. Masly, C.D. Jones, M.A.F. Noor, J. Locke, H.A. Orr. "Gene Transposition as a Cause of Hybrid Sterility in Drosophila," Science, Vol. 313, No.5792, pp.1448-1450, 2006.

[6.4] T.M. Geiman, K.D. Robertson. "Chromatin Remodelling, Histone Modifications, and DNA Methylation-how Does It All Fit Together?" Journal of Cellular Biochemistry, Vol. 87, No.2, pp. 117-125, 2002.

[6.5] M. Spivakov, A.G. Fisher. "Epigenetic Signature of Stem-cell Identity," Nature Reviews Genetics, Vol. 8, pp. 263-271, 2007.

[6.6] M. Fabian, A. Peter, O. Alexander, P. Christian. "Feature Selection for DNA Methylation based Cancer Classification. Bioinformatics, Vol.17: S157-S164, 2001.

[6.7] R. Das, N. Dimitrova, Z. Xuan, R.A. Rollins, F. Haghighi, J.R. Edwards, J. Ju, T.H. Bestor, M.Q. Zhang. "Computational Predictoin of Methylation Status in Human Genomic Sequences," PNAS, Vol.103, No.28, pp.10713-10716, 2006.

[6.8] M. Bhasin, H. Zhang, E.L. Reinherz, P.A. Reche. “Prediction of Methylated CpGs in DNA Sequences Using a Support Vector Machine,” FEBS Letters, Vol.20, pp. 4302-4308, 2005.

[6.9] P. Marjoram, et al.. “Cluster Analysis for DNA Methylation Profiles Having a Detection Threshold,” BMC Bioinformatics, Vol. 7, pp. 361, 2006.

[6.10] M. Bibikova, et al..“Human Embryonic Stem Cells Have a Unique Epigenetic Signature,” Genome Research, Vol.16, No.9, pp.1075-1083, 2006.

[6.11] M. Bibikova, et al.. “High-throughput DNA Methylation Profiling Using Universal Bead Arrays,” Genome Research, Vol. 16, No.3, pp.383-393, 2006.

[6.12] Qiagen Inc.. http://www.qiagen.com/.

[6.13] Protocol online. http://www.protocol-online.org/prot/Protocols/.

[6.14] Illumina Inc.. http://www.illumina.com/.

[6.15] R. Katso, K. Okkenhaug, K. Ahmadi, S. White, J. Timms, M.D. Waterfield. “Cellular Function of Phosphoinositide 3-kinases: Implications for Development, Homeostasis, and Cancer,” Annu. Rev. Cell Dev. Biol. Vol. 17, pp. 615-675, 2001.

[6.16] Y. Li, et al.. “Plasma Neuropeptide Y (NPY) Levels in Patients with Gastric and Colorectal Carcinomas,” Chinese Journal of Oncology, Vol. 20, No.3, pp. 213-215, 1998.

[6.17] S. Dudoit, J. Fridlyand, and T.P. Speed. “Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data,” J. Am. Statistical Assoc. Vol. 97, pp.77-87, 2002.

[6.18] E.J. Yeoh, et al.. “Classification, Subtype Discovery, and Prediction of Outcome in Pediatric Lymphoblastic Leukemia by Gene Expression Profilingk,” Cancer Cell, Vol.1, pp.133-143, 2002.

[6.19] J. Khan, et al.. “Classification and Diagnostic Prediction of Cancers Using Expression Profiling and Artificial Neural Networks,” Nature Medicine, Vol. 7, pp. 673-679, 2001.

[6.20] A.A. Alizadeh, et al.. “Distinct Types of Diffuse Large B-cell Lymphoma Identified by Gene Expression Profiling,” Nature, Vol. 403, pp. 503-511, 2002.

[6.21] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine. “Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays,” Proc. Nat’l Academy of Science, Vol. 96, pp. 6745-6750, 1999.

[6.22] D. Singh. “Gene Expression Correlates of Clinical Prostate Cancer Behaviour,” Cancer Cell, Vol. 1, No. 2, pp. 203-209, 2002.

[6.23] Waikato University. http://www.cs.waikato.ac.nz/ml/weka/.

[6.24] I. Nishimura, S. Shimizu, J.Y. Sakoda, K. Yoshikava. “Expression of Drosophila MAGE

Gene Encoding a Necdin Homologous Protein in Postembryonic Neurogenesis," Gene Expr. Patterns, Vol. 7, No. 3, pp. 244-251, 2007.

[6.25] K. Anja, et al.: Annemarie P. "The Systematic Functional Characterization of Xq28 Genes Prioritises Candidate Disease Genes," BMC Genomics, Vol. 7, pp.29, 2006.

[6.26] A.B. Isabelle, et al.. "Neural Differentiation of Mouse Embryonic Stem Cells in Chemically Defined Medium," Brain Research Bulletin, Vol. 68, pp.62-75, 2005.

[6.27] J.H. Kim, S. Park, M.R. Kang, S. Oh, T.H. Lee, M.T. Muller, I.K. Chung. "Ubiquitin Ligase MKRN1 Modulates Telomere Length Homeostasis Through a Proteolysis of hTERT," Genes & Development, Vol.19, pp. 776-781, 2005.

[6.28] S. Movafag, J.P. Hobson, S. Spiegel, H.K. Kleinman, Z. Zukowska. "Neuropeptide Y Induces Migration, Proliferation, and Tube Formation of Endothelial Cells," FASEB Journal, Vol. 20, pp. 1924-1926, 2006.

[6.29] Y. Shen, J. Chow, Z. Wang, G. Fan. "Abnormal CpG Island Methylation Occurs during in vitro Differentiation of Human Embryonic Stem cells,"Human Molecular Genetics, Vol. 15, No.17, pp. 2623-2635, 2006.

[6.30] J. Shan. Transcriptional Regulation of the Human Prostatic Acid Phosphatase Gene (Chapter 2), http://herkules.oulu.fi/isbn9514267621/html/x516.html.

[6.31] M. Castren, et al.. "Altered Differentiation of Neural Stem Cells in Fragile X Syndrome. PNAS, Vol.102, No.49, pp.17834-17839, 2005.

[6.32] B.J. Yoon, Herman H, Sikora A, Smith LT, Plass C, SolowayPD. "Regulation of DNA Methylation of Rasgrf1," Nature Genetics, Vol.30, pp.92-96, 1992.

[6.33] K. Takahashi, S. Yamanaka S. "Induction of Pluripotent Stem Cells from Mouse Embryonic and Adult Fibroblast Cultures by Defined Factors," Cell, Vol. 126, No.4, pp.663-76, 2007.

[6.34] S.M. Rowe, S. Miller, E.J. Sorscher. "Cystic Fibrosis," N Engl J Med, Vol. 352 No.19, pp.1992-2001, 2006.

[6.35] E. Fredrik, P. Fredrik, S. Goran. "Molecular Analyses of the Candidate Tumour Suppressor Gene PLAGL1 in Benign and Malignant Salivary Gland Tumours," European Journal of Oral Sciences, Vol. 112, No.6, pp.545-547, 2004.

[6.36] G. Sandra, J.S. Edward. "Analysis of Apoptosis-associated Genes and Pathways in Oral Cancer Cells," Journal of Oral Pathology and Medicine, Vol. 35, No.3, pp.146-154, 2006.

[6.37] E.J. Bryan, et al.. "Mutation Analysis of EP300 in Colon, Breast and Ovarian Carcinomas," International Journal of Cancer, Vol.102, No.2, pp.137-141, 2002.

[6.38] M. Thangaraju, S.H. Kaufmann, F.J. Couch. "BRCA1 Facilitates Stress-induced Apoptosis in Breast and Ovarian Cancer Cell Lines," J. Biol. Chem., Vol. 275, No.43, pp. 33487-33496, 2000.

[6.39] R. Kim, M. Emi, K. Tanabe, S. Murakami. "Role of the Unfolded Protein Response in Cell Death," Apoptosis, Vol.11, No.1, pp.5-13, 2006.

[6.40] A.V. Ivanov, et al.. Human ribosomal protein S26 suppresses the splicing of its pre-mRNA. Biochim Biophys Acta 2005, 1727(2): 134-140.

[6.41] C.X. Deng, R.H. Wang. "Roles of BRCA1 in DNA Damage Repair: a Link between Development and Cancer," Human Molecular Genetics, Vol. 12, No.1,pp. r113-r123, 2003.

[6.42] A. Klungland, T. Lindahl. Second Pathway for Completion of Human DNA base Excision-repair: Reconstitution with Purified Proteins and Requirement for DNase IV(FEN1)," EMBO Journal, Vol. 16, pp.3341-3348, 1997.

[6.43] J.R.G. Peter, C.F.S. Ann, A.P. Roger. "Epigenetic Status of Human Embryonic Stem Cells," Nature Genetics, Vol. 37, pp. 585-587, 2005.

[6.44] D. Alexander. "B-cell Lymphoma: Suppressing a Tumour Suppressor," Nature Medicine, Vol. 11, No. 22, 2005.

[6.45] A.H. Kat, et al.. "Absence of ST7 Mutations in Tumour-derived Cell Lines and Tumours," Nature Genetics, Vol. 29, pp.380-381, 2001.

[6.46] J. Li, Y. Liu, B. Ru. "Effect of Metallothionein on Cell Viability and Its Interactions with Cadmium and Zinc in HEK293 Cells," Cell Biol. Int., Vol.29, No.10, pp. 843-848, 2005.

[6.47] C. Heurteaux, et al.. "Deletion of the Background Potassium Channel TREK-1 Results in a Depression-resistant Phenotype. Nature Neuroscience, Vol.9, pp.1134-1141, 2006.

[6.48] M.S. Turker. "Gene Silencing in Mammalian Cells and the Spread of DNA Methylation," Oncogene, Vol. 21, No.35, pp.5388-5393, 2002.

[6.49] D.E. Bredesen, R.V. Rao, R. Mehlen. "Cell Death in the Nervous System," Nature Immunology, Vol. 443, pp.796-802, 2006.

[7.1] Marco Loog, Robert P. W. Duin, Reinhold Haeb-Umbach: Multiclass Linear Dimension Reduction by Weighted Pairwise Fisher Criteria. IEEE Trans. Pattern Anal. Mach. Intell. 23(7): 762-766 (2001)

[7.2] Gordon F. Hughes: On the Mean Accuracy of Statistical Pattern Recognizers. IEEE Trans. Information Theory. 14(1): 55-63 (1968)

[7.3] Yoshihiko Hamamoto, Shunji Uchimura, Shingo Tomita: On the Behavior of Artificial Neural Network Classifiers in High-Dimensional Spaces. IEEE Trans. Pattern Anal. Mach. Intell. 18(5): 571-574 (1996)

[7.4] Jianping Hua, Zixiang Xiong, Edward R. Dougherty: Determination of the optimal number of features for quadratic discriminant analysis via the normal approximation to the discriminant distribution. Pattern Recognition 38(3): 403-421 (2005)

[7.5] Klaus-Robert Müller, Sebastian Mika et al: An introduction to kernel-based learning algorithms. IEEE Trans. Neural Networks 12(2) (2001).

[8.1] T.S. Lee. “Image Representation Using 2D Gabor Wavelets,” IEEE Trans. Pattern Anal. Mach. Intell., Vol.18, No. 10, pp. 959-971, 1996.

Ben's publications

[1] Ben Niu, Simon C.K. Shiu: Using Similarity Measure to Enhance the Robustness of Web Access Prediction Model. Proc. International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES) 2005, pp. 107-111, (LNCS).

[2] Ben Niu, Simon C.K. Shiu: A Novel 3D Face Recognition Algorithm Using Template Based Registration Strategy and Artificial Neural Networks. Proc. International Conference of Pattern Recogntion and Machine Intelligence (PReMI) 2005, pp. 315-317, (LNCS).

[3] Ben Niu, Simon C.K. Shiu, Sankar K. Pal: Two Dimensional Laplacianfaces for Face Recognition. Proc. Rough Sets and Current Trends in Computing (RSCTC) 2006, pp. 852-861, (LNAI).

[4] Ben Niu, Qiang Yang, Jinyan Li, Hannah Hong Xue, Simon C.K. Shiu, Huiqing Liu, Sankar K. Pal: Pattern Detection and Co-methylation Analysis of Epigenetic Features in Human Embryonic Stem Cells. ISCB International Conference on Bioinformatics (InCoB) 2007, (accepted).

[5] Ben Niu, Qiang Yang, Jinyan Li, Simon C.K. Shiu, Sankar K. Pal: Discovering patterns of DNA Methylation: Rule mining with Rough Sets and Decision Trees, and Comethylation analysis, International Conference of Pattern Recogntion and Machine Intelligence (PReMI) 2007, pp. 389-397, (LNCS).

[6] Jian Yang, David Zhang, Jingyu Yang, Ben Niu: Globally Maximizing, Locally Minimizing: Unsupervised Discriminant Projection with Applications to Face and Palm Biometrics. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 2007, Vol. 29(4), pp. 650-664.

[7] Ben Niu, Qiang Yang, Simon C.K. Shiu, Sankar K. Pal: Two Dimensional Laplacianfaces for Face Recognition, (Journal of Pattern Recognition, accepted in press).

[8] Ben Niu, Minghu Ha, Xizhao Wang: Design and Implementation of a Multi-agent System based on VIP and VC. Journal of Hebei University (Natural Science Edition), issue 2, 2002.

[9] Ben Niu, Minghu Ha, Xizhao Wang: Object Oriented Knowledge Representation based on VC and VIP. Journal of Hebei University of Science and Technology, Issue 1, 2002.