Sensors **2008**, *8*, 4186-4200; DOI: 10.3390/s8074186

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.org/sensors

Article

Kernel Based Nonlinear Dimensionality Reduction and Classification for Genomic Microarray

Xuehua Li * **and Lan Shu**

School of Applied Mathematics, University of Electronic Science and Technology of China, Chengdu, 610054, P.R. China.

* Author to whom correspondence should be addressed; E-mail: leesoftcom@gmail.com

Received: 4 June 2008; in revised form: 17 June 2008 / Accepted:6 July 2008 / Published: 15 July 2008

Abstract: Genomic microarrays are powerful research tools in bioinformatics and modern medicinal research because they enable massively-parallel assays and simultaneous monitoring of thousands of gene expression of biological samples. However, a simple microarray experiment often leads to very high-dimensional data and a huge amount of information, the vast amount of data challenges researchers into extracting the important features and reducing the high dimensionality. In this paper, a nonlinear dimensionality reduction kernel method based locally linear embedding(LLE) is proposed, and fuzzy K-nearest neighbors algorithm which denoises datasets will be introduced as a replacement to the classical LLE's KNN algorithm. In addition, kernel method based support vector machine (SVM) will be used to classify genomic microarray data sets in this paper. We demonstrate the application of the techniques to two published DNA microarray data sets. The experimental results confirm the superiority and high success rates of the presented method.

Keywords: Manifold learning; Dimensionality reduction; Locally linear embedding; Kernel methods; Support vector machine.

1 Introduction

The recent sequencing of the human genome has opened a new era in biomedical research; genomic microarray data have attracted a great deal of attention, as reflected by the ever increasing number of publications on this technology in the past decade. The application of microarrays technology encom-

passes many fields of study. From the search for differentially expressed genes, genomic microarrays data present enormous opportunities and challenges for machine learning, data mining, pattern recognition, and statistical analysis, among others. In particular, microarray technology is a rapidly maturing technology that provides the opportunity to assay the expression levels of thousands or tens of thousands of genes in a single experiment[1]. Nevertheless, microarrays experiments usually produce a huge amount of data and high dimensionality in relatively small sample sizes (commonly on the order of tens or hundreds). Hence, the biggest challenge of microarrays experiments is data mining and dimensionality reduction. Manifold learning is a perfect tool for data mining that discovers the structure of high dimensional data sets and provides better understanding of the data. Several different manifold learning algorithms have been developed to perform dimensionality reduction of low-dimensional nonlinear manifolds embedded in a high dimensional space. Isomap[2], LLE[3], Laplacian eigenmaps, and Stochastic neighbor embedding were originally proposed as a generalization of multidimensional scaling.

The LLE is considered as among one of the most effective dimensionality reduction algorithms for data preprocessing of high-dimensional data and streaming, and has been used to solve various problems in information processing, pattern recognition, and data mining[4–6]. LLE algorithm computes a different local quantity, and calculates the best coefficients to approximate each point by a weighted linear combination of its neighbors, and then tries to find a set of low-dimensional points, which can be linearly approximated by its neighbors with the same coefficients that have been determined from high-dimensional points. However, when LLE is applied to real world datasets and applications, it displays limitations, such as sensitivity to the noise, outliers, missing data, and poor linear correlation between variables due to poorly distributed variables. In LLE algorithms, the free parameter is the LLE's neighborhood size, which unfortunately, has no direct method of finding the optimal parameter. The optimal neighborhood size for each problem is determined by the experimenter's experience. On the other hand, if the density of training data is uneven, it will decrease the precision of classification if only the sequence of first k nearest neighbors is considered and not the differences of distances.

The purpose of this paper is to fill these gaps by presenting a kernel method based LLE algorithm(KLLE). The kernel method[7, 8] is demonstrated as having the ability to extract the complicated nonlinear information from application datasets. The kernel function of the kernel method is a nonlinear mapping from input space $\mathbb{X} \subseteq \Re^n$ onto feature space $\mathbb{H} \subseteq \Re^N$, $\varphi : \mathbb{X} \subseteq \Re^n \rightarrow \mathbb{H} \subseteq \Re^N$. The kernel method provides a powerful and principled way of detecting nonlinear relations using well-understood linear algorithms in an appropriate feature space. This approach decouples the design of the algorithm from specification of the feature space. Most importantly, based on the kernel method, the kernel matrix is guaranteed to be positive semi-definite, convenient for the learning algorithm receiving information about the feature space and input data, and projects data onto an associated manifold, such as PCA. In addition, to solve KNN's parameter problems, fuzzy KNN adopts the theory of fuzzy sets to KNN, and fuzzy KNN assigns fuzzy membership as a function of the object's distance from its K-nearest neighbors and the memberships in the possible classes. This combination has two advantages. Firstly, fuzzy KNN can denoise training datasets. And secondly, the number of nearest neighbors selection, though not the most important, can consider the neighbor's fuzzy membership value.

Recently, support vector machine(SVM) has been extensively used by the machine learning commu-

nity because it effectively deals with high dimensional data, provides good generalization properties, and defines the classifier architecture in terms of the so-called support vectors [8]. The theory of SVM is based on the idea of structural minimization, which shows that the generalization error is bounded by the sum of the training set and a term depending on the Vapnik-Chervonenkis dimension. By minimizing this bound, high generalization performance can be achieved. Moreover, unlike other machine learning methods, SVM generalization error is not related to the problem's input dimensionality.

This paper focused on genomic microarray analysis, which enables researchers to monitor the expression levels of thousands of genes simultaneously[9]. With the help of gene expressions, heterogeneous cancers can be classified into appropriate subtypes. To classify tissue samples or diagnose diseases based on gene expression profiles, both classic discriminant analysis and contemporary classification methods have been used and developed. Recently, different kinds of machine learning and statistical methods[10, 11] have been used to classify cancers using genomic microarrays expression data. To evaluate the effectiveness of the proposed KLLE dimensionality reduction method for classification, two published datasets are used. The experiment shows that dimensionality reduction of genes can significantly increase classification accuracy.

The remainder of this paper is organized as follows. In Section 2, we introduce the kernel method. The kernel method based LLE algorithm is constructed in Section 3. In Section 4, the kernel method based SVM is introduced. In section 5, we apply our proposed dimensionality reduction method to the Lymphoma and the SRBCT genomic microarray data sets, experiments and comparisons are conducted and presented. Conclusions are drawn in the final section.

2 Summary of Kernel Method

The kernel method[7] has become one of the most popular approaches to learning from examples with many potential applications in science and engineering [12]. The kernel method has been demonstrated to be able to extract the complicated nonlinear information embedded on a dataset. Many algorithms for data analysis are based on the assumption that the data can be represented as vectors in a finite dimensional vector space, such as linear discrimination, PCA, or least squares regression, making extensive use of the linear structure. Roughly speaking, the kernel method allows natural derivations of nonlinear versions. The general idea is described as follows. Given a linear algorithm (i.e., an algorithm which works in a vector space), one first maps the data living in a space $\mathbb{X}$ (the input space) to a vector space $\mathbb{H}$ (the feature space) via a nonlinear mapping $\psi : \mathbb{X} \subseteq \Re^n \rightarrow \mathbb{H} \subseteq \Re^N$, the kernel function is the form $K(x_i, x_j) = \langle \psi(x_i), \psi(x_j) \rangle$, and the kernel matrix is $K = (K_{ij}) = (K(x_i, x_j))$, respectively. Then, linear algorithms may be applied to the vector representation $\psi(x)$ of the data, which performs nonlinear analysis of data by linear method. In other words, the kernel method is an attractive computational shortcut, the purpose of the mapping $\psi(\cdot)$ is to translate nonlinear structures of data into new linear representation in $\mathbb{H}$.

The kernel methods solution comprises two parts: a module that performs the mapping into the embedding or feature space and a learn algorithm designed to discover linear patterns in that space. Firstly, we need to create a complicated linear feature space, and then work out what the inner product in that space would be, and finally find a direct method for computing that value in terms of the original inputs.

In fact, the kernel function K is directly defined by the nonlinear mapping $\psi(\cdot)$, and the feature space $\mathbb{H}$ is simply derived from its definition. The main property of kernel function is that the fundamental concept of the kernel method is the deformation of the vector (lower) space itself to a higher dimensional space. In general, a higher dimension linear space is clearer to classify than a low dimension one.

However, an explicit mapping $\psi(\cdot)$ does not always exist, and kernel method's conditions are not sufficient in guaranteeing the existence of a feature space. In practice, the mapping is performed implicitly by choosing a suitable kernel function $K(x_i, x_j) = \langle \psi(x_i), \psi(x_j) \rangle$ for the data points x_i and x_j . Moreover, there is a problem when choosing the function $K(x_i, x_j)$, since not every function is guaranteed to give a valid feature space. One way of searching for a valid kernel function is to draw on Mercer's theorem [13] which states that any continuous symmetric function $K(x_i, x_j)$ that satisfies the positive semi-definite condition

$$\int\int_{\mathbb{X}\times\mathbb{X}} K(x_i, x_j)\psi(x_i)\psi(x_j)dx_i dx_j \geq 0 \qquad and \qquad \int_{\mathbb{X}} \psi(x)^2 dx < \infty \tag{1}$$

which is ensured to be a kernel for some valid feature space. This provides a flexible way of choosing the kernel mapping functions.

The kernel matrix is taken as an information bottleneck, which follows from the fact that the learning algorithm can glean from the training data and the chosen feature space is contained in the kernel matrix. The kernel matrix is not only the central concept in the design and analysis of the kernel method, but can also be regarded as the central data structure in their implementation. It is perhaps not surprising that some properties of the kernel matrix can be used to assess the performance of a learning algorithm.

3 Kernel Method based LLE Algorithm for Dimensionality Reduction

3.1 Locally Linear Embedding

LLE[3] is a manifold learning method that has aroused a great deal of interest in machine learning. It computes low-dimensional, neighborhood-preserving embeddings of high-dimensional inputs and recovers the global nonlinear structure from locally linear fits. Essentially, the algorithm attempts to compute a low dimensional embedding with the property that nearby points in the high dimensional space remain nearby and similarly co-located with respect to one another in the low dimensional space. Put another way, the embedding is optimized to preserve the local configurations of nearest neighbors. LLE computes dimensionality reduction that preserves the local neighborhood structure of the input data in the low-dimensional transformation. The transformation models the subspace manifold as a connected patchwork of locally linear surfaces. LLE is commonly justified using Taylor's theorem which states that any differentiable function is linear at the limit in a small area around a point. LLE works by identifying local neighborhood distance relationships, and by finding a mapping into a lower dimensionality that preserves them as much as possible. The selection of k value is the key to dimensionality reduction. There have been numerous papers [14, 15] suggesting that the selection of the neighborhood number k is important to the original LLE. If the number k is larger, the algorithm will ignore or even lose the local nonlinear features on the manifold, just as the traditional PCA performs. In contrast, if the number k is defined as smaller, LLE will split the continuous manifold into detached locality pieces, because the

global characteristics are lost. On the other hand, it is well known that LLE is sensitive to noise, and LLE cannot preserve well the local geometry of the data manifolds in the embedding space when there are outliers in the data. In general, the practical input data sets are usually contaminated by noise which are caused by the disturbance and measurement, and the data sets are nonlinear correlation. Thus, the selection of the neighborhood number is difficult in real application to datasets.

3.2 Fuzzy K-Nearest Neighbor Algorithms

Conventional fuzzy KNN algorithm assigns an unlabeled pattern x to the class which appears the most among its k nearest labeled neighbors. The algorithm is described as follows. The problem of classifying N entities into M classes can be formulated as $C = \{c_1, c_2, ..., c_M\}$, where c_i denotes the ith class. The available information is assumed to be in a training data set $\Omega = \{(x_1, c_i), (x_1, c_k), ..., (x_n, c_j)\}$ of n patterns x_i and their corresponding class labels c_i taking values from C. The KNN[16] rule is well known in the pattern recognition literature. According to the rule, an unclassified pattern x is assigned to the class represented by a majority of its k nearest neighbors in Ω. [17] proposed a new approach by combining the fuzzy set theory with KNN algorithm, and named it as the fuzzy KNN classifier algorithm. According to his approach, rather than individual classes as in KNN, the fuzzy memberships of samples are assigned to different categories according to the following formulation

$$\mu_i(x) = \frac{\sum_{i=1}^{k} \mu_{ij} \|x - x_j\|^{2/m-1}}{\sum_{i=1}^{k} \|x - x_j\|^{2/m-1}} \tag{2}$$

where k is the number of nearest neighbors, m determines how heavily the distance is weighted when calculating each neighbor's contribution to the membership value, $\mu_i(x)$ denotes the membership of the test pattern x to class i, $\|x - x_j\|$ is the distance between the test pattern x and its nearest training samples x_j. In this paper, the Euclidean metric is used, and μ_{ij} is the fuzzy membership value of the jth neighbor to the ith class. After calculating all the memberships for a query sample, it is assigned to the class with which it has the highest membership value. Fuzzy KNN algorithms have two main advantages over the traditional KNN algorithms. Firstly, while determining the class of the current residue, the algorithm is capable of taking into consideration the ambiguous nature of the neighbors if any. The algorithm has been designed such that these ambiguous neighbors do not play a crucial role in the classification of the current residue. The second advantage is that residues are assigned a membership value in each class rather than binary decision of 'belongs to' or 'does not belong to'. The advantage of such assignment is that these membership values act as strength or confidence with which the current residue belongs to a particular class.

3.3 Kernel Method based LLE Algorithm

The KLLE extends LLE to work with the kernel method, which is used to map the nonlinear data into the linear feature space, which best reconstructs it as a linear combination of its neighbors. Moreover, the kernel matrix is positive semi-definite, and some properties and eigen-decomposition of kernel matrix

are used to optimize the KLLE's objective function. In another way, the larger candidate neighborhood number k is selected over the original LLE, and fuzzy KNN is used to calculate all the fuzzy memberships for the candidate neighborhood. It is then assigned to the new neighborhood set when it has the higher membership value, and the new neighborhood number $\bar{k}$ is obtained. The major modifications of KLLE to the original LLE algorithm are discussed below.

Step 1. Mapping. Let $\mathbb{X} = \{x_1, x_2, ..., x_n\}$ be a set of n points in a high-dimensional data space $\Re^D$. Suppose that the space $\mathbb{X}$ is mapped into a Hilbert space $\mathbb{H}$ through a nonlinear mapping function $\psi : \mathbb{X} \subseteq \Re^D \rightarrow \mathbb{H} \subseteq \Re^N$.

Step 2. The fuzzy neighborhood for each point. Assign neighbors to each data point $\psi(x_i)$ using the Fuzzy KNN algorithm. The $\bar{k}$ closest neighbors $\{\psi(x_i^j); j = 1, ..., \bar{k}\}$ are selected using the new define fuzzy Euclidean distance measure $\|\psi(x_i^j) - \psi(x_i)\|_F$, as follows:

$$\|\psi(x_i^j) - \psi(x_i)\|_F^2 = [\psi(x_i^j) - \psi(x_i)]^T V_i^{-1} [\psi(x_i^j) - \psi(x_i)] \tag{3}$$

where V_i is a fuzzy covariance matrix of the point x_i, and V_i is a symmetric and positive definite matrix, which specifies the shape of the clusters. The matrix V_i is commonly selected as the identity matrix, leading to Euclidean distance and, consequently, to spherical clusters, and and V_i is defined as

$$V_i = \frac{\sum_{j=1}^{k} \mu_{ij}^2 [\psi(x_i^j) - \psi(x_i)][\psi(x_i^j) - \psi(x_i)]^T}{\sum_{t=1}^{n} \mu_{ij}^2} \tag{4}$$

Step 3. The kernel method based manifold reconstruction error. The KLLE's reconstruction error is similar to those of LLE, which is measured by cost function:

$$J(W) = \sum_{i=1}^{N} \|\psi(x_i) - \sum_{j=1}^{\bar{k}} W_{ij} \psi(x_i^j)\|^2 \tag{5}$$

Considering reconstruction weights $\sum_{j=1}^{n} W_{ij} = 1$, the reconstruction error can be rewritten by

$$J(W) = \sum_{i=1}^{N} \| \sum_{j=1}^{\bar{k}} [\psi(x_i) - W_{ij} \psi(x_i^j)] \|^2 = \sum_{i=1}^{N} \| \sum_{j=1}^{\bar{k}} W_{ij} [\psi(x_i) - \psi(x_i^j)] \|^2 = \sum_{i=1}^{N} J(W_i) \tag{6}$$

$$J(W_i) = \| \sum_{j=1}^{\bar{k}} W_{ij} [\psi(x_i) - \psi(x_i^j)] \|^2 = \|QW_i\|^2 = W_i^T Q^T Q W_i = W_i^T K W_i \tag{7}$$

where $Q_i = [\psi(x_i) - \psi(x_i^1), \psi(x_i) - \psi(x_i^2), ..., \psi(x_i) - \psi(x_i^{\bar{k}})]$; it is obvious that $Q^T Q$ is a positive semi-definite matrix. Then $K = Q^T Q$ is defined as a kernel matrix. Hence Eq.(7) which is subjected to $W_i^T \mathbf{1} = 1$ can be cast as the following Lagrange formulation

$$L(W_i) = W_i^T K W_i - \lambda (W_i^T I \mathbf{1} - 1) \tag{8}$$

where the solution of Eq.(8) is $W_i = K^{-1}\mathbf{1} / \mathbf{1}^T K^{-1} \mathbf{1}$, K is a positive definite matrix, the eigen-decomposition of K is of the form $K = U^T \Lambda U$, then $W_i = U^T \Lambda^{-1} U \mathbf{1} / \mathbf{1}^T U^T \Lambda^{-1} U \mathbf{1}$. Hence, the reconstruction weights W are computed by kernel matrix's eigenvalues and eigenvectors.

Step 4. The kernel method computes low-dimensional embedding Y. In this step, KLLE is used to compute the best low-dimensional embedding Y based on the weight matrix W obtained.

$$\Phi(Y) = \sum_{i=1}^{N} \|y_i - \sum_{j=1}^{\bar{k}} W_{ij} y_i^j\|^2 = tr(Y^T M Y) \tag{9}$$

subject to the constraints $\sum_{i=1}^{N} y_i = 0$ and $\frac{1}{N}\sum_{i=1}^{N} y_i^T y_i = I$. Where $M = (I - W)^T(I - W)$, in LLE algorithm, the LLE embedding is given by the d eigenvectors correspond to the d smallest non-zero eigenvalues of matrix M[18].

In this step, we propose a method to yield KLLE embedding. M is a positive definite matrix, which has a maximum eigenvalue λ_1, and the smallest eigenvalue is **0** and the corresponding eigenvector is the uninform vector $\mathbf{1} = (1, 1, ..., 1)^T$. Since the other eigenvectors are orthogonal to **1** and their coefficient sum to 0. Defining matrix N: $N = (\lambda_1 I - M)$, it is obvious that N a positive definite matrix. If we compute the eigen-decomposition of N, the leading eigenvector is **1**, and the coordinates of the eigenvectors $2, 3, ..., d+1$ provide the KLLE embedding. Equivalently, defining a new kernel matrix K

$$K = (I - \mathbf{1}\mathbf{1}^T)N(I - \mathbf{1}\mathbf{1}^T) = (I - \mathbf{1}\mathbf{1}^T)[\lambda_1 I - (I - W)^T(I - W)](I - \mathbf{1}\mathbf{1}^T) \tag{10}$$

then, the eigenvectors $1, 2, ..., d$ of K provide the KLLE embedding in $\Re^d$.

The KLLE algorithm finds some global coordinates x_i in the kernel space, over the lower-dimensional manifold, that conserve the local relations between neighboring points in the original embedding space. Each individual coordinate is obtained only from local information within its neighborhood. The overall KLLE algorithm only involves searching for closest points and basic matrix manipulations by kernel method and kernel matrix.

4 Kernel Method based SVM Classifier

The original support vector machine can be characterized as a powerful learning algorithm based on recent advances in statistical learning theory [19]. SVM is a learning system that uses a hypothesis space of linear functions in a high-dimensional space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory. SVM uses a linear model to implement nonlinear class boundaries by mapping input vectors nonlinearly into a high-dimensional feature space using kernels. SVM has recently become one of the most popular tools for machine learning and data mining and can perform both classification and regression.

Let $S = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ be a training set with input data, where $x_i \in \Re^n$ is the training data and corresponding binary class labels $y_n \in \{-1, 1\}$. Let the weight and the bias of the separating hyperplane be w and b, respectively, and the SVM classifier is

$$f(x_i) = sgn(w\varphi(x_i) + b) \tag{11}$$

where φ is a nonlinear function, which maps x the input space into a feature space, To separate the data linearly in the feature space, the decision function satisfies the following conditions and the optimization problem is

$$\begin{array}{ll} \textit{Minimize} & \|w\|^2 =< w, w > \\ \textit{subject to} & y_i[w\varphi(x_i) + b] \geq 1, \qquad i = 1, 2, ..., n \end{array} \tag{12}$$

The objective of SVM is to maximize the margin of separation and minimize the training errors. The problem can then be transformed into the following Lagrange formulation

$$\begin{array}{l} \textit{maximize } L(\alpha) = \sum\limits_{i=1}^{n} \alpha_i - \frac{1}{2} \sum\limits_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \textit{subject to } \sum\limits_{i=1}^{n} y_i \alpha_i = 0 \qquad \alpha_i \geq 0, \qquad i = 1, 2, ...n \end{array} \tag{13}$$

where $K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$ is a kernel function and satisfies the Mercer theorem. The Karush-Kuhn-Tucker(KKT) complementarity conditions [19] provide useful information about the structure of the solution. The conditions state that the optimal solutions α^*, w^*, b^* must satisfy

$$\alpha_i^*[y(w^*\varphi(x_i) + b^*) - 1] = 0, \qquad i = 1, 2, ...n. \tag{14}$$

where the α_i^* are the solutions of the dual problem and are non-zero only for a subset of vectors x_i^* called support vectors. Then the resulting SVM for function estimation becomes

$$f(x) = sgn(\sum_{i=1}^{m} \alpha_i^* y_i K(x, x_i^*) + b) \tag{15}$$

where m is the number of support vectors. SVM technique, developed by Vapnik [20], is a powerful widely used technique for solving supervised biological classification problems due to its generalization ability[21]. In essence, SVM classifiers maximize the margin between training data and the decision boundary, which can be formulated as a quadratic optimization problem in a feature space. The subset of patterns that are closest to the decision boundary are called support vectors. More details about SVM could be found in Vapnik's [19, 21, 22] and other publications[23, 24].

5 Performance Evaluation

The wide use of microarrays is in classification–for example, the prediction of the phenotype of a biological sample based on its patterns of gene expression. The analysis of gene expression profiles, which serve as molecular signatures for cancer classification and identification of differentially expressed groups of genes, provides a high-level view of functional classes or pathways, and has become challenge and significantly affecting topics in bioinformatics research. In order to do this, one needs a 'training set' of samples that have well-defined phenotypic differences, and that can be used to generate reproducible profiles. There is a wide range of algorithms that have been used for classification, artificial neural networks [25], discriminant analysis[26], classification and regression trees, support vector machines, and a host of other applications. Essentially, each of these uses an original set of samples, or training set, to develop a rule that takes a new test sample from a test set and uses its expression vector sample, trimmed to a previously identified set of classification genes, to place this test sample into the context of the original sample set, thus identifying its class.

In this section, we evaluate the performance of our kernel based dimensionality reduction algorithms and classifier to two published DNA microarray data sets: one is small round blue cell tumors(SRBCTs) dataset, and the other is lymphoma dataset. The manipulation is described as follows, in two steps: The first step, applying KLLE to project the training sets from D-dimensional space to the embedded lower-dimensional d-dimensional space. In the next step, the new d-dimensional dataset will be classified by employing SVM and some comparisons as presented.

5.1 SRBCT Data

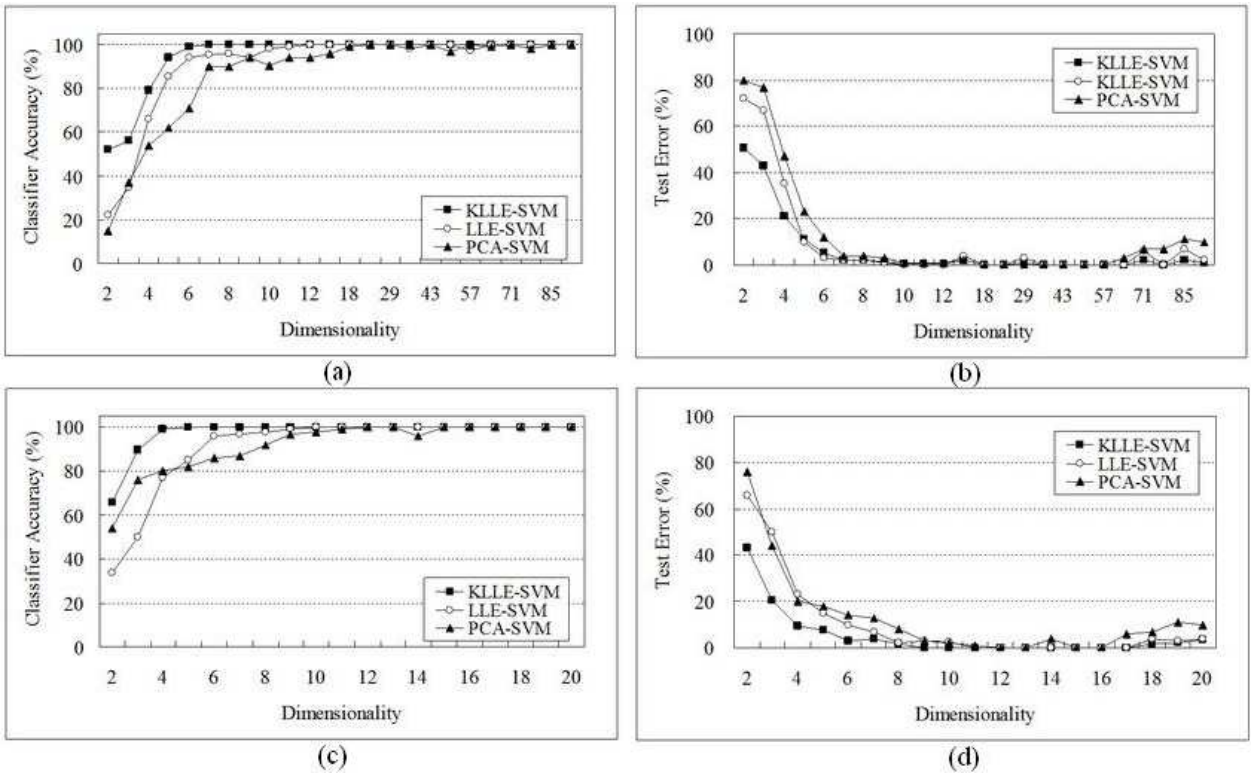

Figure 1. The experiments on the SRBCTs dataset: (a)The classifier accuracy of dimensionality reduction in 96 genes selected by Khan ;(b)The test error of dimensionality reduction in 96 genes; (c)The classifier accuracy of dimensionality reduction in 20 genes;(d)The test error of dimensionality reduction in 20 genes selected by Nikhil.

In the first series of computational experiments, we considered a data set on SRBCTs presented in the work of [27]. This study included data from a 6567 element array of the 88 samples, tested over a training set of 63 samples and 25 test samples. These data consisted of 63 samples categorized into four classes: 23 Ewing family of tumors(EWS), 20 rhabdomyosarcoma(RMS), 12 neuroblastoma(BL), and 8 non-Hodgkin lymphoma(NB) , which are represented by the expression values of 2308 genes with suspected roles in processes relevant to these tumors. The test set consists of 25 samples: 6 for EWS, 3 for BL, 6 for NB, 5 for RMS, and 5 non-SRBCTs: 2 normal muscle tissues and 3 cell lines including an undifferentiated sarcoma, an osteosarcoma, and a prostate carcinoma. The 96 genes were selected by [27] from total data set of 6567 genes by a method using artificial neural networks to best distinguish the

Table 1. The comparison of three methods on SRBCT dataset

Algorithms	96 genes			20 genes		
	Dimensional	Support vectors	Time(sec)	Dimensional	Support vectors	Time(sec)
SVM	96	-	-	20	106	4127
PCA-SVM	29	87	2672	11	64	1933
LLE-SVM	14	63	2102	9	42	1743
KLLE-SVM	7	42	1934	5	31	1307

four groups in question.

In this paper, considering some genes are irrelevant for diagnosis and would degrade the performance of the classifier, we followed Khan's and [28]'s methods for gene selection. We performed KLLE on the Khan's data set, consisting of expression levels of 96 genes, and the Nikhil's data set, only 20 genes based on FSMLP with online gene selection. In order to evaluate the dimensionality reduction, the comparisons between KLLE and principal component analysis(PCA) are done in the experiments. PCA is a standard dimensionality reduction tool, one rotates gene space, such that the variance is dominated by as few linear combinations as possible. KLLE, LLE and PCA were used to reduce the inputs to the dominant for classification. Furthermore, Gaussian kernel SVM is adopted as classifiers, and the kernel function in the analysis with parameters $\sigma = 0.16$ and $C = 4$. Using these 96 genes selected by [27], Fig.1 (a) shows the SVM classifier accuracy of the 63 training samples, and Fig.1 (b) shows the test errors of the 25 test samples. Using these 20 genes based on FSMLP with online gene selection by Nikhil, Fig.1 (c) and Fig.1 (d) show the SVM classifier accuracy and test errors of 88 samples. And all samples which are reduced to low-dimensional by KLLE, LLE and PCA, respectively.

By our purposed method, the classifier accuracy was 100% when the 20 genes were reduced to at least 5 dimensionality space, and 96 genes were reduced to at least 7 dimensionality space. However, by LLE method, the classifier accuracy was 100% when the 20 genes were reduced to at least 9 dimensionality space, and 96 genes were reduced to at least 14 dimensionality space. Finally, PCA method, the classifier accuracy was 100% when the 20 genes were reduced to at least 11 dimensionality space, and 96 genes were reduced to at least 29 dimensionality space. The implementation (classifier accuracy was 100%) returned a ranked list in about 1307 sec for the SRBCTs, 1743 sec by LLE and 1933 sec by PCA, much faster than 4127 sec by SVM without dimensionality reduction. Table 1 reports the comparison of the four methods on the SRBCTs dataset.

In fact, although the previous studies showed that linear classifiers are good enough to achieve almost perfect classification[29], there have some reports that the worst performances of the PCA based solution conform the need to take into account also nonlinear structures particularly for the SRBCTs dataset. We learned that the kernel method is more effective for classifier problems because of lying on non-linear separable feature space shown in most of cases. Using the 5 dimensionality vectors which were reduced from 20 genes, the KLLE-SVM classifier was able to correctly classify 25 test examples. Zero error

occured and no misclassified example in the blind test was identified; therefore, the result is comparable with the works of [11] and [29].

5.2 Lymphoma Data

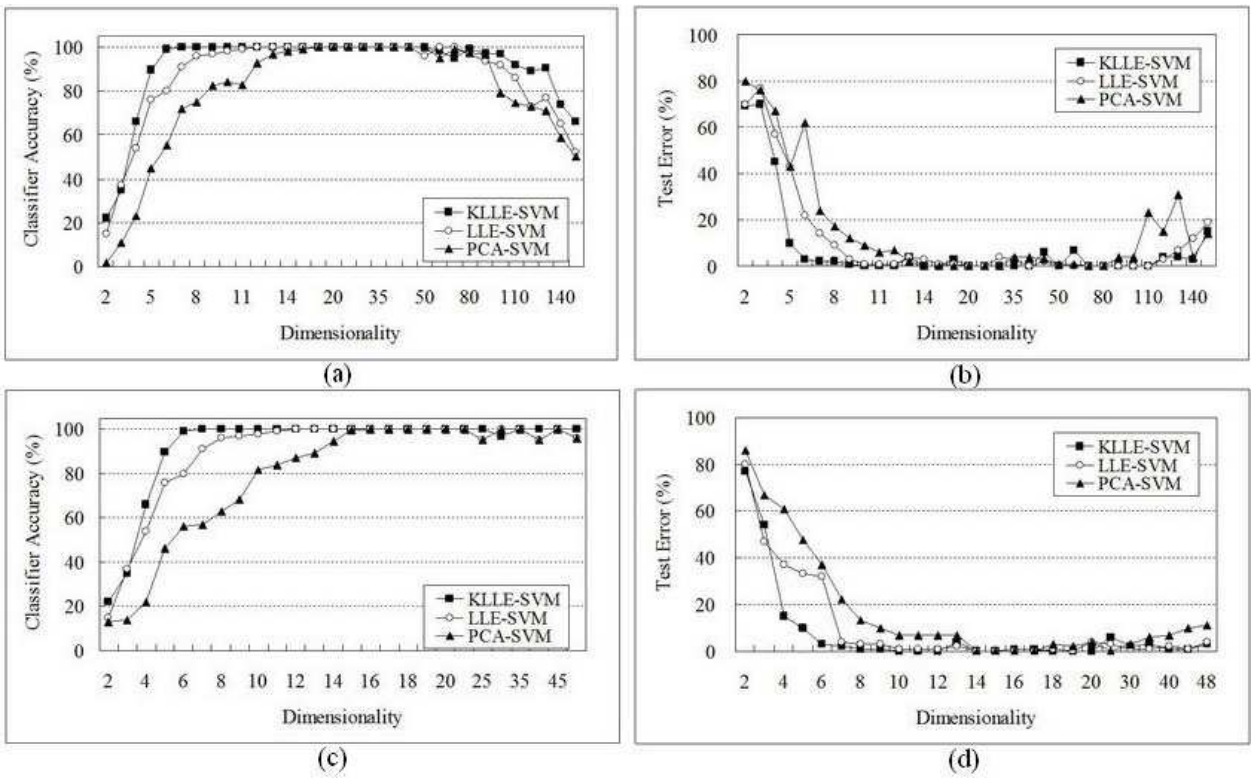

Figure 2. The experiments on the lymphoma dataset: (a)The classifier accuracy of dimensionality reduction in 165 genes selected by T-score;(b)The test error of dimensionality reduction in 165 genes; (c)The classifier accuracy of dimensionality reduction in 48 genes;(d)The test error of dimensionality reduction in 48 genes selected by nearest shrunken centroids

The second data set includes samples originating from the lymphoma dataset[Alizadeh et al], which can be obtained from http://llmpp.nih.gov/lymphoma. In this data set, there are 42 samples derived from diffuse large B-cell lymphoma (DLBCL), 9 samples from follicular lymphoma (FL), and 11 samples from chronic lymphocytic lymphoma (CLL). The entire data set includes the expression data of 4026 genes. In this data set, a small part of the data is missing. A k-nearest neighbor algorithm was applied to fill the missing values[30].

To find the genes that contribute most to the classification, the T-test, which has been used in gene selection[31] was used is to measure how large the difference is between the distributions of two groups of samples. To select important genes using the T-test involves two steps. In the first step, a score based on the T-test is calculated for each gene. In the second step, all the genes are rearranged according to their T-scores, and so on. The gene with the largest T-score is put in the first place of the ranking list, followed by the gene with the second greatest T-score. In this paper, in the lymphoma dataset, the top

165 genes selected from the lymphoma dataset by T-test. Another gene selection on lymphoma data is the nearest shrunken centroids[32] which used 48 genes to give a 100% accurate classification.

Table 2. The comparison of three methods on lymphoma dataset

Algorithms	165 genes			48 genes		
	Dimensional	Support vectors	Time(sec)	Dimensional	Support vectors	Time(sec)
SVM	165	-	-	48	124	5343
PCA-SVM	18	104	2672	22	83	3105
LLE-SVM	15	74	2133	9	56	2247
KLLE-SVM	7	56	1934	5	41	1766

We followed the same procedure as we did in the SRBCT dataset. We performed KLLE, LLE, and PCA on the dataset which selected by T-score, consisting of expression levels of the top 165 genes, and the Tibshirani's data set, of which only 48 genes were selected based on the nearest shrunken centroids for gene selection. Fig.2 shows the classifier accuracy and the testing errors happened during classification. Using these 165 genes selected by T-test, Fig.2 (a) shows the SVM classifier accuracy of the training samples, and Fig.2 (b) shows the test errors of the test samples. Using these 48 genes selected by nearest shrunken centroids, Fig.2 (c) and Fig.2 (d) show the SVM classifier accuracy and test errors of the 42 samples, and all samples which are reduced to low-dimensional by KLLE, LLE and PCA, respectively.

For high-dimensionality reduced by the KLLE method, the KLLE-SVM classifier accuracy was 100% when the 48 genes were reduced to at least 7 dimensionality space, and 165 genes were reduced to at least 10 dimensionality space. However, by LLE method, the classifier accuracy was 100% when the 48 genes were reduced to at least 11 dimensionality space, and 165 genes were reduced to at least 15 dimensionality space. Finally, PCA method, the classifier accuracy was 100% when the 48 genes were reduced to at least 18 dimensionality space, and 165 genes were reduced to at least 22 dimensionality space. The implementation (classifier accuracy was 100%) returned a ranked list in about 1766 sec for the lymphoma dataset, 2247 sec by LLE, and 3105 sec by PCA, much faster than 5343 sec by SVM without dimensionality reduction. Table 2 reports the comparison of the four methods on the lymphoma dataset.

From the results it is obviously seen that KLLE performs excellently on datasets dimensionality reduction. Two facts demonstrate this capability. One the one hand, in nonlinear structures dataset, the kernel based nonlinear dimensionality reduction KLLE preserves intrinsic properties more than the linear LLE and PCA. On the other hand, we also found that the time consumption of KLLE-SVM is smaller than those of the other methods, than the time consumption of SVM in computing lower dimensionality. The results also show that our proposed KLLE enhances SVM competence of classification in high-dimensionality.

6 Conclusion

The application of machine learning to data mining and analysis in the area of microarray analysis is rapidly gaining interest in the community. The large number of gene expressions coupled with analysis over a time course, provides an immense space of genomic dimensionality reduction and selection. In this paper, we presented an effective approach to reduce high-dimensionality and genes classifier in genomic microarray experiments. In our approach, kernel method is demonstrated to be able to extract the complicated nonlinear information embedded on the data sets by a nonlinear mapping. This paper proposed an improved kernel locally linear embedding algorithm for dimensionality reduction, based on the traditional LLE, kernel method and fuzzy KNN. The proposed algorithm compresses and denoises the redundant information in manifolds and preserves most intrinsic properties at the same time. It is conformed that our proposed KLLE has overcome the some primary shortcomings of the original algorithm, stimulating the applications of LLE.

The experimental results indicate that the proposed method performs well in dimensionality reduction and achieve high classification accuracies in SRBCT and lymphoma dataset. And the results also showed that this approach preserved the dataset's intrinsic nonlinear relationship and performed better than the current popular LLE and PCA approach. We conclude that the KLLE not only helps biological researchers classify differentiate cancers that are difficult to be classified for high-dimensionality, but also helps researchers focus on a small number of important genes to find the nonlinear relationship between those important genes.

Acknowledgements

This work is supported by Foundation of National Natural science No.10671030.

References

1. Shalon, D.; Smith, S.J.; Brown, P.O. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Research*. **1996**, 6(7), 639-45.
2. Tenenbaum, J.B.; Silva, V. de; Langford, J.C. A global geometric framework for. nonlinear dimensionality reduction. *Science*. **2000**, 260, 2319-2323.
3. Roweis, S.T.; Saul, L.K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*. *2000*, 290, 2323-2326.
4. Zhang, C.; Wang, J.; Zhao, N.; Zhang, D. Reconstruction and analysis of multi-pose face images based on nonlinear dimensionality reduction. *Pattern Recognition*. **2004**, 37(2), 325-336.
5. Elgammal, A.M.; Lee, C.S. Separating style and content on a nonlinear manifold. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. **2004**, 478-485.
6. Mekuz, N.; Bauckhage, C.; Tsotsos, J.K. Face recognition with weighted locally linear embedding. *The Second Canadian Conference on Computer and Robot Vision*. **2005**, 290-296.
7. Schölkopf, B.; Smola, A.; Müller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*. **1998**, 10(5), 1299-1319.
8. Shawe-Talyor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*. Cambridge, Cambridge Un-

inversity Press. **2004**.

9. Young, R.A. Biomedical discovery with DNA arrays. *Cell*. **2000**, 102, 9-15.
10. Brown, M.P.; Grundy, W.N.; Lin D.; Cristianini N.; Sugnet C.W.; Furey T.S.; Ares M.J.; Haussler D. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl Acad. Sci.* **2000**, 97, 262-267.
11. Lee, Y.; Lee, C.K. Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics*. **2003** 19(9), 1132-1139.
12. Wang, X.C; Paliwal, K.K. Feature extraction and dimensionality reduction algorithms and their applications in vowel recognition, Pergamon. *The Journal of the Pattern Recognition Society*. **2003**, 36, 2429-2439.
13. Haykin, S. *Neural networks: A Comprehensive Foundation*. New Jersey: Practice-Hall Press. **1999**, 330-332.
14. Marina, M.; Shi, J.b. *Learning segmentation by random walks*. Cambridge: Advances in NIPS 13. **2001**,873-879
15. Kouropteva, O.; Okun, O., Pietikainen, M. Selection of the optimal parameter value for the locally linear embedding algorithm. *Proc of the 1st International Conference on Fuzzy Systems and Knowledge Discovery, Singapore*. **2002**, 359-363.
16. Cover, T.M.; Hart, P.E. Nearest neighbour pattern classification. *IEEE Trans. Inform. Theory*. **1967**, IT-13, 21-27.
17. Keller, J.M.; Gray, M.R.; Givens, J.A. A fuzzy k-nearest neighbours algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*. **1985**, 15, 580-585.
18. Saul, L.; Roweis, S. Think globally, fit locally: unsupervised learning of nonlinear manifolds. *Technical Report MS CIS-02-18*, University of Pennsylvania. **2002,** 37, 134-135.
19. Vapnik, V.N. *Statistical Learning Theory*, John Wiley, New York. **1998**, 157-169.
20. Vapnik, V.N. *The nature of statistical learning theory.* NY: Springer-Verlag. **1995**.
21. Qian, Z.; Cai, Y.D.; Li, Y. A novel computational method to predict transcription factor DNA binding preference, *Biochem. Biophys. Res. Commun.* **2006**, 348, 1034-1037.
22. Vapnik, V.N. An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks*. **1999**, 10(5), 988-999.
23. Cristianini N.; Shawe-Talyor, J. *An introduction to support vector Machines*. Cambridge: Cambridge Uninversity Press. **2000**, 96-98.
24. Du, P.F.; He, T.; Li, Y.D. Prediction of C-to-U RNA editing sites in higher plant mitochondria using only nucleotide sequence features. *Biochemical and Biophysical Research Communications*. **2007**, 358, 336-341.
25. Ellis, M.; Davis, N.; Coop A.; Liu M.; Schumaker, L.; Lee, R.Y. et al. Development and validation of a method for using breast core needle biopsies for gene expression microarray analyses. *Clin. Cancer Res*. **2002**, 8 (5), 1155-1166.
26. Orr, M.S.; Scherf, U. Large-scale gene expression analysis in molecular target discovery. *Leukemia*. **2002**, 16 (4), 473-477.
27. Khan, J.; Wei, J.S.; Ringner, M.; Saal, L.H.; Ladanyi, M.; Westermann, F. et al. Classification and

diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat Med.* **2001**, 7(6), 673-9.

28. Nikhil, R.P.; Kripamoy, A.; Animesh, S.; Amari, S.I. Discovering biomarkers from gene expression data for predicting cancer subgroups using neural networks and relational fuzzy clustering. *BMC Bioinformatics*. **2007**, 8(5) 1-18.
29. Yeo, G.; Poggio, T. Multiclass classification of SRBCTs, *Technical Report AI Memo 2001-018 CBCL Memo 206, MIT.* **2001**.

Alizadeh et al. Alizadeh A.A.; Eisen M.B. et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, **2000**, 403, 503-511.

30. Troyanskaya, O.; Cantor, M. et al. Missing value estimation methods for DNA microarrays. *Bioinformatics*. **2001**,17,520-525.
31. Tusher, V.G.; Tibshirani R.; Chu, G. Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci. USA.* **2001**, 98, 5116-5121.
32. Tibshirani, R.; Hastie, T.; Narasimhan, B.; Chu, G. Class predicition by nearest shrunken centroids with applications to DNA microarrays. *Statistical Science*. **2003**, 18, 104-117.