Bioinformatics Advance Access published December 17, 2004

Multidimensional Support Vector Machines for Visualization of Gene Expression Data

D. Komura*
Research Center for Advanced Science and Technology,
The University of Tokyo, Tokyo 153-8904, Japan

H. Nakamura
Research Center for Advanced Science and Technology,
The University of Tokyo, Tokyo 153-8904, Japan

S. Tsutsumi
Research Center for Advanced Science and Technology,
The University of Tokyo, Tokyo 153-8904, Japan

H. Aburatani
Genome Science Div., Center for Collaborative Research,
The University of Tokyo, Tokyo 153-8904, Japan

S. Ihara
Research Center for Advanced Science and Technology,
The University of Tokyo, Tokyo 153-8904, Japan

Abstract

Motivation: Since DNA microarray experiments provide us with huge amount of gene expression data, they should be analyzed with statistical methods to extract the meanings of experimental results. Some dimensionality reduction methods such as Principal Component Analysis (PCA) are used to roughly visualize the distribution of high dimensional gene expression data. However, in the case of binary classification of gene expression data, PCA does not utilize class information when choosing axes. Thus clearly separable data in the original space may not be so in the reduced space used in PCA.

Results: For visualization and class prediction of gene expression data, we have developed a new SVM-based method called multidimensional SVMs, that generate multiple orthogonal axes. This method projects high dimensional data into lower dimensional space to exhibit properties of the data clearly and to visualize a distribution of the data roughly. Furthermore, the multiple axes can be used for class prediction. The basic properties of conventional SVMs are retained in our method: solutions of mathematical programming are sparse, and nonlinear classification is implemented implicitly through the use of kernel functions. The application of our method to

*To whom correspondence should be addressed.

Komura et al. (2004) Multidimensional Support Vector Machines for Visualization of Gene Expression Data. Symposium on Applied Computing, Proceedings of the 2004 ACM symposium on Applied computing, 175-179; http://doi.acm.org/10.1145/967900.967936

the experimentally obtained gene expression datasets for patients' samples indicates that our algorithm is efficient and useful for visualization and class prediction.

Contact: komura@hal.rcast.u-tokyo.ac.jp

Keyword: Multidimensional Support Vecotor Machines, visualization, gene expression data, binary classification

1 Introduction

DNA microarray has been the key technology in modern biology and helped us to decipher the biological system because of its ability to monitor the expression levels of thousands of genes simultaneously. Since DNA microarray experiments provide us with huge amount of gene expression data, they should be analyzed with statistical methods to extract the meanings of experimental results.

A great number of supervised learning algorithms have been proposed and applied to classification of gene expression data(Golub *et al.*, 1999)(Tibshirani *et al.*, 2002)(Khan *et al.*, 2001). Support Vector Machines (SVMs) have been paid attention in recent years because of their good performance in various fields, especially in the area of bioinformatics including classification of gene expression data(Furey *et al.*, 2000). However, SVMs predict a class of test samples by projecting the data into one-dimensional space based on a decision function. As a result, information loss of the original data is enormous.

Some methods are used for projecting high dimensional data into lower dimensional space to clearly exhibit the properties of the data and to roughly visualize the distribution of the data. Principal Component Analysis (PCA)(Fukunaga, 1990) and its derivatives, e.g. Nonlinear PCA(Diamantaras & Kung, 1996) and Kernel PCA(Schölkopf *et al.*, 1998), are most widely used for this purpose(Huang *et al.*, 2003). One drawback of PCA analysis is, however, that class information is not utilized for class prediction because PCA chooses axes based on the variance of overall data. Thus clearly separable data in the original space may not be so in the reduced space used in PCA. Another method for visualization and reducing dimension of data is discriminant analysis. It chooses axes based on class information in terms of within- and between-class variance. However, it is reported that SVMs often outperform discriminant analysis(Brown *et al.*, 2000).

The main purpose of this paper is to cover the shortcoming of SVMs by introducing multiple orthogonal axes for reducing dimensions and visualization of gene expression data. To this end, we have developed multidimensional SVMs (MD-SVMs), a new SVM-based method that generates multiple orthogonal axes based on margin between two classes to minimize generalization errors. The axes generated by this method reduce dimensions of original data to extract information useful in estimating the discriminability of two classes. This method fulfills the requirement of both visualization and class prediction. The basic properties of SVMs are retained in our method: solutions of mathematical programming are sparse, and nonlinear classification of data is implemented implicitly through the use of kernel functions.

This paper is organized as follows. In Section 2, we introduce the fundamental of SVMs. In Section 3, we describe the algorithm of MD-SVMs. In Section 4 and 5, we show numerical experiments on real gene expression datasets and reveal that our algorithm is effective for data visualization and class prediction.

1.1 Notation

$\mathbb{R}$ is defined as the set of real numbers. Each component of a vector $\boldsymbol{x} \in \mathbb{R}^n, i = 1, \cdots, m$ will be denoted by $x_j, j = 1, \cdots, n$. The inner product of two vectors $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^n$ will be denoted by $\boldsymbol{x} \cdot \boldsymbol{y}$. For a vector $\boldsymbol{x} \in \mathbb{R}^n$ and a scalar $a \in \mathbb{R}$, $a \leq \boldsymbol{x}$ is defined as $a \leq x_i$ for all $i = 1, \cdots, n$. For an arbitrary variable x, x^k is just a name of the variable with upper suffix , not defined as k-th power of x.

2 Support Vector Machines

Since details of SVMs are fully described in the articles(Vapnik, 1998)(Cristianini & Shawe-Taylor, 2000), we briefly introduce the fundamental principle of SVMs in this section. We consider a binary classification problem, where a linear decision function is employed to separate two classes of data based on m training samples $\boldsymbol{x}_i \in \mathbb{R}^n, i = 1, \cdots, m$ with corresponding class values $y_i \in \{\pm 1\}, i = 1, \cdots, m$. SVMs map a data $\boldsymbol{x} \in \mathbb{R}^n$ into a higher, probably infinite, dimensional space $\mathbb{R}^N$ than the original space with an appropriate nonlinear mapping $\phi : \mathbb{R}^n \to \mathbb{R}^N, n < N$. They generate the linear decision function of the form $f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w} \cdot \phi(\boldsymbol{x}) + b)$ in the high dimensional space, where $\boldsymbol{w} \in \mathbb{R}^N$ is a weight vector which defines a direction perpendicular to the hyperplane of the decision function, while $b \in \mathbb{R}$ is a bias which moves the hyperplane parallel to itself. The optimal decision function given by SVMs is a solution of an optimization problem

$$\begin{aligned} \min_{\boldsymbol{w},\xi} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi_i, \\ \text{s.t.} \quad & y_i(\boldsymbol{w} \cdot \phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i, i = 1, \cdots, m, \\ & \boldsymbol{\xi} \geq 0, \end{aligned} \tag{1}$$

with $C > 0$. Here, $\boldsymbol{\xi} \in \mathbb{R}^m$ is a vector whose elements are slack variables and $C \in \mathbb{R}$ is a regularization parameter for penalizing training errors. When $C \to \infty$, no training errors are allowed, and thus this is called hard margin classification. When $0 < C < \infty$, this is called soft margin classification because it allows some training errors. Note that a geometric margin γ between two classes is defined as $\frac{1}{\|\boldsymbol{w}\|^2}$. The optimization problem formalizes the tradeoff between maximizing margin and minimizing training errors. The problem is transformed into its corresponding dual problem by introducing lagrange multiplier $\boldsymbol{\alpha} \in \mathbb{R}^m$ and replacing $\phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)$ by kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)$ to be solved in an elegant way of dealing with a high dimensional vector space. The dual problem is

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i=1}^{m}\alpha_i, \\ \text{s.t.} \quad & 0 \leq \boldsymbol{\alpha} \leq C, \quad \sum_{i=1}^{m}\alpha_i y_i = 0. \end{aligned} \tag{2}$$

By virtue of the kernel function, the value of the inner product $\phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)$ can be obtained without explicit calculation of $\phi(\boldsymbol{x}_i)$ and $\phi(\boldsymbol{x}_j)$. Finally, the decision function becomes $f(\boldsymbol{x}) = \text{sign}\Big(\sum_{i=1}^{m}\alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\Big)$. by using kernel functions between training samples $\boldsymbol{x}_i, i = 1, \cdots, m$ and a test sample $\boldsymbol{x}$.

3 Multidimensional Support Vector Machines

In order to overcome the drawback that SVMs cannot generate more than one decision function, we propose a SVM-based method that can be used for both data visualization and class prediction in this section. We call this method multidimensional SVMs (MD-SVMs). We deal with the same problem as mentioned in Section 2. Conventional SVMs give an optimal solution set $(\boldsymbol{w}, b, \boldsymbol{\xi})$ which corresponds to a decision function, while our MD-SVMs give the multiple sets $(\boldsymbol{w}^k, b^k, \boldsymbol{\xi}^k)$,$k = 1, 2, \cdots, l$ with $l \leq n$, so that all the directions $\boldsymbol{w}_k$ are orthogonal to one another. The orthogonal axes can be used for reducing the dimension of original data and data visualization in three dimensional space by means of projection. Here the first set $(\boldsymbol{w}^1, b^1, \boldsymbol{\xi}^1)$ is equivalent to that obtained by conventional SVMs. Now we only refer to the steps of obtaining $(\boldsymbol{w}^k, b^k, \boldsymbol{\xi}^k)$, $k = 2, 3, \cdots, l$. In practice, the k-th set $(\boldsymbol{w}^k, b^k, \boldsymbol{\xi}^k)$ $k = 2, 3, \cdots, l$ are found with iterative computations of the optimization problem

$$\begin{aligned} \min_{\boldsymbol{w}^k,\boldsymbol{\xi}^k} \quad & \frac{1}{2}\|\boldsymbol{w}^k\|^2 + C\sum_{i=1}^{m}\xi_i^k, \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^k \cdot \phi(\boldsymbol{x}_i) + b^k) \geq 1 - \xi_i^k, \ i = 1, \cdots, m, \\ & \boldsymbol{\xi}^k \geq 0, \ \boldsymbol{w}^k \cdot \boldsymbol{w}^j = 0, \ j = 1, \cdots, k-1. \end{aligned} \tag{3}$$

This problem differs from that of conventional SVMs in the last constraint $\boldsymbol{w}^k \cdot \boldsymbol{w}^j = 0$. The weight vector $\boldsymbol{w}^j, j = 1, \cdots, k-1$ should be computed in advance by solving other optimization problems(3). The optimization problem is modified by introducing lagrange multipliers $\boldsymbol{\alpha}^k, \boldsymbol{\gamma}^k \in \mathbb{R}^m, \boldsymbol{\beta}^k \in \mathbb{R}^{k-1}$ and kernel functions. The primal Lagrangian is

$$\begin{aligned} L(\boldsymbol{w}^k, b^k, \boldsymbol{\xi}^k) &= \frac{1}{2}\|\boldsymbol{w}^k\|^2 + C\sum_{i=1}^{m} \xi_i^k \\ &+ \sum_{i=1}^{m} \alpha_i^k (1 - \xi_i^k - y_i(\boldsymbol{w}^k \cdot \phi(\boldsymbol{x}_i) + b^k)) \\ &+ \sum_{j=1}^{k-1} \beta_j^k (\boldsymbol{w}^k \cdot \boldsymbol{w}^j) - \sum_{i=1}^{m} \gamma_i^k \xi_i. \end{aligned} \tag{4}$$

Consequently, the optimization problem is

$$\begin{aligned} \max_{\boldsymbol{\alpha}^k, \boldsymbol{\beta}^k} \quad & -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i^k \alpha_j^k y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ & + \frac{1}{2}\sum_{i=1}^{k-1} \beta_i^k \beta_i^k (\boldsymbol{w}^i \cdot \boldsymbol{w}^i) + \sum_{i=1}^{m} \alpha_i^k, \\ \text{s.t.} \quad & 0 \leq \boldsymbol{\alpha}^k \leq C, \sum_{i=1}^{m} \alpha_i^k y_i = 0, \\ & \sum_{i=1}^{m} \alpha_i^k y_i \Big(\phi(\boldsymbol{x}_i) \cdot \boldsymbol{w}^j\Big) = 0, j = 1, \cdots, k-1 \end{aligned} \tag{5}$$

Here $\phi(\boldsymbol{x}_p) \cdot \boldsymbol{w}^q$ and $\boldsymbol{w}^p \cdot \boldsymbol{w}^p$ are calculated recursively as follows:

$$\begin{aligned} \phi(\boldsymbol{x}_p) \cdot \boldsymbol{w}^q &= \sum_{i=1}^{m} \alpha_i^q y_i K(\boldsymbol{x}_p, \boldsymbol{x}_i) \\ & \quad - \sum_{i=1}^{q-1} \beta_i^q \Big(\phi(\boldsymbol{x}_p) \cdot \boldsymbol{w}^i\Big), \end{aligned} \tag{6}$$

$$\begin{aligned} \boldsymbol{w}^p \cdot \boldsymbol{w}^p &= \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i^p \alpha_j^p y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ & - \sum_{i=1}^{m}\sum_{j=1}^{p-1} \alpha_i^p y_i \beta_j^p \Big(\phi(\boldsymbol{x}_i) \cdot \boldsymbol{w}^j\Big) + \sum_{i=1}^{p-1} \beta_i^p \beta_i^p (\boldsymbol{w}^i \cdot \boldsymbol{w}^i) \\ & - \sum_{i=1}^{m}\sum_{j=1}^{p-1} \alpha_i^p y_i \beta_j^p \Big(\phi(\boldsymbol{x}_i) \cdot \boldsymbol{w}^j\Big), \end{aligned} \tag{7}$$

(8)

where $\phi(\boldsymbol{x}_p) \cdot \boldsymbol{w}^1 = \sum_{i=1}^{m} \alpha_i^1 y_i K(\boldsymbol{x}_p, \boldsymbol{x}_i)$ and $\boldsymbol{w}^1 \cdot \boldsymbol{w}^1 = \sum_{i=1}^{m} \alpha_i^1 y_i \Big(\phi(\boldsymbol{x}_i), \boldsymbol{w}^1\Big)$. As can be seen, there is no need to calculate nonlinear map of data $\phi(\boldsymbol{x})$ in problem(5) because all nonlinear mappings can be replaced with kernel functions.

Note that this optimization problem is a nonconvex quadratic problem when k is more than 1. As a consequence, the optimal solutions are not easy to be obtained. In Section 4, we use local optimum for numerical experiments when k is 2 or 3. We note the experimental results are still encouraging.

The corresponding Karush-Kuhn-Tucker conditions are

$$\alpha_i^k\{1 - \xi_i^k - y_i(\boldsymbol{w}^k \cdot \phi(\boldsymbol{x}_i) + b^k)\} = 0, \tag{9}$$

$$\xi_i^k(\alpha_i^k - C) = 0, i = 1, \cdots, m. \tag{10}$$

These are exactly the same as conventional SVMs. We highlight the other properties conserved from conventional SVMs:

- Projecting data into high dimensional space is implicit, using kernel functions to replace inner products.
- The solutions $\boldsymbol{\alpha}^k$ of the optimization problem is sparse. Then the corresponding decision function depends only on few "Support Vectors".

Since each decision function is normalized independently to hold $\boldsymbol{w}^k \cdot \phi(\boldsymbol{x}_i) + b^k = y_i$ for $i = 1, \cdots, m$, data scales of the axes should be aligned with first axis (k=1) for visualization. The margin γ^k, the L2-distance between support vectors of each class of k-th axis, is

$$\left(\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i^k \alpha_j^k y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{k-1} \beta_i^k \beta_i^k (\boldsymbol{w}^i \cdot \boldsymbol{w}^i)\right)^{-\frac{1}{2}}. \tag{11}$$

So a scaling factor $s^k = \gamma^1/\gamma^k$ is

$$\sqrt{\frac{\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i^1 \alpha_j^1 y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i^k \alpha_j^k y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{k-1} \beta_i^k \beta_i^k (\boldsymbol{w}^i \cdot \boldsymbol{w}^i)}}. \tag{12}$$

The decision function of k-th step has the form $f^k(\boldsymbol{x}) = \text{sign}\Big(\sum_{i=1}^m \alpha_i^k y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b^k\Big)$. Since the right hand side of the equation has the function of projecting original data into one dimensional space, the data can be plot in up to three dimensional space for visualization . The coordinate of data $\boldsymbol{x} \in \mathbb{R}^m$ in three dimensional space is

$$(s^{k_1} g^{k_1}(\boldsymbol{x}), s^{k_2} g^{k_2}(\boldsymbol{x}), s^{k_3} g^{k_3}(\boldsymbol{x})), \tag{13}$$

where $g^k(\boldsymbol{x}) = \sum_{i=1}^m \alpha_i^k y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b^k$. The space represents a distribution of data clearly based on the margin between two classes.

4 Numerical Experiments

4.1 Method

In order to confirm the effectiveness of our algorithm, we have performed numerical experiments. MD-SVMs can generate multiple axes, up to the number of features. Here we choose three axes, $k = 1, 2, 3$, to simplify the experiments. When k is 2 or 3, we use local optimum in problem (5) since it is difficult to obtain the global solutions. In our experiments, we carry out hold-out validation because cross-validation changes decision functions every time the dataset is split. Then we compare the results obtained by MD-SVMs with those obtained by PCA.

In the experiments, the expression values for each of the genes are normalized such that the distribution over the samples has a zero mean and unit variance. Before normalization, we discard genes in the dataset with the overall average value less than 0.35. Then we calculate a score $F(x(j)) = \Big|(\mu^+(j) - \mu^-(j))/(\sigma^+(j) + \sigma^-(j))\Big|$, for the remaining genes. Here $\mu^+(j)(\mu^-(j))$ and $\sigma^+(j)(\sigma^-(j))$ denote the mean and standard deviation of the j-th gene of the samples labeled +1(-1), respectively. This score becomes the highest when the corresponding expression levels of the gene differ most in the two classes and have small deviations in each class. We select 100 genes with the highest scores and use them for hold-out validation. These procedures for gene selection are done only for training data for fair experiments.

The regularization parameter C in problem (5) is set to 1000. This value is rather large but finite because we would like to avoid ill-posed problems in a hard margin classification. We choose linear kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$ and RBF kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp -\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2$ with $\gamma = 0.001$ in the experiments of MD-SVMs.

4.2 Materials

Leukemia dataset(Golub *et al.*, 1999). This gene expression dataset consists of 72 leukemia samples, including 25 acute myeloid leukemia (AML) samples and 47 acute lymphoblastic leukemia (ALL) samples. They are obtained by hybridization on the Affymetrix GeneChip containing probe sets for 7070 genes. Training set contains 20 AML samples and 42 ALL samples. Test set contains 5 AML samples and 5 ALL samples. AML samples are labeled +1 and ALL samples are labeled -1.

Lung tissue dataset(Bhattacharjee *et al.*, 2001). This dataset consists of 203 samples from lung tissue, including 16 samples from normal tissue and 187 samples from cancerous tissue, and is obtained by hybridization on the Affymetrix U95A Genechip containing probe sets for 12558 genes. Training set includes 13 samples from normal tissue and 157 samples from cancerous tissue. Test set includes 3 samples from normal tissue and 30 samples from cancerous tissue. Samples from normal tissue are labeled +1 and samples from cancerous tissue are labeled -1.

5 Results and Discussion

The results of numerical experiments are shown in Figure 1, and Table 1 and 2. The distributions obtained by MD-SVMs on the leukemia dataset and the lung tissues dataset are given in Figure 1-(1) and 1-(3), respectively. Those obtained by PCA are given in Figure 1-(2) and 1-(4), respectively. The number of misclassified samples by MD-SVMs are summarized in Table 1 and 2. In these tables, the class of the samples is predicted based on decision functions $f^k(\boldsymbol{x}), k = 1, 2, 3$, corresponding to each of the three axes.

Figure 1-(1) and 1-(3) illustrate that MD-SVMs are likely to separate the samples of each class in all the three directions. However, as shown in Figure 1-(2) and 1-(4), PCA does not separate the samples in the directions of the 2nd or the 3rd axis. These axes by PCA are dispensable with the objective of visualization for class prediction. In other words, MD-SVMs gather the plots of the samples into the appropriate clusters of each class, while PCA rather scatters them. Furthermore, in the distribution by MD-SVMs for the lung tissues dataset, one sample outlies from correct clusters (indicated by arrows in Figure 1-(3)). Though this sample also seems to be an outlier in the distribution by PCA (also indicated in Figure 1-(4)), the outlier significantly deviates in MD-SVMs. This may arise from the fact that MD-SVMs can separate the samples in all the directions. These observations indicate that MD-SVMs are well suited for visualizing in binary classification problems.

The significant advantage of MD-SVMs over PCA is the ability to predict the classes. MD-SVMs can predict the classes of samples based on the decision functions $f^k(\boldsymbol{x})$ without extra computation, while PCA cannot. The predicted class of a sample should be matched by the all the decision functions in an ideal case. However that does not always occur as seen in Table 1 and 2. In such cases, the simplest method for prediction is to use only the 1st axis, which corresponds to the decision function generated by conventional SVMs. The idea is supported by the fact that the 1st decision function classifies the samples most correctly in almost all cases in Table 1 and 2. The more advanced method is weighted voting. Scaling factor or normalized objective values in problem (5) are the candidate of the weight.

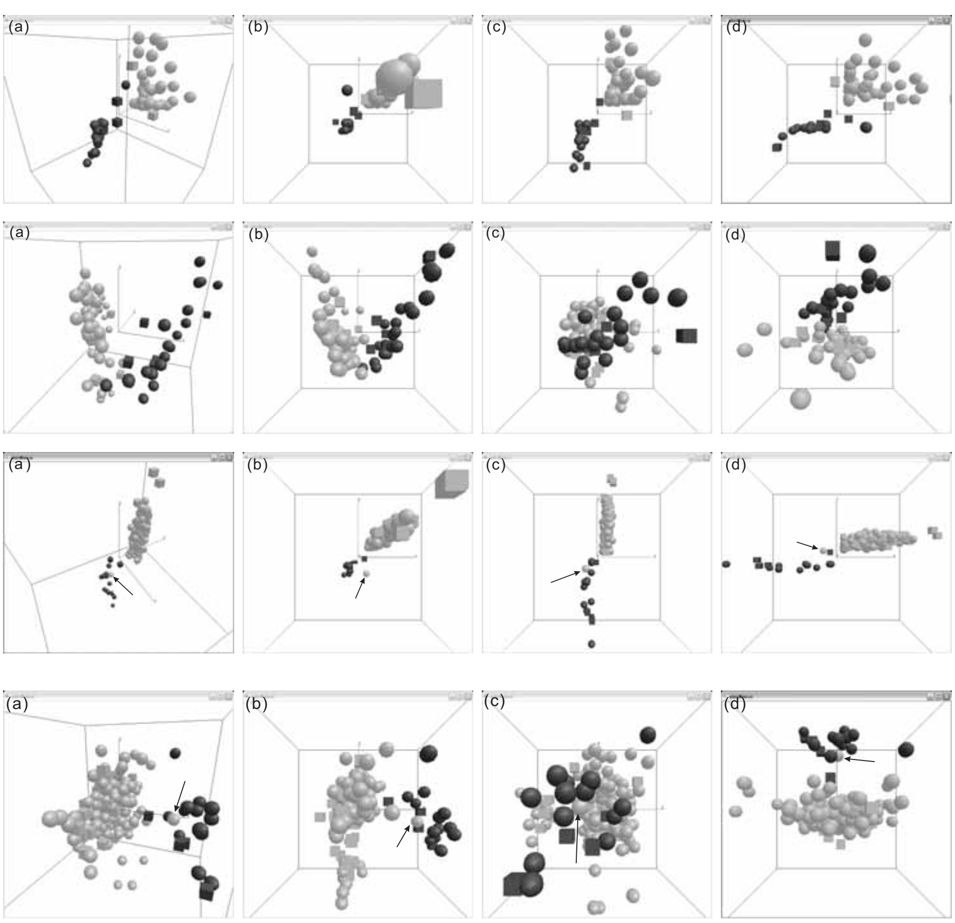
Multiple decision functions generated by MD-SVMs are useful for outlier detection. Samples misclassified by multiple decision functions may be mislabeled or categorized into unknown classes. For example, see the column "3 axes" of test sample of the lung tissues dataset with RBF kernel in Table 2. This sample is misclassified by all decision functions, so we can say that this data contains some experimental error. The hierarchical clustering method also supports our result. These results indicate that MD-SVMs can be used for finding candidates of outliers.

6 Conclusion

For both visualization and class prediction of gene expression data, we propose a new method called Multidimensional Support Vector Machines. We formulate the method as a quadratic program and implement the algorithm. This is motivated by the following facts: 1)SVMs perform better than the other classification algorithms, but they generate only one axis for class prediction. 2)PCA chooses multiple orthogonal axes, but it cannot predict classes of samples without other classification algorithms. We have tried to cover the shortcomings of both methods. MD-SVMs choose multiple orthogonal axes, which correspond to decision functions, from high dimensional space based on a margin between two classes. These multiple axes can be used for both visualization and class prediction.

Numerical experiments on real gene expression data indicate the effectiveness of MD-SVMs. All axes generated by MD-SVMs are taken into account for separating class of samples, while the 2nd and the 3rd axes by PCA are not. The samples in the distributions by MD-SVMs gather into appropriate clusters more vividly than those by PCA. MD-SVMs can predict the classes of the samples with multiple decision functions. We also indicate that MD-SVMs are useful for outlier detection with multiple decision functions.

There are several future works to be done on MD-SVMs: (1) application of our method to wider variety of gene expression datasets, (2) investigation of gene selection for preprocess of analysis, and (3) investigation on class prediction method with multiple decision functions. Firstly, the use of more suitable samples may show that the axes chosen by MD-SVMs separate samples more clearly than those by PCA. Secondly, since the conventional SVMs show good generalization performance especially with large number of features, it is expected that MD-SVMs show much better performance than PCA with increasing the number of genes used in the numerical experiments. Since the element of weight vector generated by SVMs is one of the measures of discrimination power of the corresponding genes(Guyon *et al.*, 2002), that generated by MD-SVMs can be used for gene selection. Thirdly, the classification with prob-



(1)Distribution obtained by MD-SVMs for the leukemia dataset with linear kernel.

(2)Distribution obtained by PCA on the leukemia dataset.

(3) Distribution obtained by MD-SVMs for the lung tissues dataset with linear kernel. The sample indicated by arrows appears to be an outlier.

(4) Distribution obtained by PCA for the lung tissues dataset. The sample indicated by arrows is the same on in (3) but less deviates.

Figure 1: (a)Cross shot, (b)1st axis (x axis) and 2nd axis (y axis), (c)2nd axis (x axis) and 3rd axis (y axis), (d)3rd axis (x axis) and 1st axis (y axis). Black objects and white objects indicate AML samples (or normal tissues) ALL samples (or cancerous tissues), respectively. Training data and test data are expressed as a sphere and a cube, respectively.

ability as well as the weighted voting mentioned in Section 4 may be achieved in our scheme since the conventional SVMs have been already expanded for the purpose with sigmoid functions(Platt, 1999). We hope that our method sheds some lights on the future study of gene expression experiments.

References

Bhattacharjee, A., Richards, W., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E., Lander, E., Wong, W., Johnson, B., Golub, T., Sugarbaker, D. & Meyerson, M. (2001) Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma sub-classes. *Proceedings of the National Academy of Sciences of the United States of America,* **98**, 13790–13795.

Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., Ares, M. & Haussler, D.

Table 1: Number of classification errors in the MD-SVMs for the leukemia dataset. The columns "n-th axis", $n = 1, 2, 3$, indicates the number of samples misclassified by n-th decision function. The columns "n axes", $n = 1, 2, 3$, indicates the number of samples misclassified by n decision functions.

Kernel	Sample	# of samples	1st axis	2nd axis	3rd axis	1 axis	2 axes	3 axes
linear	Training	62	0	1	2	1	1	0
RBF	Training	62	0	2	7	5	2	0
linear	Test	10	1	1	2	2	1	0
RBF	Test	10	0	2	0	2	0	0

(2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America,* **97**, 262–267.

Cristianini, N. & Shawe-Taylor, J. (2000) *An introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, New York.

Diamantaras, K. & Kung, S. (1996) *Principal Component Neural Networks Theory and Applications.* John Wiley & Sons, New York.

Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition.* Academic Press, New York.

Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M. & Haussler, D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics,* **16**, 906–914.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C. & Lander, E. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science,* **286**, 531–537.

Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. (2002) Gene selection for cancer classification using support vector machines. *Journal of Machine Learning,* **46**, 389–422.

Huang, E., Ishida, S., Pittman, J., Dressman, H., Bild, A., Kloos, M., D'Amico, M., Pestell, R., West, M. & Nevins, J. (2003) Gene expression phenotypic models that predict the activity of oncogenic pathways. *Nature Genetics,* **34**, 226–230.

Khan, J., Wei, J., Ringnér, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C., Peterson, C. & Meltzer, P. (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine,* **7**, 673–679.

Platt, J. (1999) *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.* MIT Press, Cambridge, MA.

Schölkopf, B., Smola, A. & Müller, K. (1998) Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation,* **10**, 1299–1319.

Tibshirani, R., Hastie, T., Narasimhan, B. & Chu, G. (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences of the United States of America,* **99**, 6567–6572.

Vapnik, V. (1998) *Statistical Learning Theory.* John Wiley & Sons, New York.

Table 2: Number of classification errors in the MD-SVMs on the lung dataset. See the caption of Table 1 for other explanation.

Kernel	Sample	# of samples	1st axis	2nd axis	3rd axis	1 axis	2 axes	3 axes
Linear	Training	170	0	1	1	0	1	0
RBF	Training	170	0	3	5	2	3	0
Linear	Test	33	1	0	0	1	0	0
RBF	Test	33	1	1	1	0	0	1