BIOINFORMATICS

Vol. 19 no. 9 2003, pages 1132–1139
DOI: 10.1093/bioinformatics/btg102

Classification of multiple cancer types by multicategory support vector machines using gene expression data

Yoonkyung Lee[1,*] and Cheol-Koo Lee[2]

[1]Department of Statistics, The Ohio State University, Columbus, OH 43210 and [2]Molecular and Environmental Toxicology Center, University of Wisconsin, Madison, WI 53706, USA

Received on April 30, 2002; revised on July 31, 2002; December 10, 2002; accepted on December 10, 2002

ABSTRACT

Motivation: High-density DNA microarray measures the activities of several thousand genes simultaneously and the gene expression profiles have been used for the cancer classification recently. This new approach promises to give better therapeutic measurements to cancer patients by diagnosing cancer types with improved accuracy. The Support Vector Machine (SVM) is one of the classification methods successfully applied to the cancer diagnosis problems. However, its optimal extension to more than two classes was not obvious, which might impose limitations in its application to multiple tumor types. We briefly introduce the Multicategory SVM, which is a recently proposed extension of the binary SVM, and apply it to multiclass cancer diagnosis problems.

Results: Its applicability is demonstrated on the leukemia data (Golub *et al.*, 1999) and the small round blue cell tumors of childhood data (Khan *et al.*, 2001). Comparable classification accuracy shown in the applications and its flexibility render the MSVM a viable alternative to other classification methods.

Supplementary Information: http://www.stat.ohio-state.edu/∼yklee/msvm.html

Contact: yklee@stat.ohio-state.edu

INTRODUCTION

The advent of DNA microarray technology shifted the scale of genomics research and several thousand genes can be studied in a single experiment, nowadays. DNA microarray measures the relative amount of mRNA in isolated cells or biopsied tissues from patients. Since transcriptional changes accurately reflect the status of disease including cancers (DeRisi *et al.*, 1996; Zhang *et al.*, 1997; Perou *et al.*, 1999) gene expression profiles can be used to classify different types of cancers. Currently, cancer diagnosis highly depends on a variety of histological observations, including immunohistochemical assays, which detect cancer biomarker molecules. However, these assays have limitations due to morphological similarity and lack of available biomarkers of cancers. Accurate diagnosis promotes the efficacy of a proper treatment of cancers. Under the premise of gene expression patterns as fingerprints at the molecular level, systematic methods to classify tumor types using gene expression data have been studied (Golub *et al.*, 1999; Khan *et al.*, 2001).

Most training data sets (a set of pairs of a gene expression profile and the tumor type that it falls into) have a fairly small sample size compared to the number of genes investigated. This data structure creates an unprecedented challenge to some classification methodologies. The Support Vector Machine (SVM) was one of the methods successfully applied to the cancer diagnosis problem in the previous studies (Mukherjee *et al.*, 1999; Furey *et al.*, 2000). In principle, the SVM can be applied to very high-dimensional data without altering its formulation. Such capacity is well suited to the microarray data structure. Since the SVM was mainly developed for two-class problems, multiclass problems have been tackled indirectly by solving a series of binary problems. Using the SVMs in the one-versus-rest fashion is very common, but it has potential drawbacks when classes overlap considerably. Combining the binary SVMs for all pairs of classes is another popular approach. The DAG SVM algorithm is one of this kind with fast testing time (Platt *et al.*, 2000). The pairwise approach often exhibits large variability since each binary classifier is estimated from a small subset of the training data and it allows only a simple cost structure when different misclassification costs are concerned. As a generic approach to multiclass problems, we consider treating all the classes simultaneously. Although several extensions to the multiclass case have been proposed (Vapnik, 1998; Bredensteiner and Bennett,

*To whom correspondence should be addressed.

1999), its optimal extension was not obvious in relation to the theoretically best classification rule. In order to overcome possible limitations, the MSVM, an optimal extension of the binary SVM, was proposed recently (Lee *et al.*, 2001). We apply the MSVM to two gene expression data sets to demonstrate its effectiveness for the diagnosis of multiple cancer types. Also, we discuss how to assess the prediction strength of the MSVM, and mention other issues often arising in microarray data analysis; the effect of data preprocessing, gene selection, and dimension reduction.

METHODS

Binary SVM

The binary SVM paradigm has a nice geometrical interpretation of discriminating one class from the other by a separating hyperplane with maximum margin (Vapnik, 1998). Now, it is commonly known that the SVM can be cast as a regularization problem (Wahba, 1998). In classification problems, we are given a training data set that consists of n samples, $(\boldsymbol{x}_i, y_i)$ for $i = 1, \ldots, n$. $\boldsymbol{x}_i \in R^d$ represents the input vector and y_i denotes the class label. In the binary SVM setting, y_i is either 1 or -1, and the methodology seeks a function $f(\boldsymbol{x}) = h(\boldsymbol{x}) + b$ with $h \in H_K$, a reproducing kernel Hilbert space (RKHS) and b, a constant minimizing

$$\frac{1}{n}\sum_{i=1}^{n}(1 - y_i f(\boldsymbol{x}_i))_+ + \lambda\|h\|^2_{H_K} \qquad (1)$$

where $(x)_+ = \max(x, 0)$. $\|h\|^2_{H_K}$ denotes the square norm of the function h defined in the RKHS with the reproducing kernel function $K(\cdot,\cdot)$, measuring the complexity or smoothness of h (Wahba, 1990). λ is a tuning parameter which balances the data fit and the complexity of $f(\boldsymbol{x})$. The classification rule induced by $f(\boldsymbol{x})$ is $\phi(\boldsymbol{x}) = \text{sign}[f(\boldsymbol{x})]$. The zero level curve of $f(\boldsymbol{x})$ yields the classification boundary of the rule $\phi(\boldsymbol{x})$. Lin (2002) showed that the solution $f(\boldsymbol{x})$ approximates directly the majority class label $sign(p_1(\boldsymbol{x}) - 1/2)$, when flexible kernel functions are used. Here $p_1(\boldsymbol{x}) = P(Y = 1|X = \boldsymbol{x})$ with (X, Y) denoting a random sample from the underlying distribution $P(\boldsymbol{x}, y)$. Thereby, the SVM efficiently implements the Bayes rule that predicts the most likely class at $\boldsymbol{x}$ if $p_1(\boldsymbol{x})$ is known. However, because of this particular mechanism, using SVMs to solve multiclass problems in the one-vs-rest fashion may fail under some circumstances.

Multicategory SVM

For the multiclass problem, assume the class label $y_i \in \{1, \ldots, k\}$ without loss of generality. k is the number of classes. We briefly review the MSVM (Lee *et al.*, 2001, 2002). Each class label is coded as a k-dimensional vector with 1 in the jth coordinate and $-\frac{1}{k-1}$ elsewhere if it falls into class j. We define a k-tuple of separating functions $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$ with the sum-to-zero constraint, $\sum_{j=1}^{k} f_j(\boldsymbol{x}) = 0$ for any $\boldsymbol{x} \in R^d$. Note that the constraint holds implicitly for coded class labels $\boldsymbol{y}_i$. Analogous to the binary case, we consider $\boldsymbol{f}(\boldsymbol{x}) \in \prod_{j=1}^{k}(\{1\} + H_K)$, the product space of k RKHS's. Thus, each component $f_j(\boldsymbol{x})$ can be expressed as $h_j(\boldsymbol{x}) + b_j$ with $h_j \in H_K$. Define Q as the k by k matrix with 0 on the diagonal, and 1 elsewhere. It represents the cost matrix when all the misclassification costs are equal. Let L be a function which maps a class label $\boldsymbol{y}_i$ to the jth row of the matrix Q if $\boldsymbol{y}_i$ indicates class j. The MSVM finds $\boldsymbol{f}(\boldsymbol{x}) \in \prod_{1}^{k}(\{1\} + H_K)$, with the sum-to-zero constraint, minimizing

$$\frac{1}{n}\sum_{i=1}^{n} L(\boldsymbol{y}_i)\cdot(\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{y}_i)_+ + \frac{1}{2}\lambda\sum_{j=1}^{k}\|h_j\|^2_{H_K} \qquad (2)$$

where $(\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{y}_i)_+ \equiv [(f_1(\boldsymbol{x}_i) - y_{i1})_+, \ldots, (f_k(\boldsymbol{x}_i) - y_{ik})_+]$ and the $\cdot$ operation indicates the Euclidean inner product. The classification rule induced by $(f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$ is $\phi(\boldsymbol{x}) = \arg\max_j f_j(\boldsymbol{x})$. We can verify that the binary SVM formulation (1) is a special case of (2) when $k = 2$. This formulation generalizing the binary SVM paradigm carries over the efficiency of implementing the Bayes rule in the same fashion as the binary case. To establish this, we identify the asymptotic target function of (2), which is the minimizer of its limit data fit functional, $E[L(Y)\cdot(\boldsymbol{f}(X) - Y)_+]$.

LEMMA 1. *(Lee* et al., *2001) The minimizer of* $E[L(Y)\cdot(\boldsymbol{f}(X) - Y)_+]$ *under the sum-to-zero constraint is* $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$ *with*

$$f_j(\boldsymbol{x}) = \begin{cases} 1 & \text{if } j = \arg\max_{\ell=1,\ldots,k} p_\ell(\boldsymbol{x}) \\ -\frac{1}{k-1} & \text{otherwise} \end{cases} \qquad (3)$$

where $p_\ell(\boldsymbol{x}) = P(Y = \ell|X = \boldsymbol{x})$ *for* $\ell = 1, \ldots, k$.

So, for flexible RKHS and appropriately chosen λ, the solution $\boldsymbol{f}(\boldsymbol{x})$ to (2) is expected to be close to the most probable class code, bypassing the estimation of probabilities $p_j(\boldsymbol{x})$. In addition, its extension to accommodate unequal misclassification costs is straightforward by replacing the cost matrix Q by a general cost matrix and redefining $L(\cdot)$ accordingly.

The problem of finding constrained functions $(f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$ minimizing (2) is then transformed into that of finding finite-dimensional coefficients instead, with the aid of a variant of the representer theorem. It was shown that to find $(f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$ with the

sum-to-zero constraint, minimizing (2) is equivalent to find $(f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$ of the form

$$f_j(\boldsymbol{x}) = b_j + \sum_{i=1}^{n} c_{ij} K(\boldsymbol{x}_i, \boldsymbol{x}) \text{ for } j = 1, \ldots, k \quad (4)$$

with the sum-to-zero constraint only at $\boldsymbol{x}_i$ for $i = 1, \ldots, n$, minimizing (2). Omitting intermediate steps and introducing nonnegative Lagrange multipliers $\boldsymbol{\alpha}_j \in R^n$, we get the following dual problem:

$$\min_{\boldsymbol{\alpha}_j} \frac{1}{2} \sum_{j=1}^{k} (\boldsymbol{\alpha}_j - \bar{\boldsymbol{\alpha}})^t K (\boldsymbol{\alpha}_j - \bar{\boldsymbol{\alpha}}) + n\lambda \sum_{j=1}^{k} \boldsymbol{\alpha}_j^t \boldsymbol{y}_{\cdot j} \quad (5)$$

$$\text{subject to } 0 \le \boldsymbol{\alpha}_j \le L_j \text{ for } j = 1, \ldots, k \quad (6)$$

$$(\boldsymbol{\alpha}_j - \bar{\boldsymbol{\alpha}})^t \boldsymbol{e} = 0 \text{ for } j = 1, \ldots, k \quad (7)$$

where $L_j \in R^n$ is the jth column of the n by k matrix with the ith row $L(\boldsymbol{y}_i)$, and similarly $\boldsymbol{y}_{\cdot j}$ denotes the jth column of the n by k matrix with the ith row $\boldsymbol{y}_i$. $\bar{\boldsymbol{\alpha}}$ is the average of $\boldsymbol{\alpha}_j$'s, and $\boldsymbol{e}$ denotes the vector of ones of length n. With some abuse of notation, the n by n matrix $K \equiv (K(\boldsymbol{x}_i, \boldsymbol{x}_\ell))$. Once we solve the quadratic programming (QP) problem, the coefficients $\boldsymbol{c}_{\cdot j} = (c_{1j}, \ldots, c_{nj})^t = -\frac{1}{n\lambda}(\boldsymbol{\alpha}_j - \bar{\boldsymbol{\alpha}})$. b_j can be found from any of the examples with unbounded $\boldsymbol{\alpha}_{ij}$ satisfying (6) strictly by the Karush–Kuhn–Tucker complementarity conditions. $(\boldsymbol{\alpha}_{i1}, \ldots, \boldsymbol{\alpha}_{ik}) = 0$ implies $(c_{i1}, \ldots, c_{ik}) = 0$, so removing such example $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ would not affect the solution at all. Like the binary SVM, we call examples with $\boldsymbol{c}_i = (c_{i1}, \ldots, c_{ik}) \neq 0$, support vectors in the multicategory case. The MSVM retains the sparsity of the solution in the same way as the binary SVM.

Comparison and tuning

MSVMs would be more effective than the common one-vs-rest SVMs for overlapping classes (Lee *et al.*, 2002). However, there could be a tradeoff between accuracy and speed. The MSVM needs solving a QP problem with $(k-1)n$ variables once, while the one-vs-rest approach amounts to solving k QP problems with n variables and the pairwise approach leads to $k(k-1)/2$ QP problems of size less than n. A QP algorithm typically requires computing time at least in some polynomial order of its problem size. So, solving smaller binary problems several times may be computationally cheaper than solving a big multiclass problem once. As an empirical study, Hsu and Lin (2002) compared several methods to solve multiclass problems using SVMs in terms of their performance and computing time. Although our method was not included in the study, the comparisons are still relevant because some multiclass extensions in the study share the same computational complexity. It was reported that considering all the classes at once tends to be slower than solving a series of binary problems, however, the former needed fewer support vectors. The two different approaches showed pretty comparable accuracy with nonlinear kernels, while using the linear kernel resulted in the worst accuracy to the one-versus-rest approach. In real timescale, the computing time differences are in the order of some seconds for small problems like the applications in consideration, thus practically negligible. All the computations in this paper were done via MATLAB 6.1 with an interface to PATH 3.0 (Ferris and Munson, 1999).

As with other regularization methods, the efficiency of our method depends on the tuning parameter(s), λ (and other parameter in the kernel function). 5-fold or 10-fold cross validation based on misclassification counts is often used. Alternatively, an approximate leaving-out-one cross validation (LOOCV) function, called generalized approximate cross validation (GACV) has been derived for the MSVM (Lee *et al.*, 2002). In practice, one can choose the minimizer of GACV as appropriate tuning parameters without really doing LOOCV, which might be computationally prohibitive for large samples. For cancer diagnosis problems using gene expression patterns, LOOCV is still feasible since most of available data sets so far, are of small sample size.

Assessment of prediction strength

This section concerns how to measure strength or confidence of a class prediction made by SVMs. In many applications such as medical diagnosis, making a wrong prediction could be more serious than reserving a call. A weakly diagnosed example would require further specialized investigation for a more informative call. So, we wish to reject weak predictions with a reasonable strength measure. For classification methods that provide an estimate of the conditional probability of each class at $\boldsymbol{x}$, the probability estimate itself can serve as a strength measure. The mechanism of the SVM to extract the necessary information for the minimum error rate is very simple and efficient, however inevitably limited in restoring the probability from the estimated class code. Yet, there have been a couple of approaches to address this issue for SVMs in the context of microarray applications (Mukherjee *et al.*, 1999; Yeo and Poggio, 2001). Although other elaborate methods to map SVM outputs to probabilities have been proposed by Vapnik (1998) and Platt (1999) in general settings, it would be still difficult to restore accurate probability estimates from SVM outputs without heavily relying on a prior assumption, if data are quite separable or flexible kernel functions are used. Our motivation is to attach a confidence statement to each prediction, which may not be a precise probability estimate but reflects relative accuracy of the prediction, so that it can be useful in detecting borderline cases.

Multicategory case. Mukherjee *et al.* (1999) suggested a confidence measure for an SVM class prediction in the binary case, based on the idea that the bigger the margin $|f|$, the stronger the prediction. A simple variant of this treatment is devised for the multiclass case. An MSVM output $(f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$ close to a class code indicates a strong prediction away from the classification boundary. The multiclass loss function in (2), $g(\boldsymbol{y}, \boldsymbol{f}) \equiv L(\boldsymbol{y}) \cdot (\boldsymbol{f} - \boldsymbol{y})_+$ sensibly measures the proximity between an MSVM decision vector and a coded class, reflecting how strong their association is in the classification context. For the time being, a class label and its vector valued class code will be used interchangeably as an input argument of the loss g and other occasions without causing much confusion. Suppose that the probability of a correct prediction given $\boldsymbol{f}(\boldsymbol{x}) = (f_1, \ldots, f_k)$ at $\boldsymbol{x}$, $P(Y = \arg\max_j f_j | \boldsymbol{f})$ depends on $\boldsymbol{f}$ only through the loss, $g(\arg\max_j f_j, \boldsymbol{f})$ for the predicted class. The smaller the loss, the stronger the prediction. Then the strength of the MSVM prediction, $P(Y = \arg\max_j f_j | \boldsymbol{f})$ can be inferred from the training data by cross validation. For example, leaving out the ith example $(\boldsymbol{x}_i, y_i)$, we get the MSVM decision vector $\boldsymbol{f}(\boldsymbol{x}_i)$ based on the remaining samples, and the corresponding pair of the loss, $g(\arg\max_j f_j(\boldsymbol{x}_i), \boldsymbol{f}(\boldsymbol{x}_i))$ and the indicator of a correct decision $I(y_i = \arg\max_j f_j(\boldsymbol{x}_i))$. Then, $P(Y = \arg\max_j f_j | \boldsymbol{f})$, as a function of $g(\arg\max_j f_j, \boldsymbol{f})$ can be estimated using the pairs of the loss and the indicator from the training data. If we further assume the symmetry of k classes, that is, $P(Y = 1) = \cdots = P(Y = k)$ and $P(\boldsymbol{f} | Y = y) = P(\pi(\boldsymbol{f}) | Y = \pi(y))$ for any permutation operator π of $\{1, \ldots, k\}$, it follows that $P(Y = \arg\max_j f_j | \boldsymbol{f}) = P(Y = \pi(\arg\max_j f_j) | \pi(\boldsymbol{f}))$. Consequently, under these symmetry and invariance assumption with respect to k classes, we can pool the pairs of the loss and the indicator for all the classes, and estimate the invariant prediction strength function in terms of the loss, regardless of the predicted class. In almost separable classification problems, we might see the loss values for correct classifications only, impeding the estimation of the prediction strength. One may use heuristics of predicting a class only when its projected loss is less than, say, the 95th percentile of the empirical loss distribution. This cautious measure will be exercised in the application following this section.

RESULTS AND DISCUSSION

Leukemia data

We revisited the leukemia data set as a three-class problem. Golub *et al.* (1999) suggested gene expression monitoring for the classification of two leukemias, ALL (acute lymphoblastic leukemia) and AML (acute myeloid leukemia). These two cancer types were identified based on their origins, lymphoid (lymph or lymphatic tissue related) and myeloid (bone marrow related), respectively. ALL could be further divided into B-cell and T-cell ALLs. The 'weighted voting scheme', a variant of quadratic discriminant analysis was applied to the data set as a two-class (ALL/AML) problem in the original paper. The number of genes in the study is 7129. The data set consists of 38 examples in the training set and 34 examples in the test set. For the parsimony and the accuracy of prediction, we considered selecting relevant genes (variables) first. Typically, standardization of the variables precedes variable selection. Although standardization of each variable across samples is common practice, standardization of each sample (array) across genes is often adopted in gene expression analysis. Additional preprocessing steps were taken in Dudoit *et al.* (2002) before standardization: (i) thresholding (floor of 100 and ceiling of 16000), (ii) filtering (exclusion of genes with $\max/\min \leq 5$ and $\max - \min \leq 500$ across the samples), (iii) base 10 logarithmic transformation. This filtering resulted in 3571 genes. To see the effect of preprocessing and standardization, we tried either (A) standardizing each gene, or (B) preprocessing the data first as above, and standardizing each array. Selecting important variables out of 7129 would be a formidable task if we require learning classifiers with all the possible subsets of the variables. To circumvent the difficulty, simple prescreening measures were used to pick out relevant variables in the previous applications. We used the ratio of between classes sum of squares to within class sum of squares for each gene, and picked genes with the largest ratios (Dudoit *et al.*, 2002). For gene ℓ, $x_{i\ell}$ denotes the expression level from patient i, and the ratio is defined as

$$\frac{BSS(\ell)}{WSS(\ell)} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{k} I(y_i = j)(\bar{x}_{\cdot\ell}^{(j)} - \bar{x}_{\cdot\ell})^2}{\sum_{i=1}^{n}\sum_{j=1}^{k} I(y_i = j)(x_{i\ell} - \bar{x}_{\cdot\ell}^{(j)})^2} \quad (8)$$

where n is the training sample size, $I(\cdot)$ is the indicator function, $\bar{x}_{\cdot\ell}^{(j)}$ indicates the average expression level of gene ℓ for class j, and $\bar{x}_{\cdot\ell}$ is the overall mean expression levels of gene ℓ in the training set. Figure 1 depicts the expression levels of 40 most important genes in the training set. The heat map illustrates that the selected 40 genes are very informative in discriminating the three classes. The supplementary information website contains the list of top 20 genes. Those genes encode functional proteins responsible for transcription factor, development, metabolism and structure. Since B-cell and T-cell ALL (ALLB/ALLT) arise from the same origin, we expected that these classes show similar trend in gene expression. However, surprisingly, the inspection of 20 top ranked genes revealed that gene expression patterns in ALLB are much closer to those in AML than ALLT. Figure 2 illustrates four different patterns of those genes. More than

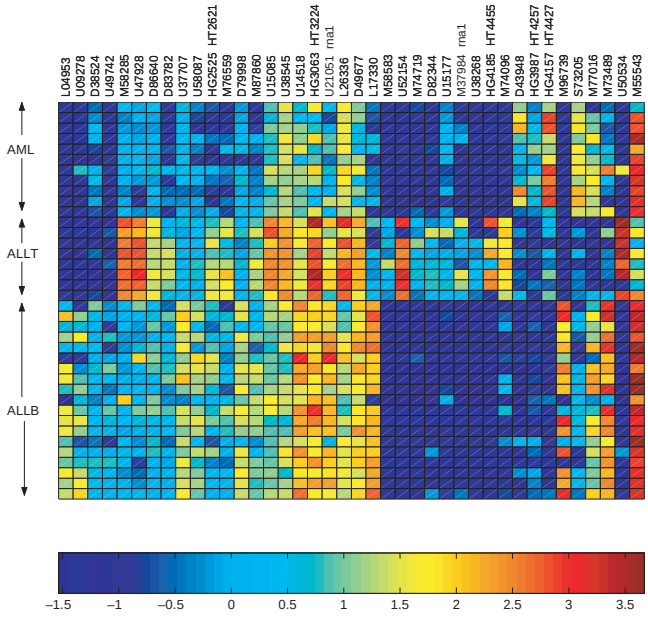

Fig. 1. The heat map shows the expression levels of 40 most important genes for the training samples standardized according to (B). Each row corresponds to a sample, which is grouped into the three classes, and the columns represent genes. The 40 genes are clustered and rearranged in a way the similarity within each class and the dissimilarity between classes are easily recognized.

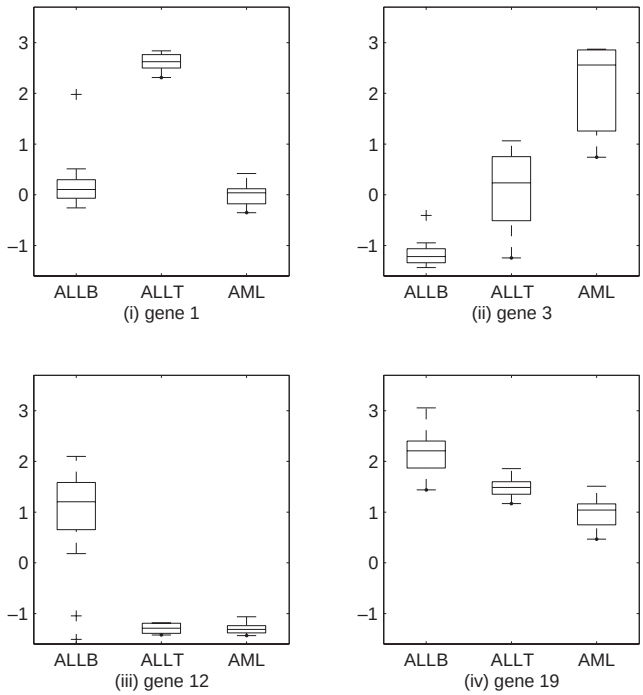

Fig. 2. The box plots show four different gene expression patterns from the top 20 genes, each numbered as its rank. A possible grouping of the genes depending on the patterns is (i) genes 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 14, 16 and 17, (ii) genes 3 and 18, (iii) genes 12, 13 and 15, and (iv) genes 19. Genes 20, not included in the grouping, showed a slightly different pattern than the others. Only one gene from each group is shown. The details of the genes are found in the table of supplementary information.

ten genes showed patterns similar to (i), which possibly implies that ALLB might be closer to AML than ALLT. Whereas, only a few genes matched with the patterns in (ii), (iii) and (iv). This explains essentially why the predictive genes for ALL/AML differentiation in Golub *et al.* (1999) do not overlap any of the top 20 genes.

We applied the MSVM to the data with two different kernel functions, and tuning methods. The Gaussian kernel $\exp(-\frac{\|x_1-x_2\|^2}{2\sigma^2})$ and the linear kernel $x_1^t x_2$ were the choice of $K(x_1, x_2)$. We compared two tuning methods; LOOCV, and GACV. Table 1 summarizes the classification results. The first column is the number of genes, with indication of the applied preprocessing procedure (A) or (B). For each kernel function in the second column, a grid search was made for λ in the linear case, and (λ, σ) jointly in the Gaussian case for tuning. Typically, GACV gives a unique minimizer, which is a part of the LOOCV multiple minima. The misclassification counts out of 34 test samples are in the last two columns. When there were multiple equally good tuning parameters, the average performance was reported. None of the tuning methods gave a dominantly better result than the other. Comparable or even smaller test error rate was achieved using only 40 genes when we preprocessed the data according to (B). Not directly comparable to the test errors (0 to 5) for ALL/AML problem (Golub *et al.*, 1999; Furey *et al.*, 2000; Mukherjee *et al.*, 1999), the performance of the MSVM appears encouraging, given that multiclass problems are harder than binary problems.

Table 1. Classification results for the leukemia data

No. of genes (preprocessing)	Kernel function	Test errors GACV	LOOCV
50	Gaussian	4	6
(A)	Linear	4	4
100	Gaussian	1	1
(A)	Linear	2	2.25
40	Gaussian	1	0.8
(B)	Linear	1	1

Small round blue cell tumors data

Khan *et al.* (2001) successfully diagnosed the small round blue cell tumors (SRBCTs) of childhood into four classes;

neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL) and the Ewing family of tumors (EWS) using Artificial Neural Networks. The data set contains 2308 genes out of 6567 after filtering for a minimal level of expression. The training set consists of 63 samples (NB: 12, RMS: 20, BL: 8, EWS: 23), and the test set has 20 SRBCT samples (NB: 6, RMS: 5, BL: 3, EWS: 6) and five non-SRBCTs. Note that Burkitt lymphoma (BL) is a subset of NHL. Yeo and Poggio (2001) applied nearest neighbor, weighted voting and linear SVM in one-vs-rest fashion to this data. In the paper, perfect classification was possible in testing the blind 20 samples as well as cross validating 63 training samples, with five to 100 genes for each binary classifier.

For comparison, we applied the MSVM to the problem. We took logarithm base 10 of the expression levels and standardized arrays before applying the classification method. Most of the top 20 genes were consistently selected from the top 96 genes in Khan *et al.* (2001). However, the list included four additional genes, which are *neurofibromin 2*, *Isg20*, *cold shock domain protein A*, and *WASP*, and their biological functions are poorly characterized. Table 2 is a summary of the results. Although the linear kernel could achieve similar performances, we chose flexible Gaussian kernel which is particularly effective for multiclass problems. The second column presents the tuning parameters λ and σ on log 2 scale chosen by the GACV. Again, the minimizer of GACV turned out to be a part of the LOOCV tuning error minima. The proposed MSVMs were cross validated for the training set with zero LOOCV error attained for 20, 60 and 100 genes. The test results are given in the last column. Using the top ranked 20, 60 and 100 genes, the MSVMs correctly classified 20 test examples. With all the genes included, one error occurred in LOOCV and the misclassified example was identified as EWS-T13, which frequently occurred as an LOOCV error (Khan *et al.*, 2001; Yeo and Poggio, 2001). The test error using all genes varied from 0 to 3 depending on tuning measures. The MSVM tuned by GACV gave three test errors while LOOCV tuning gave 0 to three test errors. High dimensional data oftentimes reside in a low dimensional subspace. In order to visualize the data approximately in a much lower dimension, we conducted the principal component analysis. Figure 3 displays the three principal components of the top 100 genes. Notice that the principal coordinates of five non-SRBCTs land on 'no man's land', encircled by the samples from the four known classes. The three principal components contain total 66.5% variation of 100 genes in the training set. They contribute 27.52, 23.12 and 15.89%, each and the fourth component explains only 3.48% of variation. With the three principal components (PCs) only, we applied the MSVM, and the corresponding classification result is in the last row of Table 2. Again, perfect classification was achieved in cross validating and testing. Indeed, we have checked that the quadratic discriminant analysis, which could not be applied when the dimension of input space exceeds the sample size, gives the same zero test error once the data are represented by three PCs. Figure 4 shows the predicted decision vectors (f_1, f_2, f_3, f_4) at the test samples for the MSVM with the three PCs. For example, the blue bars correspond to EWS samples, and their ideal decision vector is $(1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3})$. The estimated decision vectors are pretty close to the ideal class code and their maximum components are the first one, yielding correct classification. The plot confirms that all the 20 test examples from four classes are classified correctly. Note that the test examples are rearranged in the order of EWS, BL, NB, RMS, and non-SRBCT. The last five decision vectors are for the five non-SRBCT samples. In clinical settings, it is important to be able to reject classification whenever samples in fact do not fall into the known classes. It is shown that the MSVM outputs are specific enough to identify the five non-SRBCTs. The last panel depicts the loss g, evaluated at each test sample for the MSVM prediction. The dotted line indicates the threshold of rejecting a prediction. It was set at 0.2171, which is a jackknife estimate of the 95th percentile of the loss distribution from 63 correct predictions. The losses corresponding to the predictions of five non-SRBCTs all exceed the threshold, while three test samples out of 20 can not be classified confidently by thresholding.

Table 2. LOOCV and Test errors for SRBCT data

No. of genes	Tuning parameters $\log_2 \lambda$, $\log_2 \sigma$	No. of errors LOOCV	Test
20	−22, 1.4	0	0
60	−23, 2.4	0	0
100	−23, 2.6	0	0
all	−25, 4.8	1	0–3
3 PCs	−19, 1.6	0	0

CONCLUSION

We demonstrated that the MSVM can classify cancer types accurately based on gene expression profiles. For the leukemia data, the MSVM resulted in 0 to 1 test error at best when the profiles were appropriately preprocessed. This accuracy is comparable to 1 to 3 median test errors of other methods in a slightly different study design (Dudoit *et al.*, 2002). Additionally, inspecting the patterns of highly relevant genes to the class separation revealed that B-cell ALL might be closer to AML than T-cell ALL. With various combinations of genes, the proposed method yielded perfect or near perfect classification for

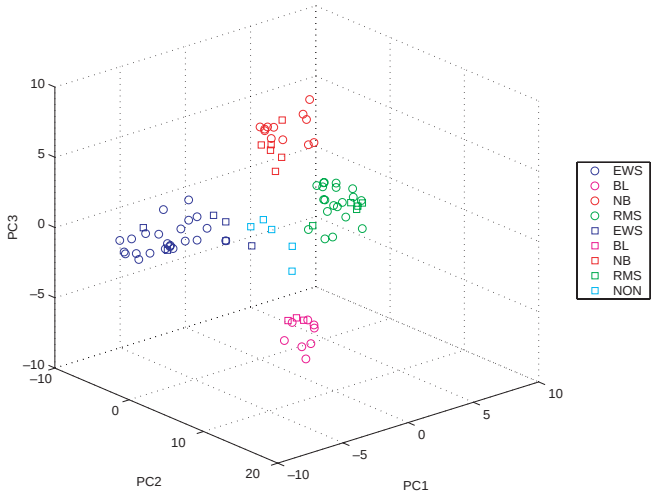

Fig. 3. Three principal components of 100 gene expression levels in the training set are plotted as circles. The squares are the corresponding principal coordinates of the test samples including non-SRBCTs. The tumor types are distinguished by colors. Three principal components show a nice separation of the four tumor types.

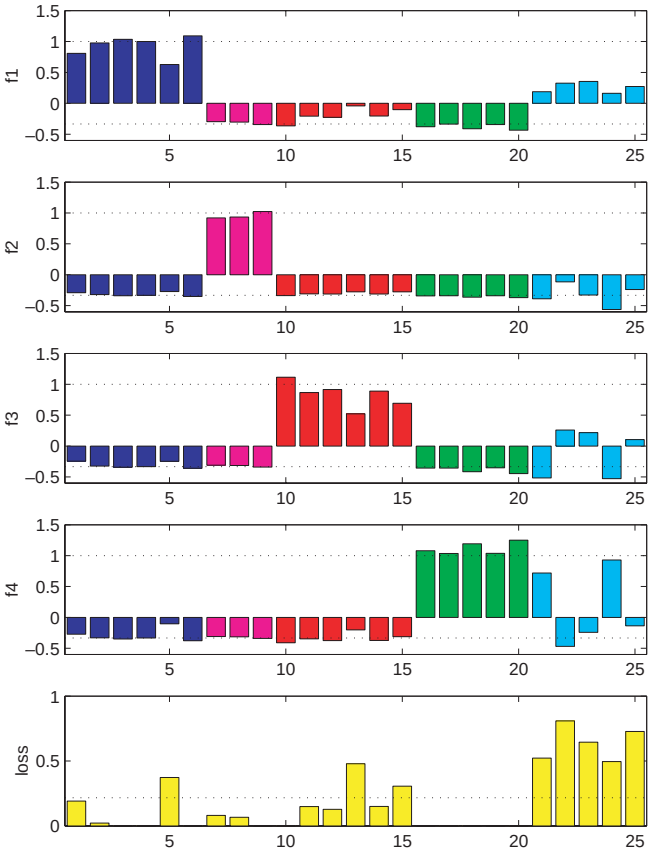

Fig. 4. The first four panels show the decision vectors (f_1, f_2, f_3, f_4) at the test samples. The four classes are coded as EWS in blue: $(1, -\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3})$, BL in purple: $(-\frac{1}{3}, 1, -\frac{1}{3}, -\frac{1}{3})$, NB in red: $(-\frac{1}{3}, -\frac{1}{3}, 1, -\frac{1}{3})$, and RMS in green: $(-\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}, 1)$. The colors indicate the true class identities. The five non-SRBCTs are plotted in cyan. The last panel depicts the loss for each decision vector. The last 5 losses for the non-SRBCTs all exceed the threshold (the dotted line) below which means a strong prediction.

the small round blue cell tumors data. The strength measure attached to each prediction turned out to be useful in identifying five non-SRBCT samples. Because of its classification accuracy and flexibility, the MSVM can be very useful for medical diagnosis problems.

In the analysis, we screened predictive genes by a marginal association between each gene and class distinction, and trained classifiers with the prescreened genes. Such marginal criterion tends to yield a set of redundant genes. It would be interesting to know how parsimonious results would be obtained if we integrate gene selection with learning. As a consequence, a reasonable number of selected genes essential for the class distinction could be printed or synthesized on customized mini-arrays for cancer diagnosis.

The MSVM treats all the classes simultaneously. If we restrict classifiers to simple ones, say, those yielding linear boundaries only, and pooling some classes into a hyperclass gives much simpler boundaries, then this simultaneous approach may not be very efficient, let alone its increased computational complexity. Nevertheless, the difficulty is how to form such hyperclasses inductively from the data. If classifiers are flexible enough to provide arbitrary boundaries, then its advantages of aggregating classes become murky. The effectiveness of the various approaches to solve multiple tumor types problems remains to be addressed as we collect more evidence.

ACKNOWLEDGEMENTS

Y. L. would like to thank Grace Wahba and Yi Lin for their helpful suggestions and discussions, and Michael Ferris for his comments and helps on computation. The anonymous referees provided many helpful suggestions. This research was partly supported by NSF Grant DMS0072292 and NIH Grant EY09946.

REFERENCES

Bredensteiner,E.J. and Bennett,K.P. (1999) Multicategory classification by support vector machines. *Comput. Optim. Appl.*, **12**, 35–46.

DeRisi,J., Penland,L., Brown,P.O., Bittner,M.L., Meltzer,P.S., Ray,M., Chen,Y., Su,Y.A. and Trent,J.M. (1996) Use of a cDNA microarray to analyse gene expression patterns in human cancer.

Nat. Genet., **14**, 457–460.

Dudoit,S., Fridlyand,J. and Speed,T. (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc.*, **97**, 77–87.

Ferris,M.C. and Munson,T.S. (1999) Interfaces to PATH 3.0: Design, implementation and usage. *Comput. Optim. Appl.*, **12**, 207–227.

Furey,T., Cristianini,N., Duffy,N., Bednarski,D., Schummer,M. and 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. *et al.* (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

Hsu,C.-W. and Lin,C.-J. (2002) A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Networks*, **13**, 415–425.

Khan,J., Wei,J.S., Ringner,M., Saal,L.H., Ladanyi,M., Westermann,F., Berthold,F., Schwab,M., Atonescu,C.R., Peterson,C. and Meltzer,P.S. (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat. Med.*, **7**, 673–679.

Lee,Y., Lin,Y. and Wahba,G. (2001) Multicategory Support Vector Machines. In *Proceedings of the 33rd Symposium on the Interface*.

Lee,Y., Lin,Y. and Wahba,G. (2002) Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Technical Report 1064*. Department of Statistics, University of Wisconsin.

Lin,Y. (2002) Support vector machines and the Bayes rule in classification. *Data Mining and Knowledge Discovery*, **6**, 259–275.

Mukherjee,S., Tamayo,P., Slonim,D., Verri,A., Golub,T., Mesirov,J. and Poggio,T. (1999) Support vector machine classification of microarray data. *Technical Report AI Memo 1677*. MIT.

Perou,C.M., Jeffrey,S.S., van de Rijn,M., Rees,C.A., Eisen,M.B., Ross,D.T., Pergamenschikov,A., Williams,C.F., Zhu,S.X., Lee,J.C. *et al.* (1999) Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl Acad. Sci. USA*, **96**, 9212–9217.

Platt,J. (1999) Probabilities for SV machines. In Smola,A., Bartlett,P., Schölkopf,B. and Schuurmans,D. (eds), *Advances in Large Margin Classifiers*. MIT Press, pp. 61–74.

Platt,J., Cristianini,N. and Shawe-Taylor,J. (2000) Large margin DAGs for multiclass classification. In Solla,S.A., Leen,T.K. and Müller,K.-R. (eds), *Advances in Neural Information Processing Systems 12*. MIT Press, pp. 547–553.

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

Wahba,G. (1990) *Spline Models for Observational Data*, Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia.

Wahba,G. (1998) Support vector machines, reproducing kernel Hilbert spaces, and randomized GACV. In Schoelkopf,B., Burges,C.J.C. and Smola,A.J. (eds), *Advances in Kernel Methods: Support Vector Learning*. MIT Press, pp. 69–87.

Yeo,G. and Poggio,T. (2001) Multiclass classification of SRBCTs. *Technical Report AI Memo 2001-018 CBCL Memo 206*. MIT.

Zhang,L., Zhou,W., Velculescu,V.E., Kern,S.E., Hruban,R.H., Hamilton,S.R., Vogelstein,B. and Kinzler,K.W (1997) Gene expression profiles in normal and cancer cells. *Science*, **276**, 1268–1272.