BMC Bioinformatics

BioMed Central

Research article

Kernel-based distance metric learning for microarray data classification

Huilin Xiong[1] and Xue-wen Chen*[1,2]

Address: [1]Bioinformatics and Computational Life Sciences Laboratory, Department of Electrical Engineering and Computer Science, University of Kansas, 2335 Irving Hill Road, Lawrence, Kansas 66045, USA and [2]Kansas Masonic Cancer Research Institute, Kansas City, Kansas, USA

Email: Huilin Xiong - hlxiong@ittc.ku.edu; Xue-wen Chen* - xwchen@ku.edu

* Corresponding author

Published: 14 June 2006

BMC Bioinformatics 2006, **7**:299 doi:10.1186/1471-2105-7-299

Received: 05 December 2005
Accepted: 14 June 2006

This article is available from: http://www.biomedcentral.com/1471-2105/7/299

Abstract

Background: The most fundamental task using gene expression data in clinical oncology is to classify tissue samples according to their gene expression levels. Compared with traditional pattern classifications, gene expression-based data classification is typically characterized by high dimensionality and small sample size, which make the task quite challenging.

Results: In this paper, we present a modified K-nearest-neighbor (KNN) scheme, which is based on learning an adaptive distance metric in the data space, for cancer classification using microarray data. The distance metric, derived from the procedure of a data-dependent kernel optimization, can substantially increase the class separability of the data and, consequently, lead to a significant improvement in the performance of the KNN classifier. Intensive experiments show that the performance of the proposed kernel-based KNN scheme is competitive to those of some sophisticated classifiers such as support vector machines (SVMs) and the uncorrelated linear discriminant analysis (ULDA) in classifying the gene expression data.

Conclusion: A novel distance metric is developed and incorporated into the KNN scheme for cancer classification. This metric can substantially increase the class separability of the data in the feature space and, hence, lead to a significant improvement in the performance of the KNN classifier.

Background

DNA microarray technology is designed to measure the expression levels of tens of thousands of genes simultaneously. As an important application of this novel technology, the gene expression data are used to determine and predict the state of tissue samples, which has shown to be very helpful in clinical oncology. The most fundamental task using gene expression data in clinical oncology is to classify tissue samples according to their gene expression levels. In combination with pattern classification techniques, gene expression data can provide more reliable means to diagnose and predict various types of cancers than the traditional clinical methods.

Compared with traditional pattern classifications, gene expression-based data classification is typically characterized by high dimensionality and small sample size, which make the task quite challenging. In the literature, a number of methods have been applied or developed to classify microarray data [1-6]. These methods include K-nearest-neighbor (KNN), boosting, linear discriminant

analysis (LDA), and support vector machines (SVM), etc. we herein briefly review some of the approaches.

K-Nearest-Neighbor (KNN)

The KNN method is a simple, yet useful approach to data classification. The error rate of the KNN has been proven to be asymptotically at most twice that of the Bayessian error rate [7]. However, its performance deteriorates dramatically when the input data set has a relatively low local relevance [8]. The most important factor impacting the performance of KNN is the distance metric. It is desirable to adopt an appropriate distance metric for the KNN algorithm. In practice, the Euclidean distance is usually used as the distance metric.

Diagonal Linear Discriminant Analysis (DLDA)

DLDA is the simplest case of the maximum likelihood discriminant rule, in which the class densities are supposed to have the same diagonal covariance matrix. In the special case of binary classification, the DLDA scheme can be viewed as the "weighted voting scheme" proposed by Golub *et al*. in [3]. The major advantage of the DLDA algorithm lies in its computational efficiency.

Linear Discriminant Analysis (LDA)

The classical LDA method aims to find the most discriminatory projection directions of the input data and classifies the data in the projected space. A major problem in employing the classical LDA algorithm for classifying gene expression data is that the so called scatter matrices are always singular, due to the nature of high dimensionality and relatively small sample size. The singularity makes the classical LDA algorithm inapplicable. In the areas such as face recognition and text classification, the principal component analysis (PCA) technique is introduced as a preprocessing procedure in order to reduce the dimensionality of the input data. However, since the projection criterion of PCA is essentially different from that of LDA, losing discriminatory information in the PCA step becomes inevitable. A recent development in LDA is the generalized discriminant analysis [9,10], in which a more delicate matrix technique, namely, the generalized singular value decomposition (GSVD), is used to modify the classical LDA into a more general version.

Support Vector Machines (SVM)

SVM has been recognized as the most powerful classifier in various applications of pattern classification. For binary classification, SVM searches for a hyperplane that separates the two classes of data with the maximum margin. It has been shown that support vector machines perform well in many areas of computational biology [11-13]. In the experimental part of this paper, we follow the way in [14] to implement the SVM algorithm.

Generally speaking, due to the high dimensionality and small sample size, linear classifiers such as the linear discrimiant analysis (LDA), and the support vector machines (SVM) with linear kernels are used favorably. However, based on some benchmark tests, researchers have shown that nonlinear classfiers are capable of exploring the nonlinear discriminatory information in the microarray data, and usually produce more precise classification results [15,16]. This is especially true when more patients' samples are available or the data dimension is substantially reduced, since, in these cases, the linear separability of the microarray data could be considerably degraded.

Among the general algorithms of pattern classification, K-nearest-neighbor (KNN) is a simple yet useful one. However, in practice, the performance of KNN algorithm is often inferior to those of the sophisticated approaches such as SVM and generalized linear discriminant analysis (GLDA) [9,10]. Since the distance metric is of great importance for the KNN scheme, an attractive way to improve the performance of KNN is to adopt a more adaptive distance metric to the input data than the Euclidean diatnce. In this paper, we propose to learn the adaptive distance metric via optimizing a data-dependent kernel. Experimental results show that, compared with the ordinary Euclidean distance-based KNN scheme, our kernel-based KNN algorithm, denoted KerNN, always achieves significant improvement in the performance of classifying gene expression data. Moreover, the performance of the KerNN classifier is shown to be competitive, if not better, to those of the sophisticated classifiers, e.g., SVM and the uncorrelated linear discriminant analysis (ULDA) [10], in classifying microarray data.

Results

We conducted intensive experiments to compare the performances of our KerNN scheme to the commonly-used classification algorithms, i.e., KNN, DLDA [3], ULDA [10], and SVM. Ten publicly available microarray data sets were chosen to test our algorithms. The basic information about these data sets is summarized below. Each data set is first normalized to a distribution with zero mean and unity variance in every feature direction, and then, randomly partitioned into two disjoint subsets with equal number of samples, one is used as the training data, and the other the test data. We only consider Gaussian kernel function in the proposed and SVM algorithms.

1. *ALL-AML Leukemia Data*: This data set, taken from the website [17], contains 72 samples of human acute leukemia. 47 samples belong to acute lymphoblastic leukemia (ALL), and the other acute myeloid leukemia (AML). Each sample presents the expression levels of 7129 genes. For the detailed information, one can refer to [3].

2. *ALL-MLL-AML Leukemia Data*: This leukemia microarray data set is available on the website [17]. It includes 72 human leukemia samples, 24 of them belong to acute lymphoblastic leukemia (ALL), 20 of them to mixed lineage leukemia (MLL), a subset of human acute leukemia with a chromosomal translocation, and 28 of the samples are acute myelogenous leukemia (AML). Each sample gives the expression levels of 12582 genes. Further information about this data set can be found in [21].

3. *Embryonal Tumors of the Central Nervous System (CNS)*: This data set, available at the website [17], contains 60 patient samples, 21 are survivors of a treatment, and 39 are failures. There are 7129 genes in the data set. One can refer to [22] to find more information about this data set.

4. *Breast Cancer Data*: The data are available on the website [18]. The expression matrix monitors 7129 genes in 49 breast tumor samples. There are two response variables respectively describing the status of the estrogen receptor (ER) and the lymph nodal (LN) status. For the ER status, 25 samples are ER+, whereas the remaining 24 samples are ER-. For the LN variable, there are 25 positive sample and 24 negative samples. The detailed information about this data set can be found in [6].

5. *Colon Tumor Data*: This data set is adopted from the website [17]. The data contain 62 samples collected from colon-cancer patients. Among them, 40 samples are from tumors, and 22 normal biopsies are from healthy parts of the colons of the same patients. 2000 genes were selected to measure their expression levels. One can refer to [23].

6. *Lung Cancer Data*: This data set is taken from the website [17]. It contains 181 tissue samples, which are classified into two classes: malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA). Each sample is described by 12533 genes. More information about this data set can be found in [24].

7. *Lymphoma Data*: The data are available on the website [19]. This data set contains 77 tissue samples, 58 are diffuse large B-cell lymphomas (DLBCL) and the remaining 19 samples are follicular lymphomas (FL). Each sample is represented by the expression levels of 7129 genes. The detailed information about this data set can be found in [25].

8. *Ovarian Cancer Data*: This data set, available on the website [17], is to distinguish ovarian cancer from non-cancer. It contains 253 samples, and each sample has 15154 features. More details can be found in [26].

9. *Prostate Cancer Data*: This data set, adopted from the website [19], contains the gene expression levels of 12600 genes for 52 prostate tumor samples and 50 normal prostate samples. One can refer [4] for the details about this data set.

10. *Subtypes of Acute Lymphoblastic Leukemia*: This data set, available on the website [20], contains 6 subtypes of pediatric acute lymphoblastic leukemia, corresponding to six diagnostic groups: BCR-ABL, E2A-PBX1, MLL, T-ALL, TEL-AML1, Hyperdiploid>50. Each sample contains 12625 genes.

Comparisons in terms of the best results

For each data set, we chose the N_f most discriminatory genes, where N_f = 10, 20, 40, 60, 80, 100, 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, respectively; repeated the experiment 100 times at each value of N_f; and then, calculated the average test error rates and their standard deviations over the 100 experiments. Table 1 lists the best results, i.e., the smallest average test error rate, of different algorithms. It can be seen that the proposed KerNN algorithm reaches the best, which are in bold face, on four data sets. On the other data sets, the performance of the KerNN algorithm is still competitive, if not better, to those of the SVM and ULDA schemes.

In Table 1, if we assign a score 1 to the best result, 2 to the next best result, ..., and so on, then, the global performance of a classifier can be roughly evaluated in terms of the average score. We show the average scores of the five classifiers in Table 1. It can be seen that the proposed KerNN scheme achieves the lowest score among the five classifiers.

Comparisons under different gene numbers

To investigate the stability of the 5 classification algorithms, we compared their performances when different number of genes were selected. The experimental results are shown in Fig. 1, for the *ALL-AML* data, Fig. 2, for the *Colon* data, and Fig. 3, for the *Prostate* data, where the horizontal axis is the number of the selected genes and the vertical axis is the average test error rates of the classifiers over 100 experiments. While Fig. 1 (a), Fig. 2 (a), and Fig. 3 (a) illustrate the results in the case of choosing a relatively small number of features (from 10 to 100), Fig. 1 (b), Fig. 2 (b), and Fig. 3 (b) demonstrate the corresponding results when more genes are chosen (from 200 to 2000). It can be seen that the proposed KerNN scheme performs favorably in most cases. Compared with the ULDA scheme, which always performs poorly in the case of small feature size, and the DLDA algorithm, whose performances usually degrade for relatively large feature size, our KerNN algorithm works with more stability with different feature numbers. More importantly, compared with the ordinary KNN classifier, the kernel optimization-based KNN classifier always gains significant improve-

Table 1: Comparison of the classifiers in terms of the best results. The comparison of all the classifiers in terms of the best results of the average test error rates (%). For each data set, we chose the N_f most discriminatory genes, where N_f = 10, 20, 40, 60, 80, 100, 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000, respectively; repeated the experiment 100 times at each value of N_f; and then, calculated the average test error rates and their standard deviations over the 100 experiments. In comparison, we assign a classifier a score 1 as it achieves the best result on one data set, and 2 if it achieves the next best result, and so on. The average score roughly evaluates the global performance of a classifier on these twelve data sets.

	KNN	ULDA	DLDA	SVM	KerNN
ALL-AML	3.32 (1.21)	3.08 (1.09)	2.95 (0.78)	**2.70** (0.00)	**2.70** (0.00)
ALL-MLL-AML	6.17 (2.75)	**2.14** (1.97)	5.19 (2.95)	2.83 (2.37)	3.21 (2.18)
CNS	19.52 (5.88)	**12.26** (7.04)	22.42 (5.58)	13.35 (7.52)	15.32 (5.60)
Breast-ER	7.12 (4.12)	4.92 (4.40)	**3.21** (3.04)	4.64 (4.39)	4.48 (4.45)
Breast-LN	13.12 (5.91)	9.92 (5.16)	**7.76** (4.85)	7.92 (5.39)	8.36 (4.48)
Colon	14.03 (3.76)	16.84 (6.14)	12.65 (4.58)	11.84 (4.28)	**11.58** (4.97)
Lung	1.21 (0.98)	0.81 (0.73)	0.47 (0.57)	0.53 (0.61)	**0.31** (0.54)
Lymphoma	2.05 (2.58)	2.05 (2.09)	6.23 (2.88)	**1.03** (1.59)	1.90 (2.05)
Ovarian	0.74 (0.87)	0.02 (0.13)	1.58 (0.81)	0.17 (0.42)	**0.01** (0.08)
Prostate	7.41 (2.47)	5.22 (2.99)	6.73 (3.02)	**4.86** (2.77)	4.90 (2.53)
Subtypes	2.57 (0.86)	**1.73** (0.90)	2.45 (0.92)	2.60 (1.02)	2.42 (0.82)
Average Score	4.5	2.8	3.3	2.3	**1.9**

ments, which implies that the procedure of kernel optimization induces a distance metric that adapts better than the Euclidean metric to the gene expression data in the data space.

Discussion

Parameter tuning

In the experiments, for KNN, ULDA, and the proposed algorithm, the final classification is done via the K-nearest-neighbor algorithm with K = 3. For KNN, ULDA, and DLDA algorithms, the only parameter is the number of selected genes N_f. For SVM, in addition to the gene number, two parameters, the γ in the Gaussian kernel function and the regulation constant C, need to be set in advance. As for the KerNN algorithm, there are more parameters. To avoid the intensive computation in parameter tuning using the cross validation, we respectively chose the N_f most discriminatory genes, where N_f = 10, 20, 40, 60, 80, 100, 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000. The best performance for each method is reported in Table 1. For our kernel optimization method, the initial learning rate η_0 and the total iteration number N are always set to 0.01 and 1000 respectively. Furthermore, for the sake of computational simplicity, we empirically set the two Gaussian parameters in the proposed method as $\gamma_0 = \frac{10^{-5}}{\sqrt{N_f}}$ and $\gamma_1 = \frac{10^{-2}}{\sqrt{N_f}}$, rather than tune them by the cross validation. This may not be the optimal settings for the parameters γ_0 and γ_1. However, high computational complexity can be avoided. It is expected that even better results could be obtained if we were to choose them by the cross validation. Therefore, for the KerNN method, there is only one parameter σ_ε, the standard variance of the disturbance added to the data in Eq. (10), that need to be tuned. As to the SVM, two parameters are tuned by the cross validation.

In the experiments, we employed the leave-one-out technique on the training data to choose these parameters. We followed [14] to implement the SVM algorithm, in which the parameter C is chosen from {1.0e+00, 1.0e+01, 1.0e+02, 1.0e+03, 1.0e+04, 1.0e+05, 1.0e+06, 1.0e+07} and γ from {1.0e-07, 5.0e-07, 1.0e-06, 5.0e-06, 1.0e-05, 5.0e-05, 1.0e-04, 5.0e-04, 1.0e-03, 5.0e-03, 1.0e-02} using the leave-one-out cross validation. For our KerNN algorithm, the parameter σ_ε is selected from {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}. Note that only the training samples were used for setting parameters. Test samples are independent of this process.

Gene selection

In this paper, we employ the BW ratio used in [2,10] to select genes. This ratio is essentially a Fisher discriminant measure. Given a gene j, the ratio on gene j is calculated as

$$g(j) = \frac{\sum_{k=1}^{p} m_k (\bar{x}_k(j) - \bar{x}(j))^2}{\sum_{k=1}^{p} \sum_{i \in C_k} (x_i(j) - \bar{x}_k(j))^2}$$

where C_k denotes the index set of the k-th class (k = 1,2,...,p), m_k is the number of samples in C_k ($\sum_{k=1}^{p} m_k = m$), $\bar{x}_k(j)$ and $\bar{x}(j)$ represent the average

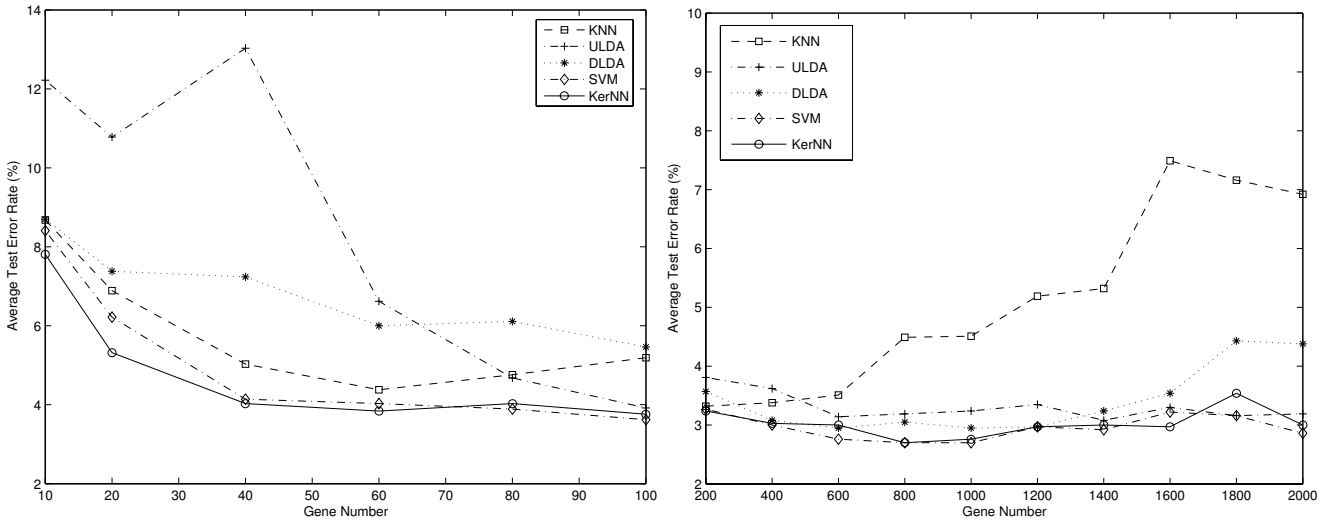

Figure 1
Test error rates for the ALL-AML data set. Stability comparison on the *ALL-AML* data. The average test error rate (%) as a function of the selected gene number.

expression levels cross the *k*-th class and whole training samples on gene *j*, respectively.

Gene selection usually has a strong impact on the performances of various classifiers, due to the effect of correlation between genes. Our experiments show that the impact can be considered in two aspects: 1)with different numbers of genes, the performance of a classifier could be remarkably different. For example, the ULDA method usually works quite well as a large number of genes is used, but performs poorly in the case of small gene number. Contrarily, the DLDA classifier often reaches its best performance at small number of features. 2) with different numbers of genes, the model parameters, especially for the nonlinear methods, need to be set differently to achieve better result.

The effect of the disturbed resampling

Due to the lack of enough training samples, the scheme of the kernel optimization-based classification may lead to an overfitting result in classifying gene expression data. To

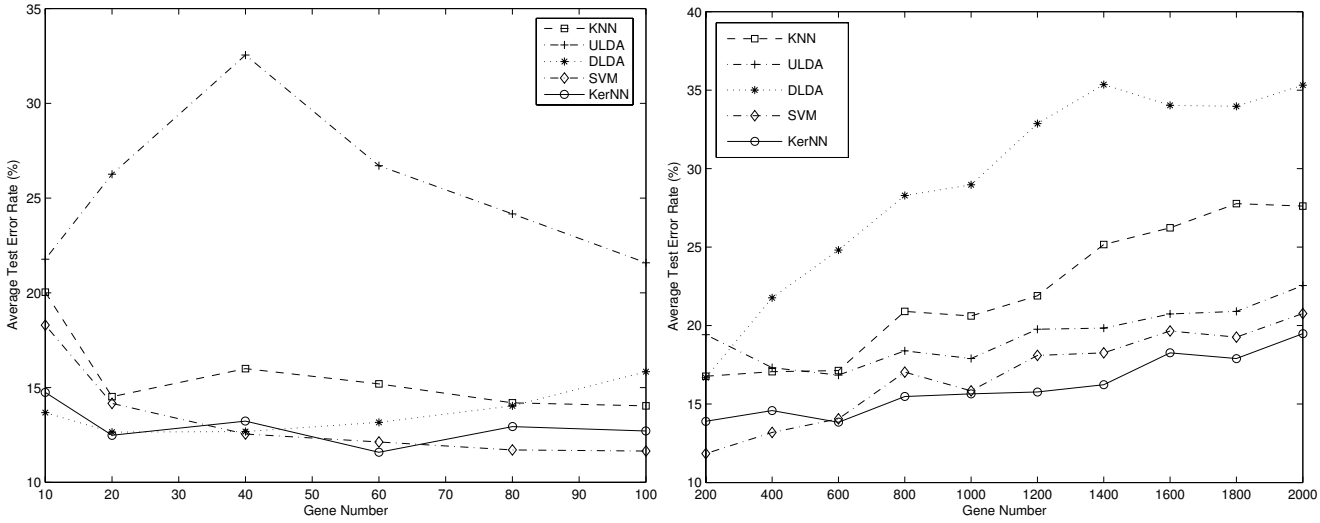

Figure 2
Test error rates for the Colon data set. Stability comparison on the *Colon* data. The average test error rate (%) as a function of the selected gene number.

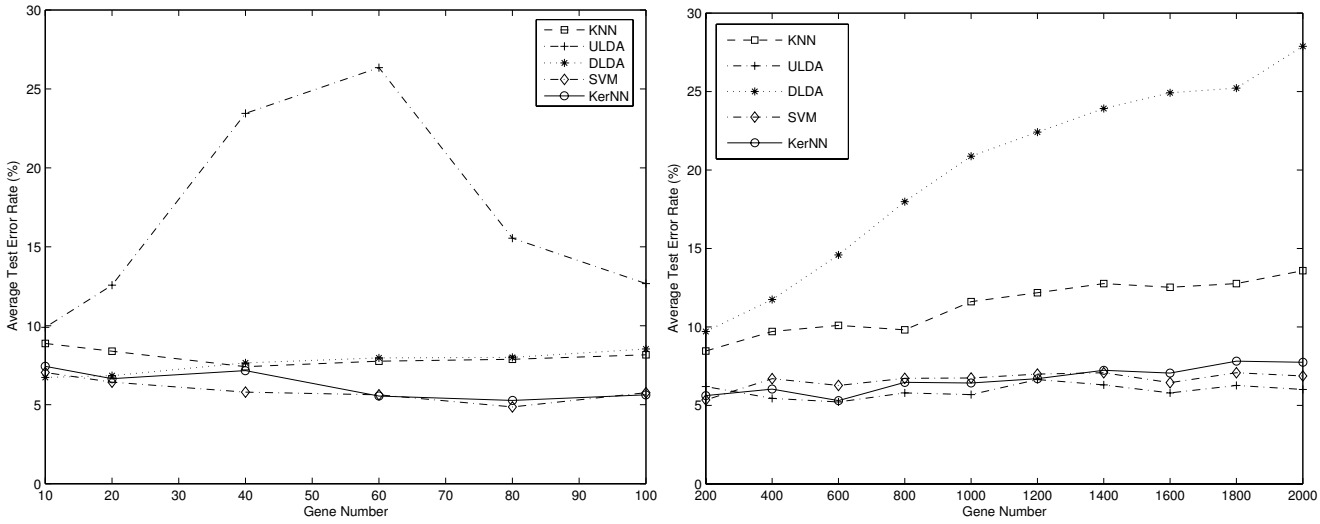

Figure 3
Test error rates for the Prostate data set. Stability comparison on the *Prostate* data. The average test error rate (%) as a function of the selected gene number.

alleviate the possible overfitting, a strategy of disturbed resampling, as shown in Eq. (10), was adopted. In this section, we demonstrate that using this strategy, the overfiting could be effectively reduced.

In the case that there are relatively large number of samples, the kernel optimization-based KNN classifier without using the strategy of disturbed resampling, denoted by KerNN0, usually works well on both the training and test data. Fig. 4 illustrates the performances of KNN, KerNN0, and KerNN on both the training and test data of the *Prostate* data set, which includes 102 samples. It can be seen that, compared with the KNN algorithm, both the KerNN0 and KerNN methods gain significant improvements, not only on the training data, but also on the test data. However, when the sample size is relatively small, the KerNNO algorithm may lead to serious overfitting. We choose the *Breast-ER* data set, which contains only 49 samples, to demonstrate our argument. Fig. 5 (a) shows the average error rates of KNN, KerNN0, and KerNN algorithms on the training data, and Fig. 5 (b) presents the corresponding results on the test data. It can be seen that, although KerNN0 works quite well on the training data, its performance degrades remarkably on the test data. On the contrary, for the KerNN scheme, no overfitting occurred.

Conclusion

In this paper, a novel distance metric is developed and incorporated into a KNN scheme for cancer classification. This metric, derived from the procedure of a data-dependent kernel optimization, can substantially increase the class separability of the data in the feature space, and hence, lead to a significant improvement in the performance of the KNN classifier. Furthermore, in combination with a disturbed resampling strategy, the kernel optimization-based K-nearest-neighbor scheme can achieve competitive performance to the fine tuned SVM and the uncorrelated linear discriminant analysis (ULDA) scheme in classifying gene expression data. Experimental results show that the proposed scheme performs with more stability than the ULDA scheme, which works poorly in the case of small feature size, and the DLDA scheme, whose performance usually degrades in the case of a relatively large feature size.

Methods

0.1 Data-dependent kernel model

In this paper, we employ a special kernel function model, which is called date-dependent kernel model, as the objective kernel to be optimized. Apparently, there is no benefit at all if we simply use the common kernel such as the Gaussian kernel or the polynomial kernel in the KNN scheme, since the distance ranking in the Hilbert space derived from the kernel function is the same as that in the input data space. However, when we adopt the data-dependent kernel, especially after the kernel is optimized, the distance metric could be appropriately modified so that the local relevance of the data is significantly improved.

Let $\{x_i, \zeta_i\}$ $(i = 1,2, ..., m)$ be m d-dimensional training samples of the given gene expression data, where ζ_i represent the class labels of the samples. We refer the data-dependent kernel as,

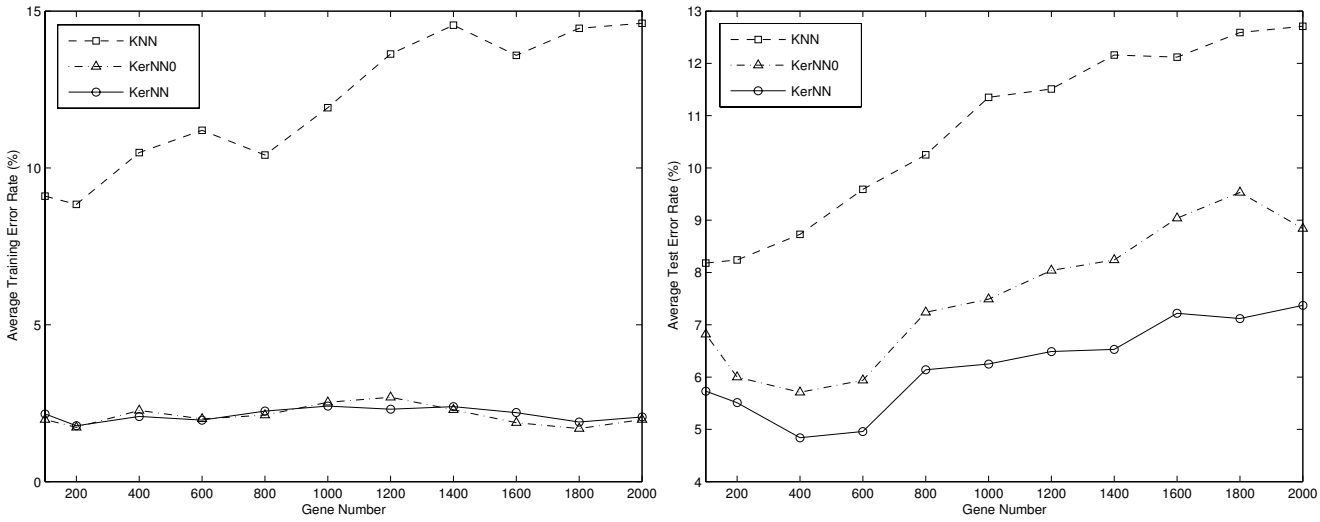

Figure 4
The effect of the disturbed resampling on Prostate. The effect of adopting the technique of disturbed resampling on a relatively large data set, *Prostate*, which contains 102 samples. (a) Results on the training data. (b) Results on the test data.

$$k(x, y) = q(x)q(y)k_0(x, y) \qquad (1)$$

where $x, y \in \mathbf{R}^d$, $k_0(x, y)$, called the basic kernel, is an ordinary kernel such as a Gaussian or a polynomial kernel function, and $q(.)$, the factor function, takes the form as

$$q(x) = \alpha_0 + \sum_{i=1}^{l} \alpha_i k_1(x, a_i) \qquad (2)$$

in which $k_1(x, a_i) = e^{-\gamma_1 ||x-a_i||^2}$, α_i's are the combination coefficients, and a_i's denote the local centers of the training data.

Let the kernel matrices corresponding to $k(x, y)$ and $k_0(x, y)$ be K and k_0. Obviously, $K = [q(x_i)q(x_j)k_0(x_i, x_j)]_{m \times m} = QK_0Q$, where Q is a diagonal matrix whose diagonal elements are $q(x_1)$, $q(x_2)$,...,$q(x_m)$. Let us denote the vector $(q(x_1), q(x_2),..., q(x_m))^T$ and $(\alpha_0, \alpha_1, \alpha_2,..., \alpha_n)^T$ by q and α respectively, we have $q = K_1\alpha$, where K_1 is an $m \times (l + 1)$ matrix

$$K_1 = \begin{pmatrix} 1 & k_1(x_1, a_1) & \cdots & k_1(x_1, a_l) \\ 1 & k_1(x_2, a_1) & \cdots & k_1(x_2, a_l) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & k_1(x_m, a_1) & \cdots & k_1(x_m, a_l) \end{pmatrix} \qquad (3)$$

0.2 Kernel optimization for binary-class data

We optimized the data-dependent kernel in Eq.(l). This requires optimizing the combination coefficient vector α, aiming to increase the class separability of the data in the feature space. A Fisher scalar measuring the class separability of the training data in the feature space is adopted as a criterion for our kernel optimization

$$J = \frac{\mathrm{tr}(S_b)}{\mathrm{tr}(S_w)} \qquad (4)$$

where S_b represents the "between-class scatter matrix", and S_w "within-class scatter matrix".

Suppose that the training data are grouped according to their class labels, i.e., the first m_1 data belong to one class, and the remaining m_2 data belong to the other class ($m_1 + m_2 = m$). Then the basic kernel matrix k_0 can be partitioned as

$$K_0 = \begin{pmatrix} K_{11}^0 & K_{12}^0 \\ K_{21}^0 & K_{22}^0 \end{pmatrix} \qquad (5)$$

where the sizes of the submatrices $K_{11}^0, K_{12}^0, K_{21}^0$, and K_{22}^0 respectively are $m_1 \times m_1$, $m_1 \times m_2$, $m_2 \times m_1$, and $m_2 \times m_2$. A close relation between the class separability measure J and the kernel matrices can be established [27].

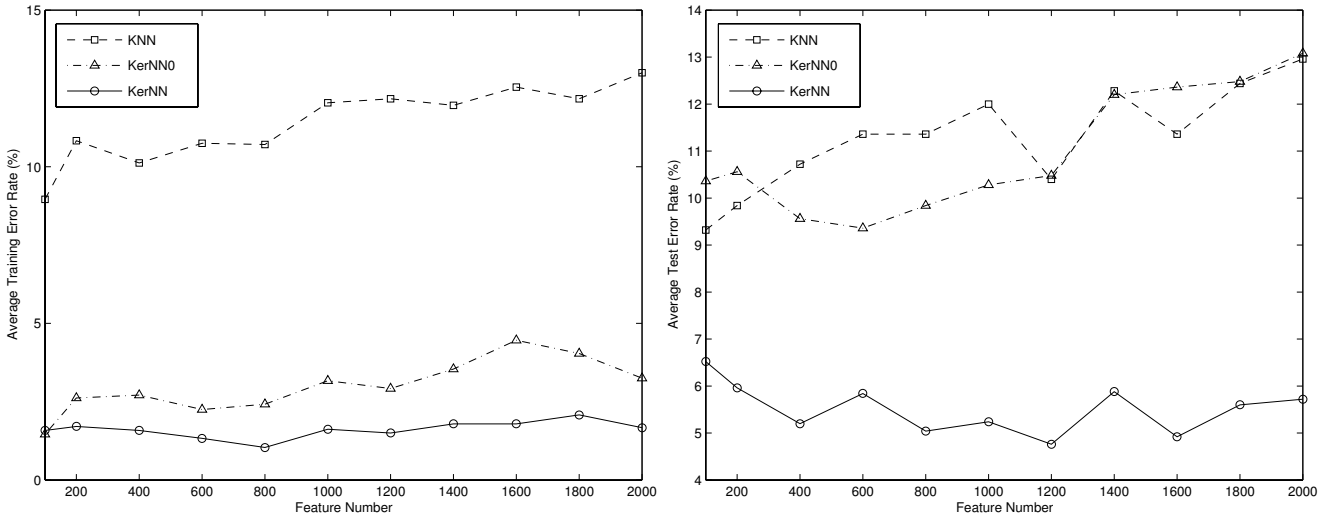

Figure 5

The effect of the disturbed resampling on Breast-ER. The effect of adopting the technique of disturbed resampling on the a relatively small data set, *Breast-ER*, which contains only 49 samples. (a) Results on the training data. (b) Results on the test data.

$$J(\alpha) = \frac{\alpha^T M_0 \alpha}{\alpha^T N_0 \alpha} \qquad (6)$$

where $M_0 = K_1^T B_0 K_1$, $N_0 = K_1^T W_0 K_1$, in which

$$B_0 = \begin{pmatrix} \frac{1}{m_1} K_{11}^0 & 0 \\ 0 & \frac{1}{m_2} K_{22}^0 \end{pmatrix} - \frac{1}{m} K_0$$

$$W_0 = \mathrm{diag}\left(k_{11}^0, k_{22}^0, \ldots, k_{mm}^0\right) - \begin{pmatrix} \frac{1}{m_1} K_{11}^0 & 0 \\ 0 & \frac{1}{m_2} K_{22}^0 \end{pmatrix}$$

To avoid using the eigenvector solution, an updating algorithm based on the standard gradient approach is developed. This algorithm is summarized below, in which the learning rate $\eta(n)$ is adopted in a gradually decreasing form as

$$\eta(n) = \eta_0 (1 - \frac{n}{N}) \qquad (7)$$

where η_0 represents an initial learning rate.

1. Group the data according to their class labels. Calculate K_0 and K_1 first, then B_0 and W_0, and then M_0, N_0.

2. Initialize $\alpha^{(0)}$ by a vector $(1,0,\ldots,0)^T$, and set $n = 0$.

3. Calculate $q = K_1 \alpha^{(n)}$, and $J_1 = q^T B_0 q$, $J_2 = q^T W_0 q$, and J.

4. Update $\alpha^{(n)}$:

$$\alpha^{(n+1)} = \alpha^{(n)} + \eta(n)(\frac{1}{J_2} M_0 - \frac{1}{J_2} N_0)\alpha^{(n)}$$

and normalize $\alpha^{(n+1)}$ so that $||\alpha^{(n+1)}|| = 1$.

5. If n reaches a pre-specified number N, stop. Otherwise, set $n = n + 1$, go to 3.

0.3 Kernel optimization for multi-class data

In the case of multi-class data, we decompose the problem of kernel optimization into a series of binary-class kernel optimizations.

Let $(x_i, \zeta_i) \in \mathbf{R}^d \times \zeta\ (i = 1, 2, \ldots, m)$ be the training data set containing p classes, that is, $\zeta = \{1,2,\ldots,p\}$. We assume the data to be grouped in order, that is, the first m_1 data belong to the first class, the next m_2 data belong to the sec-

ond class, and so on, where $\sum_{i=1}^{p} m_i = m$. Then, the kernel matrix can be written as

$$K = \begin{pmatrix} K_{11} & K_{12} & \cdots & K_{1p} \\ K_{21} & K_{22} & \cdots & K_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ K_{p1} & K_{p2} & \cdots & K_{pp} \end{pmatrix} \quad (8)$$

where the submatrix k_{ij} is of size $m_i \times m_j$, and K_{ii} represents the kernel matrix corresponding to the data in the i-th class. The class separability of the i-th and j-th class, denoted by J^{ij} ($i,j = 1, 2,...,p, i \neq j$), is calculated as

$$J^{ij}(\alpha) = \frac{J_i^{ij}}{J_2^{ij}} = \frac{1_{m_i+m_j}^T B^{ij} 1_{m_i+m_j}}{1_{m_i+m_j}^T W^{ij} 1_{m_i+m_j}} \quad (9)$$

where the between-class and within-class kernel scatter matrices B^{ij} and W^{ij} are defined as

$$B^{ij} = \begin{pmatrix} \frac{1}{m_i} K_{ii} & 0 \\ 0 & \frac{1}{m_j} K_{jj} \end{pmatrix} - \frac{1}{m_i + m_j} \begin{pmatrix} K_{ii} & K_{ij} \\ K_{ji} & K_{jj} \end{pmatrix}$$

$$W^{ij} = D^{ij} - \begin{pmatrix} \frac{1}{m_i} K_{ii} & 0 \\ 0 & \frac{1}{m_j} K_{jj} \end{pmatrix}$$

in which D^{ij} denotes a diagonal matrix whose diagonal elements are composed of the diagonal entries of the matrix K_{ii} and K_{jj}. We also denote the between-class and within-class kernel matrices corresponding to the basic kernel by B_0^{ij} and W_0^{ij} respectively.

In each iteration of the updating algorithm, we first find the class index (u, v) that corresponds to the minimum J^{ij} in current step, then the value of α is updated in such a way that the class separability of the u-th and v-th class J^{uv} will be maximized. In other words, the objective of the kernel optimization becomes

$$\max_{\alpha} \min_{ij} J^{ij}(\alpha)$$

It is easy to modify the kernel optimization algorithm from the case of binary class data to the case of multi-class data. The detailed kernel optimization algorithm for multi-class data set is summarized below, where Γ_{ij} denotes the union of the data index sets of the i-th and j-th class, and $q(\Gamma_{ij})$ and $K_1(\Gamma_{ij'}:)$ represent the submatrix extraction as in MATLAB.

1. Group the data according to their class labels. Calculate k_0 and K_1.

2. Initialize $\alpha^{(0)}$ by a vector $(1,0,..., 0)^T$, and set $n = 0$.

3. Calculate $q = K_1\alpha^{(n)}$, $J_1^{ij} = q(\Gamma_{ij})^T B_0^{ij} q(\Gamma_{ij})$, $J_2^{ij} = q(\Gamma_{ij})^T W_0^{ij} q(\Gamma_{ij})$, and J^{ij}, where $i, j = 1,2,...,p$, and $i \neq j$.

4. Find $(u, v) = \arg\min_{ij} J^{ij}(\alpha)$, and calculate $M_0^{uv} = K_1(\Gamma_{uv}:)^T B_0^{uv} K_1(\Gamma_{uv}:)$, and $N_0^{uv} = K_1(\Gamma_{uv}:)^T W_0^{uv} K_1(\Gamma_{uv}:)$.

5. Update $\alpha^{(n)}$

$$\alpha^{(n+1)} = \alpha^{(n)} + \eta(n)\left(\frac{1}{J_2^{uv}} M_0^{uv} - \frac{J^{uv}}{J_2^{uv}} N_0^{uv}\right)\alpha^{(n)}$$

and normalize $\alpha^{(n+1)}$ so that $||\alpha^{(n+1)}|| = 1$.

6. If n reaches a prespecified number N, stop. Otherwise, set $n = n + 1$, go to step 3.

0.4 KNN classification using the optimized kernel distance metric

Given two samples $x,y \in \mathbf{R}^d$, the inner product is defined as: $x \cdot y = \langle x, y \rangle = k(x, y)$; therefore, their derived distance can be calculated

$$d(x, y) = \langle x, x \rangle + \langle y, y \rangle - 2\langle x, y \rangle = k(x, x) + k(y, y) - 2k(x, y).$$

Using our data-dependent kernel model, the distance can be expressed as

$$d(x, y) = q^2(x) + q^2(y) - 2q(x)q(y)k_0(x, y) = [q(x) - q(y)]^2 + 2q(x)q(y)(1 - k_0(x, y))$$

where we assume that the basic kernel function satisfy: $k_0(x,x) = 1$, just like the Gaussian function.

Since the kernel optimization scheme increases the class separability of the data in the feature space, the performances of kernel machines should be improved. However, for the classification of gene expression data, due to the small size of training samples, the kernel optimization, which performs on training data, may cause overfitting,

which means the algorithm may work very well on the training data, but worse on the test data. To handle this problem, we adopted a disturbed resampling strategy to increase the sample size of the training data.

Suppose that $\{x_i, \zeta_i\}$ $(i = 1,2, \ldots m)$ are the training data, we construct a new set of training data $\{\gamma_i, \xi_i\}(i = 1,2,\ldots,3m)$, where

$$\gamma_i = \begin{cases} x_i & \text{if } 1 \le i \le m \\ x_r + \varepsilon & \text{if } i > m \end{cases} \qquad (10)$$

in which x_r is a sample randomly selected form $\{x_i\}$ with replacement and ε denotes a normal random disturb, that is, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$. The class labels are determined as

$$\xi_i = \begin{cases} \zeta_i & \text{if } 1 \le i \le m \\ \zeta_r & \text{if } i > m \end{cases}$$

Due to the very high dimensionality and small number of the patient samples, the training data are sparsely distributed in the high dimensional Euclidean space. It is reasonable to assume that the near points of a training datum have the same class characteristic as that of the training datum. Experimentally, using the technique of disturbed resampling (Eq.(l0)), we can effectively reduce the possible overfitting and computational instability, which are mainly caused by the lack of enough training samples for the gene expression data.

Abbreviations

KNN: K-nearest-Neighbor

SVM: support vector machine

DLDA: diagonal linear discriminant analysis

ULDA: uncorrelated linear discriminant analysis

KerNN: kernel optimization-based KNN

ALL: acute lymphoblastic leukemia

AML: acute myeloid leukemia

MLL: mixed lineage leukemia

CNS: embryonal tumor of central nervous system

Authors' contributions

HX and XWC conceived the study. HX designed and implemented the algorithms, and drafted the manuscript. XWC coordinated the study, participated in the algorithm design, and helped draft the manuscript.

Availability

The core source codes of our algorithms are available at http://www.ittc.ku.edu/~xwchen/BMCbioinformatics/kernel/

Acknowledgements

This investigation was based upon work supported by the National Science Foundation under Grant No. EPS-0236913, by matching support from the State of Kansas through Kansas Technology Enterprise Corporation, and by the University of Kansas General Research Fund allocations #2301770-003 and #2301478-003.

References

1. Ben-Dor A, Bruhn L, Friedman N, Nachman I, Schummer M, Yakhini Z: **Tissue classification with gene expression profiles.** *J Computational Biology* 2000, **7:**559-584.
2. Dudoit S, Fridlyand J, Speed TP: **Comparison of discrimination method for the classification of tumor using gene expression data.** *J Am Statistical Assoc* 2002, **97:**77-87.
3. Golub TR, Slonim DK, Tamayo P, Huard C, Gassenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES: **Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.** *Science* 1999, **286:**531-537.
4. Singh D, Febbo PG, Ross K, Jackson DG, Manola J, Ladd C, Tamayo P, Renshaw AA, D'Amico AV, Richie JP, Lander ES, Loda M, Kantoff PW, Golub TR, Sellers WR: **Gene expression correlations of clinical prostate cancer behavior.** *Cancer Cell* 2004, **1:**203-209.
5. van't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AA, Mao M, Peterse HL, van der Kooy K, Martoon MJ, Witteveen AT, Schreiber GJ, Kerkhoven RM, Roberts C, Linsley PS, Bernards R, Friend SH: **Gene expression profiling predicts clinical outcome of breast cancer.** *Nature* 2002, **419:**530-536.
6. West B, Blanchette C, Dressman H, Huang E, Ishida S, Spang R, Zuzan H, Olson JA Jr, Marks JR, Nevins JR: **Predicting the clinical status of human breast cancer by using gene expression profiles.** *Proc Natl Acad Sci USA* 2001, **98:**11462-11467.
7. Duda RO, Hart PE, Stork DG: *Pattern Classification* 2nd edition. A Wiley-Interscience Publication; 2000.
8. Friedman JH: **Flexible metric nearest neighbor classification.** In *Technical report* Dept. of Statistics, Stanford University; 1994.
9. Howland P, Park H: **Generalizing discriminant dnalysis using the generalized singular value decomposition.** *IEEE Trans. on Pattern Analysis and Machine Intelligence* 2004, **26:**995-1006.
10. Ye J, Li T, Xiong T, Janardan R: **Using uncorrelated discriminant analysis for tissue classification with gene expression data.** *IEEE/ACM Trans. on Computational Biology and Bioinformatics* 2004, **1:**181-190.
11. Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D: **Support vector machine classification and validation of cancer tissue samples using microarray expression data.** *Bioinformatics* 2000, **16:**906-914.
12. Jaakkola T, Diekhans M, Haussler D: **Using the Fisher kernel method to detect remote protein homologies.** In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology* Edited by: Menlo Park, CA. AAAI Press.
13. Zien A, Rätsch G, Mika S, Schölkopf B, Lemmen C, smola A, Lengauer T, Müller K: **Engineering support vector machine kernels that recognize translation initiation sites.** *Bioinformatics* 2000, **16:**799-807.
14. Cawley GC: **MATLAB support vector machine toolbox.** 2000 [http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox]. University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ
15. Pochet N, Smet FD, Suykens JAK, Moor BLRD: **Systematic benchmarking of microarray data classification: Assessing the role of non-linearity and dimensionality reduction.** *Bioinformatics* 2004, **20:**3185-3195.
16. Natsoulis G, Ghaoui LE, Lanckriet GRG, Tolley AM, Leroy F, Dunlea S, Eynon BP, Pearson CI, Tugendreich S, Jarnagin K: **Classification of a large microarray data set: Algorithm comparison and analysis of drug signatures.** *Genome Res* 2005, **15:**724-736.

17. **Bio-medical Data Analysis** [http://sdmc.lit.org.sg/GEDatasets/]
18. **Center for Applied Genomics and Technology** [http://mgm.duke.edu/genome/dna_micro/work/]
19. **Cancer Program Data Sets** [http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi]
20. **St. Jude Research** [http://www.stjuderesearch.org/data/]
21. Armstrong SA, Staunton JE, Silverman LB, Pieters R, den Boer ML, Minden MD, Sallan SE, Lander ES, Golub TR, Korsmeyer SJ: **MLL translocations specify a distinct gene expression profile that distigushes a unique leukemia.** *Nature Genetics* 2001, **30:**41-47.
22. Pomeroy SL, Tamayo P, Gaasenbeek M, Sturla LM, Angelo M, McLaughlin ME, Kim JYH, Goumnerova LC, Black PM, Lau C, Allen JC, Zagzag D, Olson JM, Curran T, Wetmore C, Biegel JA, Poggio T, Mukherjee S, Rifkin R, Califano A, Stolovitzky G, Louis DN, Mesirov JP, Lander ES, Golub TR: **Prediction of central nervous system embryonal tumor outcome based on gene expression.** *Letters to Nature Nature* 2002, **415:**436-442.
23. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ: **Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissue probed by oligonucleotide arrays.** *Proc Natl Acad Sci USA* 1999, **96:**6745-6750.
24. Gordon GJ, Jenson RV, Hsiao L-L, Gullans SR, Blumenstock JE, Ramaswamy S, Richards WG, Sugarbaker DJ, Bueno R: **Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelima.** *Cancer Research* 2002, **62:**4936-4967.
25. Shipp MA, Ross KN, Tamayo P, Weng AP, Kutok JL, Aguiar RCT, Gaasenbeek M, Angelo M, Reich M, Pinkus GS, Ray TS, Koval MA, Last KW, Norton A, Lister TA, Mesirov J, Neuberg DS, Lander ES, Aster JC, Golub TR: **Diffuse large B-cell lymphoma outcome prediction by gene expression profiling and supervised machine learning.** *Nature Medicine* 2002, **8:**68-74.
26. Petricoin EF, Ardekanl AM, Hitt BA, Levine PJ, Fusaro VA, Steinberg SM, Mills GB, Simone C, Fishman DA, Kohn EC, Liotta LA: **Use of proteomic patterns in serum to identify ovarian cancer.** *The Lancet* 2002, **359:**572-577.
27. Xiong H, Swamy MNS, Ahmad MO: **Optimizing the data-dependent kernel in the empirical feature space.** *IEEE Trans. on Neural Networks* 2005, **16:**460-474.
28. Ruiz A, Lopez-de Teruel PE: **Nonlinear kernel-based statistical pattern analysis.** *IEEE Trans. on Neural Networks* 2001, **12:**16-32.
29. Baudat G, Anouar F: **Generalized discriminant analysis using a kernel approach.** *Neural Computation* 2000, **12:**2385-2404.
30. Cristianini N, Shawe-Taylor J: *An Introduction to Support Vector Machines* Cambridge Univ. Press, Cambridge, UK; 2000.
31. Amari S, Wu S: **Improving support vector machine classifiers by modifying kernel functions.** *Neural Networks* 1999, **12:**783-789.
32. Müller K-R, Mika S, Rätsch G, Tsuda K, Scholkopf B: **An introduction to kernel-based learning algorithms.** *IEEE Trans. on Neural Networks* 2001, **12:**181-201.
33. Pekalska E, Paclik P, Duin Robert PW: **A generalized kernel approach to dissimilarity-based classification.** *Journal of Machine Learning Research* 2001, **2:**175-211.
34. Roth V, Steinhage V: **Nonlinear discriminant analysis using kernel functions.** In *Advance in Neural Information Processing Systems 12* Edited by: Solla SA, Leen TK, Muller K-R. Cambridge, MA:MIT Press; 2000:568-574.
35. Scholkopf B, Mika S, Burges CJC, Knirsch P, Muller K-R, Ratsch G, Smola AJ: **Input space versus feature space in kernel-based methods.** *IEEE Trans. on Neural Networks* 1999, **10:**1000-1017.
36. Graf ABA, Smola AJ, Borer S: **Classification in a normalized feature space using support vector machines.** *IEEE Trans. on Neural Networks* 2003, **14:**597-605.