Vol. 47 No. SIG 10(CVIM 15) IPSJ Transactions on Computer Vision and Image Media July 2006

Regular Paper

Learning Kernels from Distance Constraints

Tsuyoshi Kato,[†,††] Wataru Fujibuchi[††] and Kiyoshi Asai[†,††]

Recently there has been a surge of interest in kernel methods such as support vector machine due to their flexibility and high performance. It is important how to define a kernel for kernel methods. Most of kernels are defined by inner-product of feature vectors in some vector space. In this paper we discuss an approach which constructs a kernel matrix from distances between examples instead of feature vectors. Namely, the input data of our algorithm are the distances among examples, not feature vectors. Dissimilar to most of conventional kernels where kernel functions are explicitly given and the kernel matrices are determined by simple calculations, our algorithm rather builds a kernel matrix by maximizing its entropy subject to distance constraints. The maximization problem is convex, so we can always attain to the optimal solution. Experiments using artificial data show the benefits of our algorithm. In addition, we apply this method to analysis of heterogeneous microarray gene expression data, and report the experimental results.

1. Introduction

Kernel methods such as support vector machine (SVM) [19] have proven to be extremely powerful in many areas of machine learning. Such a method can be applied when the kernel between examples of interest is defined. Therefore, lots of researchers have devised novel types of kernels for vectorial data as well as highly structured non-vectorial data such as biological sequences [12),13),27)], chemical compounds [8)], natural language [23)] and speech processing [21)].

Most of kernels for structured data are constructed based on extraction of feature vectors. For example, the marginalized kernel devised by Tsuda, et al. [27)] counts each symbol generated from each state. String kernels basically count k-mers [14)]. Some of kernels are instances of a class of the convolution kernel which sums up all inner products of components composing the object. Those kernel developments are procreated under consideration of constructing kernels from feature vectors, which is the presumably easiest way for preserving the necessary property such that kernels must be positive definite. However they tend to be extremely high-dimensional and can not avoid including many irrelevant features. Those feature-based kernels are useful for the first development for the fields which have not been investigated well. However, if sufficient prior knowledge is readily available, important features only should be incorporated into kernels and unimportant information should be excluded. Elimination of irrelevant features from such a kernel is unfavorably difficult, since all features are incorporated systematically and developers can not explore which features are critical via, for example, feature selection.

We also encounter the cases that complete feature vectors are not procurable for all examples. For instance, such a case can be seen in analysis of microarray data [15)]. The microarray technique measures gene expression level under a variety of conditions simultaneously. Microarray data are usually given as a matrix: the row and column specifies to which gene and which cell (or experiment) a particular piece of data corresponds (see Fig. 4). Microarray analyses have been applied in lots of studies over a wide variety of biological fields including cancer classification [6)] and identification of the unknown effects of a specific therapy [17)]. A crucial issue is that microarray data include many missing data. Nonetheless, most of the current studies about microarray data are based on multivariate statistical analyses or kernel methods, which have been developed without the assumption of existence of missing data. Alternatively, instead of feature vectors, the correlations between cells are oftentimes used for representing the relationship among cells, in which only common visible (non-missing) expression values in two cells are used for computation of those correlations. Namely, each correlation is computed from a different gene set. Hence those correlations are not positive definite. For that reason, the range of applica-

† Graduate School of Frontier Sciences, The University of Tokyo
†† AIST Computational Biology Research Center

tions for those correlations is somewhat limited; most of conventional multivariate analyses and kernel methods are not capable to work on such correlations.

In this paper we present an approach for kernel construction. Our algorithm uses the distances between examples instead of building feature vectors. Our approach is based on the locally constrained diffusion kernel (LCDK) [26] which provides the similarities among nodes on a graph. LCDK assumes an embedding of each node into a kernel Hilbert space $\mathcal{F}$, and is constructed such that the entropy of the kernel matrix is maximized as long as the distance between two images in $\mathcal{F}$ whose nodes are connected on the graph is less than a constant threshold. Exploiting that technique, we construct a kernel matrix from the upper-bounded distance between each pair of examples. Furthermore, our algorithm has the capability for meeting the demands that we wish some particular examples to be apart from particular examples. It is useful to incorporate the prior knowledge about relationships among examples.

There are similar attempts which embed examples into some vector space using distances. Presumably the most popular attempts are the multi-dimensional scaling [24] and the locally linear embedding [18]. These two methods transfer the example data into a low dimensional Euclidean space (typically two-dimensional). Weinberger et al. [31] have presented a method for learning a kernel matrix so that examples are mapped into a (typically low-dimensional) kernel Hilbert space instead of a low dimensional Euclidean space. All of them have been developed with a common purpose which is visualization. Therefore, all the examples are projected into a common low-dimensional space even if they belong to different classes. Mostly those projections are preferable for visualization but not good for classification.

Besides our algorithm and the above mentioned methods, there are other methods based on distances. For classification, the k-nearest neighbor classifier is presumably the most well-known method based on distances. For clustering, the single-linkage method is widely known. The benefit of our algorithm is that kernel matrices are obtained. Our algorithm thereby provides a device to combine any kernel method with distance information. Therefore the applications are not limited to classification or clustering. For example, the kernel matrix obtained from our algorithm can selectively be integrated with other types of data using SDP-SVM [11] or support kernel machine [2]. Supervised network inference methods recently developed [9),29),32)] require kernel matrices among nodes. Our algorithm enables the network inference methods to perform for data consisting of distances by conversion of them into a kernel matrix. Thus our algorithm is a powerful way to feed distances to kernel methods.

This paper is organized as follows: The next section recalls some basic preliminaries for kernel methods. In section 3 we describe our algorithm and carry out simulations using artificial data in section 4. In section 5, we present a new challenging problem: analyzing mixture of microarray data provided from different laboratories, and apply our algorithm to that problem. The last section concludes our paper with discussion.

2. Preliminaries

Kernel methods work by embedding the data into a kernel Hilbert space $\mathcal{F}$. The embedding is performed implicitly, by defining the inner product between each pair of examples rather than by giving their actual values of vectors [19]. Given a set of input examples $\boldsymbol{x}_i \in \mathcal{X}, \quad (i = 1, \cdots, N)$ and an embedding space $\mathcal{F}$, we consider a mapping function $\Phi(\cdot) : \mathcal{X} \to \mathcal{F}$. Given two examples $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$, the function $k(\cdot, \cdot)$ giving the inner product between their images, say $\Phi(\boldsymbol{x}_i)$ and $\Phi(\boldsymbol{x}_j)$ in the space $\mathcal{F}$, is called a kernel function. A kernel matrix $K \in \Re^{N \times N}$ is a symmetric positive definite matrix of which elements are the values of the kernel function for $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N \in \mathcal{X}$. Conversely, every symmetric positive definite matrix is a kernel matrix.

In this paper we assume a transductive setting [7] where we are given both labeled examples and test examples in advance in a learning stage. In a transductive setting, we do not have to know the kernel function $k(\cdot, \cdot)$ nor the implicitly defined mapping function Φ, nor the actual values of the images $\Phi(\boldsymbol{x}) \in \mathcal{F}$, if we have a kernel matrix. For example, the score of j-th example can be written as

$$f_j = \sum_{i=1}^{\ell} \alpha_i y_i K_{ij} + b,$$

where the first $\ell (< N)$ examples are labeled by $y_i \in \{\pm 1\}$. α_i and b are the parameters determined by the SVM learning algorithm [19]. Notice that the scores are expressed by only

kernel values and do not explicitly include any actual vectors in $\mathcal{F}$. For a given learning task, an important point is that of choosing a kernel, which corresponds to choosing a kernel matrix.

3. Method

Roughly speaking, we are now considering the following requirements given for construction of a kernel matrix:

- The distances for a particular set of example pairs should be small.
- The distances for a particular set of example pairs should be large.

Speaking again, kernels must be the inner product on a kernel Hilbert space. We wish to obtain a kernel matrix in which the points in the kernel Hilbert space satisfy the above requirements. Denoting the examples by $\boldsymbol{x}_i \in \mathcal{X} (i = 1, \cdots, N)$ and the mapping function by $\Phi : \mathcal{X} \mapsto \mathcal{F}$, we formulate these conditions using M^U constraints for the upper-bound of squared distances d_k^{U} and M^L constraints for the lower-bound of squared distances d_k^{L} in $\mathcal{F}$ as follows:

$$\|\Phi(\boldsymbol{x}_{i_k}) - \Phi(\boldsymbol{x}_{j_k})\|^2 \leq d_k^{\mathrm{U}}, \quad \text{for } k = 1, \cdots, M^U, \tag{1}$$

$$\| \Phi(\boldsymbol{x}_{i_k}) - \Phi(\boldsymbol{x}_{j_k})\|^2 \geq d_k^{\mathrm{L}}, \quad \text{for } k = 1, \cdots, M^L. \tag{2}$$

As described in the previous section, kernel methods work on a kernel matrix with elements $K_{ij} = \Phi(\boldsymbol{x}_{i_k})^\top \Phi(\boldsymbol{x}_{j_k})$. Under the constraints in Eqs. (1),(2), we wish to construct a kernel matrix with least irrelevant information. Such a kernel matrix $K \in \Re^{N \times N}$ is obtained by maximization of the Von Neumann entropy [16] defined by

$$-\mathrm{tr}\,(K \log K), \qquad \mathrm{tr}\, K = 1, \tag{3}$$

where log takes the matrix logarithm. Using the elements of a kernel matrix, the squared Euclidean distance between $\Phi(\boldsymbol{x}_i)$ and $\Phi(\boldsymbol{x}_j)$ is described as

$$\|\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{x}_j)\|^2 = K_{ii} + K_{jj} - K_{ij} - K_{ji}. \tag{4}$$

For the sake of simple notation, we define U_k $(k = 1, \cdots, M^U)$ and L_k $(k = 1, \cdots, M^L)$ by

$$U_k = E_{i_k i_k} + E_{j_k j_k} - E_{i_k j_k} - E_{j_k i_k} - d_k^{\mathrm{U}} I, \tag{5}$$

$$L_k = -E_{i_k i_k} - E_{j_k j_k} + E_{i_k j_k} + E_{j_k i_k} + d_k^{\mathrm{L}} I. \tag{6}$$

where E_{ij} denotes a matrix in which (i, j) element is one and all the others are zero. Then the upper-bound and lower-bound constraints can be rewritten as

$$\mathrm{tr}(U_k K) \leq 0 \tag{7}$$

and

$$\mathrm{tr}(L_k K) \leq 0, \tag{8}$$

respectively.

For keeping the optimization problem feasible, we relax the problem like soft-margin support vector machine as follows:

$$\begin{aligned} \min \quad & \mathrm{tr}\,(K \log K) + C^{\mathrm{U}} \|\boldsymbol{\xi}^{\mathrm{U}}\|_1 + C^{\mathrm{L}} \|\boldsymbol{\xi}^{\mathrm{L}}\|_1, \\ \text{subj. to} \quad & \mathrm{tr}\, K = 1, \\ & \mathrm{tr}(U_k K) \leq \xi_k^{\mathrm{U}}, \qquad k = 1, \cdots, M^U, \\ & \mathrm{tr}(L_k K) \leq \xi_k^{\mathrm{L}}, \qquad k = 1, \cdots, M^L, \\ & \boldsymbol{\xi}^{\mathrm{U}} \geq \boldsymbol{0}, \quad \boldsymbol{\xi}^{\mathrm{L}} \geq \boldsymbol{0}, \\ \text{w.r.t.} \quad & K, \boldsymbol{\xi}^{\mathrm{U}} = [\xi_1^{\mathrm{U}}, \cdots, \xi_{M^U}^{\mathrm{U}}]^\top, \\ & \boldsymbol{\xi}^{\mathrm{L}} = [\xi_1^{\mathrm{L}}, \cdots, \xi_{M^L}^{\mathrm{L}}]^\top,) \end{aligned} \tag{9}$$

where C^{U} and C^{L} are constant. Since the entropy function is convex, the optimization function is convex. Inasmuch as all the constraints are linear, the feasible region is a convex set. Consequently, the optimization problem does not have any local minima, and we can always attain to the optimal solution.

When giving the bounds of squared distances $d_k^{\mathrm{U}}, d_k^{\mathrm{L}}$, we should take account of the other constraint: the trace of K must be one. Due to that constraint, the average of squared norms $\|\Phi(\xi)\|^2$ becomes $1/N$. Hence a suitable heuristic is to normalize the bound of each squared distance by dividing them by N.

Learning algorithm

A steepest descent algorithm for a general problem with this type of the objective function and linear constraints has already been provided [26]. We employ that algorithm with simple modification for solving our problem. Here we describe it briefly as the following. The learning algorithm solves the dual problem instead of the primal problem given in Eq. (9). The dual problem is described by (refer the appendix for the derivation):

$$\begin{aligned} \max \quad & J(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\log \mathrm{tr}\,(\exp(-\mathcal{U}\boldsymbol{\alpha} - \mathcal{L}\boldsymbol{\beta})). \\ \text{subj. to} \quad & \boldsymbol{0} \leq \boldsymbol{\alpha} \leq C^{\mathrm{U}} \boldsymbol{1}, \quad \boldsymbol{0} \leq \boldsymbol{\beta} \leq C^{\mathrm{L}} \boldsymbol{1} \\ \text{w.r.t.} \quad & \boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_{M^U}]^\top, \\ & \boldsymbol{\beta} = [\beta_1, \cdots, \beta_{M^L}]^\top, \end{aligned} \tag{10}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are dual variable vectors, and the operators, $\mathcal{U}$ and $\mathcal{L}$, perform $\mathcal{U}\boldsymbol{\alpha} = \sum_{k=1}^{M^U} \alpha_k U_k$ and $\mathcal{L}\boldsymbol{\beta} = \sum_{k=1}^{M^L} \beta_k L_k$. $\boldsymbol{1}$ denotes a column vector in which all elements are one. For optimization, the steepest descent method

is used. The derivatives of the dual function is given by

$$\frac{\partial J}{\partial \alpha_k} = \frac{\mathrm{tr}\left(U_k \exp\left(-\mathcal{U}\boldsymbol{\alpha}-\mathcal{L}\boldsymbol{\beta}\right)\right)}{\mathrm{tr}\left(\exp\left(-\mathcal{U}\boldsymbol{\alpha}-\mathcal{L}\boldsymbol{\beta}\right)\right)}, \\ \frac{\partial J}{\partial \beta_k} = \frac{\mathrm{tr}\left(L_k \exp\left(-\mathcal{U}\boldsymbol{\alpha}-\mathcal{L}\boldsymbol{\beta}\right)\right)}{\mathrm{tr}\left(\exp\left(-\mathcal{U}\boldsymbol{\alpha}-\mathcal{L}\boldsymbol{\beta}\right)\right)}. \quad (11)$$

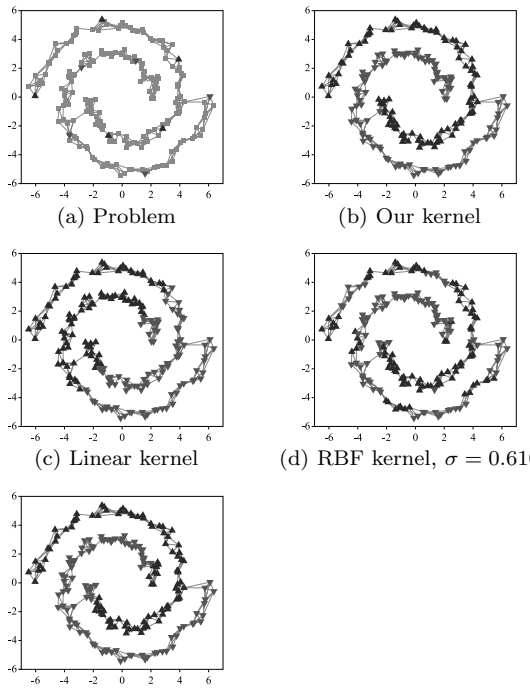
When the values of some dual variables violate the constraints in (10), they are forced to be back into the feasible region. In our simulation, we put zero to the initial values of the dual variables. However, since the optimization problem is convex, the optimal solution can always be attained from any initial values. Once we get the dual optimal, we can recover the primal optimal solution as follows:

$$K = \frac{\exp\left(-\mathcal{U}\boldsymbol{\alpha}-\mathcal{L}\boldsymbol{\beta}\right)}{\mathrm{tr}\left(\exp\left(-\mathcal{U}\boldsymbol{\alpha}-\mathcal{L}\boldsymbol{\beta}\right)\right)}. \quad (12)$$

Let us discuss the time complexity of the learning algorithm. U_k and L_k are sparse. Denote the number of non-zeros in each matrix by N_{nz}. If we use a special data structure for sparse matrices (e.g. Harwell-Boeing sparse matrix storage format [5]), addition $S = -\mathcal{U}\boldsymbol{\alpha} - \mathcal{L}\boldsymbol{\beta} \in \Re^{N\times N}$ takes $O(N_{\mathrm{nz}}(M^{\mathrm{U}}+M^{\mathrm{L}}))$. Computation of matrix exponential of S takes $O(N^3)$. The trace of the product between U_k (or L_k) and $S' = \exp(S)$, say $\mathrm{tr}(U_k S')$ (or $\mathrm{tr}(L_k S')$), requires the computational time $O(N_{\mathrm{nz}})$, because that trace can be rewritten as the inner product of vectors with N_{nz} elements. Computing the trace of S' also takes $O(N)$. The number of non-zeros of U_k (or L_k) is $N+2$. Hence, after obtaining S', we need $O(N)$ for computation of the gradient Eq. (11) for each element in a dual vector. Recovery of the kernel matrix takes $O(N(M^{\mathrm{U}}+M^{\mathrm{L}})+N^3+N)$. If we assume $M^{\mathrm{U}}+M^{\mathrm{L}} < N^2$, the total computational time is therefore $O(T_{\mathrm{iter}}N^3)$ where the number of iterations is T_{iter}.

4. Simulations

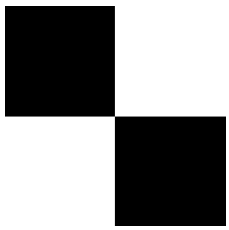
For demonstrating the performance of our kernel, simulations are performed on an artificial dataset including $N = 200$ points in two-dimensional space plotted in **Fig. 1** (a). The dataset is comprised of two whorls each of which has 100 points: The first 100 points are in one whorl, and the last 100 points are in the other whorl. Since two whorls are entangled, they cannot be divided linearly. For building the constraints for our algorithm, we extract point pairs $(i_k, j_k)(k = 1, \cdots, M^U)$ such that x_{i_k} is one of five nearest neighbors of x_{j_k} computed using squared Euclidean distances $d^{\mathrm{orig}}_{i_k j_k}$. As a result, we obtained 565 pairs which are connected in Fig. 1 (a). From them, we formed $M^U = 565$ upper-bound constraints such as $d_k = 0.05 d^{\mathrm{orig}}_{i_k j_k}/N$. In this simulation, we built no lower-bound constraints. We set $C^{\mathrm{U}} = 100$. The resulting kernel matrix is normalized.

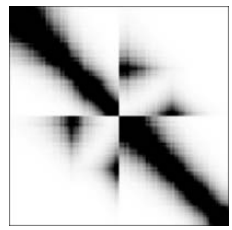


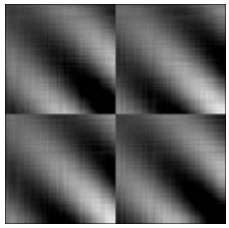
(a) Problem (b) Our kernel (c) Linear kernel (d) RBF kernel, $\sigma = 0.610$ (e) RBF kernel, $\sigma = 5.9$

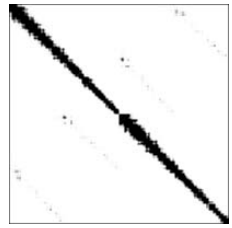
Fig. 1 Classification results. The plot (a) shows a classification problem with $N = 200$ points in two-dimensional space. Therein, five points and other five points are given positive and negative class labels indicated by downward-pointing and upward-pointing triangles, respectively. We gave the upper-bound constraints to the point pairs connected by solid line in (a). Using these points as training examples, SVM with our kernel predict class labels. The prediction results are plotted in (b). The results of SVM with linear kernel and RBF kernels with $\sigma = 0.610, 5.9$ are also shown in (c),(d) and (e), respectively.

For comparison with conventional kernels, we use linear kernel and RBF kernels. For illustrating the classification accuracy, we use a standard SVM at regularization parameter being $C = 100$ for all kernels in this simulation. We give class labels to ten points. As shown in Fig. 1 (a), class labels are indicated by upward-pointing and downward-pointing triangles; Square points denote unlabeled points.


(a) Ideal


(b) Our kernel


(c) Linear kernel


(d) RBF kernel, $\sigma = 0.610$

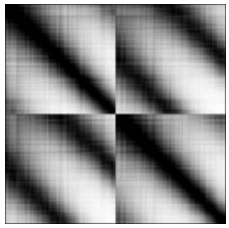

(e) RBF kernel, $\sigma = 5.9$

Fig. 2 Kernel matrices. The matrix in (a) indicates true class labels of all $N = 200$ points by block diagonals and is called an ideal matrix. (b),(c),(d) and (e) depict 200×200 matrices of our kernel, linear kernel, RBF kernel with $\sigma = 0.610$ and RBF kernel with $\sigma = 5.9$, respecitvely.

Usage of linear kernel is equivalent to classical (non-kernelized) linear multivariate analysis. RBF kernel is well-known as a kernel with the capability of handling non-linear analysis. RBF kernel has a parameter, σ. For determining the value σ, we find five nearest neighbors for each point, and take the average of the Euclidean distances. We set that average to σ. In this setting, $\sigma = 0.610$. We also report the result of RBF kernel with σ which achieves the maximum accuracy for classifying each point into the two whorls. For this purpose, we perform a search over the values: $\sigma = 0.1, 0.2, \cdots, 8.0$. The value getting the maximum accuracy was $\sigma = 5.9$.

In that setting, we obtained four kernel matrices depicted in **Fig. 2** (b)(c)(d)(e). As expected, linear kernel could not capture the nonlinear structure with two whorled clusters, and the classification result are poor as shown in Fig. 1 (c). Meanwhile, our kernel represented two clusters successfully, and almost perfectly classified each point into the two whorls (see Fig. 1 (b)). RBF kernel with $\sigma = 0.610$ does not yield good classification performance (see Fig. 1 (d)). Every column in the kernel matrix (Fig. 2 (d)) has few elements being large enough. So many points are almost orthogonal to labeled points. Empirically it has been observed that SVMs do not perform well in this situation [20]. Even if the best σ is chosen, the classification performance of the RBF kernel is inferior to our kernel (see Fig. 1 (e)). A kernel value of RBF kernel is produced from the Euclidean distance of only one pair. Our kernel can benefit from the relationships among the whole dataset, and is especially of great advantage for the dataset with clusters, as illustrated by these experimental results.

Addition of Lower-Bound Constraints

To gain some insight into the basic properties of the lower-bound constraints, we conducted another numerical experiment. Use of lower-bound constraints is effective when we have prior information that some of pairs are under different rules. If the distance between each of such pairs is large in a kernel matrix, the subsequent prediction works well. Lower-bound constraints can force each of those pairs to be distant.

We now illustrate the results of a regression problem. The problem of regression consists in predicting a real-valued label of each test point using training points with labels $z_i \in \Re$. We generated 400 points in $\Re^2$ as shown in **Fig. 3** (a). While the input data in the previous simulation are divided from two clusters, the input data in this numerical experiment do not have such a cluster structure. We switched the rule for generating labels according to the distance from the origin in the two-dimensional space (i.e. the norm of a vector in $\Re^2$). If the norm is larger than 4.5, the true label is given by

$$z_i = +(\theta([\boldsymbol{x}_i]_2, [\boldsymbol{x}_i]_1) + \pi)/2\pi,$$

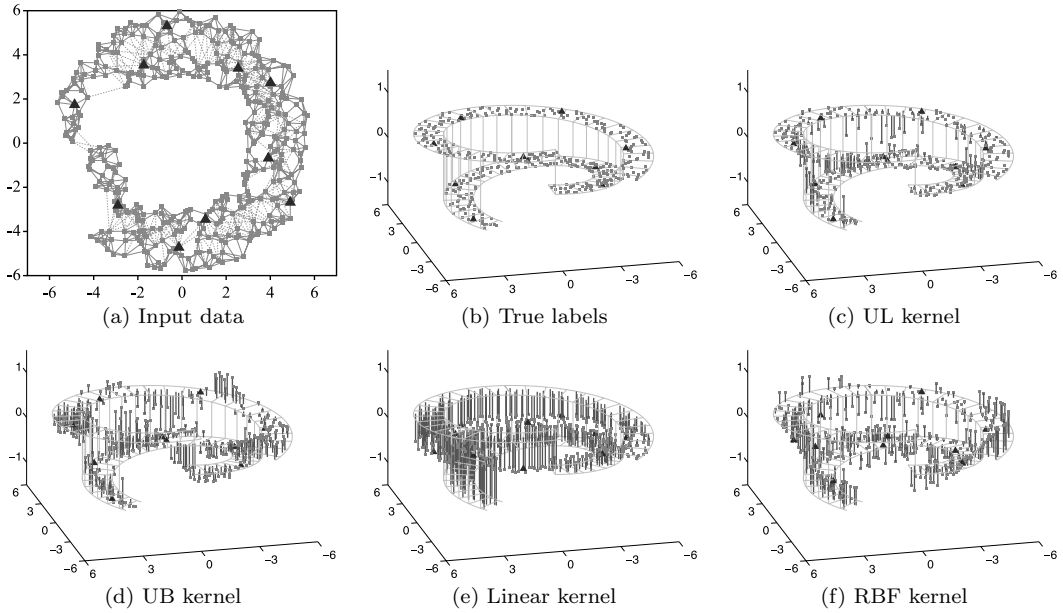
otherwise

$$z_i = -(\theta([\boldsymbol{x}_i]_2, [\boldsymbol{x}_i]_1) + \pi)/2\pi,$$

where $\theta : \Re \times \Re \mapsto [-\pi, \pi]$ is the arc tangent function computed by

$$\theta(y, x) = \mathrm{atan2}(y, x)$$

in ANSI C library. The resultant labels are shown in Fig. 3 (b). We have chosen ten points and give them the labels according to the above rule. For regression from a kernel matrix, we



(a) Input data (b) True labels (c) UL kernel

(d) UB kernel (e) Linear kernel (f) RBF kernel

Fig. 3 The effects of addition of the lower-bound constraints. The regression problem is to perform prediction of real-valued labels from the input data in (a). In (a) solid and dotted lines connect the pairs given upper-bound and lower-bound constraints, respectively. The true real-valued labels are shown in (b), where the vertical axis denotes the value of labels. Triangles denote labeled points. We tested various kernels: UL kernel, UB kernel, linear kernel and RBF kernel, shown in (c), (d), (e), and (f), respectively. The RMSEs are 0.163, 0.300, 0.556, and 0.229, respectively. It is demonstrated that addition of lower-bound constraints improves the prediction performance significantly.

employ the kernel ridge regression [19]. Kernel ridge regression has a regularization parameter which has to be adjusted manually. The results reported here are of the regularization parameter yielding the minimum root mean square error (RMSE).

First let us see the results of the case where only the upper-bound constraints are used. The upper-bound constraints are given in the same fashion as the previous simulation. The predicted values are shown in Fig. 3 (d). We also tested the linear kernel and the RBF kernel and the predictions are depicted in Fig. 3 (e),(f), where the parameter of RBF kernel, σ, is chosen by exhaustive search such that the minimum RMSE is obtained. Ridge regression using linear kernel yields a linear function, so it can not approximate the non-linear structure well (RMSE of 0.556). If our kernel with only upper-bound constraints (UB kernel) or RBF kernel is used, the prediction errors are still large. The RMSEs of the UB kernel and RBF kernel are 0.300 and 0.229, respectively. Those results implied that it is difficult to solve this regression problem from only the feature vectors in $\Re^2$ whichever kernel is used.

We next assume that additional information is available. Suppose we know the labels of some pairs of points are generated according to different rules even if the actual labels are unknown. In this experiment, we postulate that such information is available for the pairs of which both points with index divisible by four. We exploit the additional information by adding the lower-bound constraints for the following pairs (i_k, j_k): $i_k (\mathrm{mod}\ 4) = 0$, $j_k (\mathrm{mod}\ 4) = 0$, $(\|\boldsymbol{x}_{i_k}\| - 4.5)(\|\boldsymbol{x}_{j_k}\| - 4.5) < 0$ and $\|\boldsymbol{x}_{i_k} - \boldsymbol{x}_{j_k}\|^2 < 2$. We put $d_k^{\mathrm{L}} = 1/N$ for all these pairs. The upper-bound constraints are same as UB kernel. We set $C^{\mathrm{L}} = 1000$ and $C^{\mathrm{U}} = 100$. We refer the kernel from upper-bound and lower-bound constraints to UL kernel. There exists no embedding satisfying all the distance constraints with $\boldsymbol{\xi}^{\mathrm{U}} = 0$ and $\boldsymbol{\xi}^{\mathrm{L}} = 0$ and the triangle inequality simultaneously. Nonetheless, our algorithm is capable of working well due to the soft margin technique. In Fig. 3 (a), the pairs given the upper-

bound and the lower-bound constraints are connected by solid lines and dotted lines, respectively. The prediction results are depicted in Fig. 3 (c). UL kernel achieves RMSE of 0.163. Addition of the upper-bound constraints has led to significant improvement.

Actually UL kernel can be applied for cases where a number of heterogeneous data are blindly mixed and are difficult to discriminate each other. For example, in the cancer prognosis prediction [28], patients should be categorized into cancer subtypes such as kidney or lung cancers before the regression. However, it is often impossible to perfectly classify patients into distinct groups due to the complex cancer diagnoses considering many aspects of patients such as malignancy or metastasis levels of the cancer. In our algorithm, we can exploit the prior knowledge that some patients are distinct from other patients in prediction. Our algorithm using the lower-bound constraints may effectively incorporate the prior knowledge into a kernel matrix and simultaneously predict distinct data.

5. Application to Microarray Data

It is relatively easy to analyze microarray data provided from one laboratory since one might apply some automatic normalization method to them to remove biases. Most of studies have tackled such data so far. However, to our knowledge, no one has been analyzing a mixture of microarray data from multiple laboratories using multivariate statistical analyses or kernel methods. The reason might be that each of microarray data registered in public database [3] has a different type of systematic biases. Furthermore, some are pre-normalized and others are not, but we often meet microarray data with no document describing whether it is pre-normalized or not. In addition, microarray data have missing values as described in Section 1. There are various methods [4),25),30)] for imputation of missing values in microarray data. However all of those have been developed without taking account of heterogeneous microarray data. Those facts considerably make analysis difficult. Hereinafter we refer a mixture of microarray data from different laboratories to *heterogeneous microarray data*, whereas we refer microarray data from single laboratory to *homogeneous microarray data*.

To alleviate such heterogeneous biases and missing problem, Spearman rank correlation (SRC) [10] is often used for representing relationships among cells. The SRC between two cells, say x_i and x_j, is defined by

$$\mathrm{SRC}(x_i, x_j) = 1 - \frac{6}{d(d^2-1)} \sum_{h=1}^{d} (r_h(x_i) - r_h(x_j))^2 \quad (13)$$

where $r_h(x)$ is the rank of h-th gene in x. When data include missing values, the SRC is computed from common visible data between cells. The SRC obtained by that computation is not positive definite. Hence, it can not directly be fed to kernel methods, but our algorithm can transform the SRC to a kernel matrix.

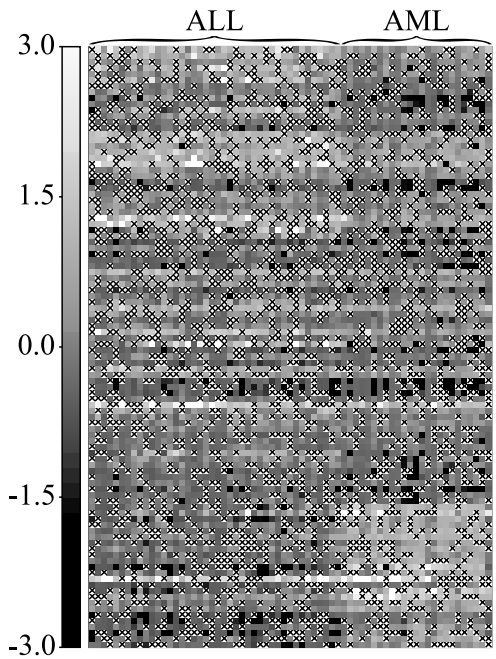
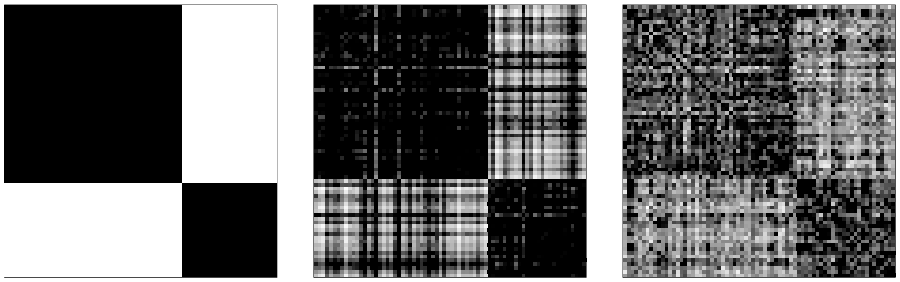


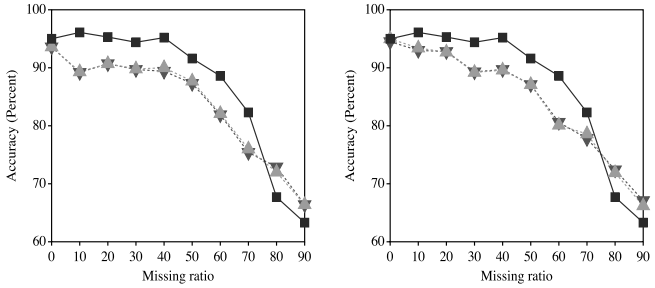
Fig. 4 Heterogeneous microarray data with 30% missing values. Each column is normalized by z-score. 'x' denotes a missing element.

We use the microarray dataset containing close but two different cell types called AML and ALL [6]. To collect this dataset, bone marrow or blood samples were taken from 72 patients including 47 with acute myeloid leukemia (AML) and 25 with acute lymphoblastic leukemia (ALL). We extract 100 genes from the dataset by the software RankGene [22]. To simulate the dataset with heterogeneous biases, we randomly chose cells and log-transformed them. To create missing values, we removed various percentages of the data (see **Fig. 4**). Then we compute the SRC, find ten nearest neighbors for each cell, and give the upper-



(a) Ideal (b) Our kernel (c) Linear

Fig. 5 Kernel matrices for heterogeneous microarray data with 30% missing values.



(a) Heterogeneous data (b) Homogeneous data

Fig. 6 Classification results on microarray data.

bounded distance $d_k^{\mathrm{U}} = (1 - \mathrm{SRC}(x_{i_k}, x_{j_k}))/2N$ to those pairs. No lower-bound constraints were given in this simulation. Independently we gave class labels (AML or ALL) to ten cells which were randomly chosen and classified 62 unlabeled cells via SVM with the kernel matrix generated by our algorithm. We repeated this procedure 20 times and evaluate the performance by average accuracy.

We also tested SVM with linear kernel and RBF kernel. Since linear kernel requires complete feature vectors without any missing elements, missing values are imputed by the average of the corresponding rows. We tried various values of the regularization parameter C for SVM and the various parameters of RBF kernel, and report the best accuracies.

Figure 5 depicts the kernel matrices built by our algorithm and the linear kernel computed from heterogeneous data with 30% missing values. Our kernel matrix clearly separated the two classes, whereas the linear kernel matrix seems to include much of irrelevant information.

Figure 6 (a) compares classification performance of the three kernels. Our kernel achieved highly accurate classification until 40% missing ratio without decreasing its accuracy. Accuracies of both methods converged to about 60% accuracy which is roughly equal to the ratio of the number of cells between ALL and AML. If the data are homogeneous (i.e. no data are log-transformed.), linear and RBF kernel work to a certain degree, but the accuracies are monotonically decreased (see Fig. 6 (b)). On the other hand, our kernel using SRC got the same accuracies as the heterogeneous data, since SRC is invariant to log-transformation prosperously.

6. Conclusion and Discussion

In this paper we discuss a methodology for building a kernel matrix from distances among examples instead of feature vectors. A notable contribution might rather be to have presented a new problem: Analysis of microarray data taken from various systematic biases due to different experimental methods. So far the SRC between non-missing data has been a common measure for representing relationship, but it suffers from a drawback: it is not directly applicable to a class of promising data analyzers, kernel methods. Our simulations suggest that our algorithm can be an effective bridge between kernel methods and non-positive definite similarities.

RBF kernel has a common property to our kernel. The RBF kernel is also computed us-

ing distances among examples instead of feature vectors. However RBF kernel suffers from two shortcomings:
(i) Distances of all pairs are required.
(ii) The negative of the distance function must be conditionally positive definite [19].
Especially, the latter drawback is rather obstructive for incorporation of prior knowledge. If the second state (ii) is violated, the resulting kernel is no longer positive definite, which breaks down whole the theory about kernel machines and yields local minima in SVM learning. Designing a conditionally positive definite function is not an easy task. Using the fact that the negative squared Euclidean distance function is conditionally positive definite, one might extract feature vectors and compute their Euclidean distances. However, it is impossible when features include missing values as mentioned in Section 1.

We can also consider another practical situation. Suppose one has a software yielding a measure of relationship between examples. For example, in order to represent the relationships between biological sequences by alignment scores, one might use alignment software such as Blast [1]. However, such a score is not conditionally positive definite generally. Meantime, our kernel tolerates any dissimilarity measures as distances even if the measures violate the triangle inequality.

Analysis of heterogeneous microarray data is a suitable application for our kernel as shown in this paper. We also expect that there are lots of other cases applicable to our kernel. Our future work is to investigate the performance of our kernel on many other problems.

Acknowledgments We wish to thank K. Tsuda and T. Kin for fruitful discussion. Suggestions by anonymous referees are greatly appreciated.

References

1) Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J.: Basic local alignment search tool, *J. Mol. Biol.*, Vol.215, No.3, pp.403–410 (1990).
2) Bach, F.R., Lanckriet, G.R.G. and Jordan, M.I.: Multiple kernel learning, conic duality, and the SMO algorithm, *Proc. 21st International Conference on Machine Learning* (2004).
3) Barrett, T., Suzek, T.O., Troup, D.B., Wilhite, S.E., Ngau, W.C., Ledoux, P., Rudnev, D., Lash, A. E., Fujibuchi, W. and Edgar, R.: NCBI GEO: Mining millions of expression profiles—database and tools, *Nucleic Acids Res.*, Vol.1, No.33, pp.D562–566 (2005).
4) Bo, T.H., Dysvik, B. and Jonassen, I.: LSimpute: Accurate estimation of missing values in microarray data with least squares methods, *Nucleic Acids Res.*, Vol.32, No.3, p.e34 (2004).
5) Duff, I.S., Grimes, R.G. and Lewis, J.G.: Sparse matrix test problems, *ACM Trans. Math. Software*, Vol.15, pp.1–14 (1989).
6) Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Downing, M.L.L. J.R., Caligiuri, M.A., Bloomfield, C.D. and Lander, E.S.: Classification of cancer: Class discovery and class prediction by gene expression monitoring, *Science*, Vol.286, No.5439, pp.531–537 (1999).
7) Joachims, T.: Transductive inference for text classification using support vector machines, *Proc. 16th International Conference on Machine Learning*, San Francisco, CA, Morgan Kaufmann, pp.200–209 (1999).
8) Kashima, H., Tsuda, K. and Inokuchi, A.: Marginalized Kernels Between Labeled Graphs, *Proc. 20th International Conference on Machine Learning* (*ICML2003*), Washington, DC USA (2003).
9) Kato, T., Tsuda, K. and Asai, K.: Selective integration of multiple biological data for supervised network inference, *Bioinformatics*, Vol.21, pp.2488–2495 (2005).
10) Kuo, W.P., Jenssen, T.K., Butte, A.J., Ohno-Machado, L. and Kohane, I.S.: Analysis of matched mRNA measurements from two different microarray technologies, *Bioinformatics*, Vol.18, No.3, pp.405–412 (2002).
11) Lanckriet, G.R.G., Bie, T.D., Cristianini, N., Jordan, M. and Noble, W.: A Statistical Framework for Genomic Data Fusion, *Bioinformatics*, Vol.20, pp.2626–2635 (2004).
12) Leslie, C., Eskin, E., Cohen, A., Weston, J. and Noble, W.: Mismatch String Kernels for Discriminative Protein Classification, *Bioinformatics*, Vol.4, pp.467–476 (2004).
13) Liao, L.I. and Noble, W.S.: Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships, *J. Comput. Biol.*, Vol.10, No.6, pp.857–868 (2003).
14) Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. and Watkins, C.: Text Classification using String Kernels, *Journal of Machine Learning Research*, Vol.2, pp.419–444 (2002).
15) Mount, D.W.: *Bioinformatics: Sequence and Genome Analysis*, Cold Spring Harbor Laboratory Press (2001).

16) Nielsen, M.A. and Chuang, I.L.: *Quantum Computation and Quantum Information*, Cambridge Univ Press (2000).
17) Perou, C.M., Sorlie, T., Eisen, M.B., vande Rijn, M., Jeffrey, S.S., Rees, C.A., Pollack, J.R., Ross, D.T., Johnsen, H., Akslen, L.A., Fluge, O., Pergamenschikov, A., Williams, C., Zhu, S.X., Lonning, P.E., Borresen-Dale, A.L., Brown, P.O. and Botstein, D.: Molecular portraits of human breast tumours, *Nature*, Vol.406, No.6797, pp.747–752 (2000).
18) Roweis, S.T. and Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding, *Science*, Vol.290, pp.2323–2326 (2000).
19) Schölkopf, B. and Smola, A.J.: *Learning with Kernels*, MIT Press, Cambridge, MA (2002).
20) Schölkopf, B., Weston, J., Eskin, E., Leslie, C. and Noble, W.S.: A kernel approach for learning from almost orthogonal patterns, *13th European Conference on Machine Learning* (*ECML 2002*) *and 6th European Conference on Principles and Practice of Knowledge Discovery in Databases* (*PKDD'2002*), Elomaa, T., Mannila, H. and Toivonen, H.(eds.), pp.511–528, Springer, Berlin (2002).
21) Smith, N. and Gales, M.J.F.: Speech Recognition using SVMs, *Advances in Neural Information Processing Systems* (2002).
22) Su, Y., Murali, T.M., Pavlovic, V., Schaffer, M. and Kasif, S.: RankGene: identification of diagnostic genes based on expression data, *Bioinformatics*, Vol.19, pp.1578–1579 (2003).
23) Suzuki, J., Hirao, T., Sasaki, Y. and Maeda, E.: Hierarchical Directed Acyclic Graph Kernel: Methods for Structured Natural Language Data, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp.32–39 (2003).
24) Trevor F.Cox, M. A.A.C.: *Multidimensional Scaling, Second Edition*, Chapman & Hall (2000).
25) Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Bostein, D. and Altman, R.B.: Missing value estimation methods for DNA microarrays, *Bioinformatics*, Vol.17, pp.520–525 (2001).
26) Tsuda, K. and Noble, W.: Learning kernels from biological networks by maximizing entropy, *Bioinformatics*, Vol.20, No.Suppl. 1, pp.i326–i333 (2004).
27) Tsuda, K., Kin, T. and Asai, K.: Marginalized kernels for biological sequences, *Bioinformatics*, Vol.18, No.90001, pp. S268–S275 (2002).
28) van't Veer, L.J., Dai, H., vande Vijver, M.J., He, Y.D., Hart, A.A., M, M.M., Peterse, H.L., vander Kooy, K., Marton, M.J., Witteveen, A.T., Schreiber, G.J., Kerkhoven, R.M., Roberts, C., Linsley, P.S., Bernards, R. and Friend, S.H.: Gene expression profiling predicts clinical outcome of breast cancer, *Nature*, Vol.415, pp.530–536 (2002).
29) Vert, J.-P. and Yamanishi, Y.: Supervised graph inference, *Advances in Neural Information Processing Systems 17*, Saul, L.K., Weiss, Y. and Bottou, L.(eds.), Cambridge, MA, MIT Press (2005).
30) Wang, X., Li, A., Jiang, Z. and Feng, H.: Missing value estimation for DNA microarray gene expression data by Support Vector Regression imputation and orthogonal coding scheme, *BMC Bioinformatics*, Vol.7, No.1, p.32 (2004).
31) Weinberger, K.Q., Sha, F. and Saul, L.K.: Learning a kernel matrix for nonlinear dimensionality reduction, *Proc. Twenty First International Conference on Machine Learning* (*ICML 2004*) (2004).
32) Yamanishi, Y., Vert, J.P. and Kanehisa, M.: Supervised enzyme network inference from the integration of genomic data and chemical information, *Bioinformatics*, Vol.21, Suppl.1, pp.i468–i477 (2005).

Appendix

A.1 Derivation of Dual Problem

Let

$$\boldsymbol{\gamma} = \begin{bmatrix} \boldsymbol{\alpha}^\top & \boldsymbol{\beta}^\top \end{bmatrix}^\top \tag{14}$$

$$\boldsymbol{\xi} = \begin{bmatrix} (\boldsymbol{\xi}^{\mathrm{U}})^\top & (\boldsymbol{\xi}^{\mathrm{L}})^\top \end{bmatrix}^\top \tag{15}$$

$$\boldsymbol{c} = \begin{bmatrix} C^{\mathrm{U}}\mathbf{1}_{M^L}^\top & C^{\mathrm{L}}\mathbf{1}_{M^U}^\top \end{bmatrix}^\top \tag{16}$$

$$M = M^{\mathrm{U}} + M^{\mathrm{L}} \tag{17}$$

and define an operator $\mathcal{A}$ by

$$\mathcal{A}\boldsymbol{\gamma} = \mathcal{U}\boldsymbol{\alpha} + \mathcal{L}\boldsymbol{\beta}. \tag{18}$$

The convex problem given in Eq. (9) can be put in the following min-max problem using Lagrangean multipliers,

$$\begin{aligned} \min_{K,\boldsymbol{\xi}} \max_{\boldsymbol{\gamma},\boldsymbol{\delta},\zeta} \; & \mathrm{tr}\,(K \log K) \\ & + \mathrm{tr}\,(K\mathcal{A}\boldsymbol{\gamma}) \\ & + \zeta(\mathrm{tr}\,K - 1) + (\boldsymbol{c} - \boldsymbol{\gamma} - \boldsymbol{\delta})^\top \boldsymbol{\xi} \end{aligned} \tag{19}$$

where $\boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\xi} \in \Re_+^M, \zeta \in \Re, \quad K \in \boldsymbol{S}_+^N$. Therein, $\boldsymbol{S}_+^N$ is the positive semidefinite cone. Eliminating the dummy variables, we readily rewrite the problem as

$$\begin{aligned} \max_{\boldsymbol{\gamma},\zeta} \min_{K} \; & \mathrm{tr}\,(K \log K) + \mathrm{tr}\,(K\mathcal{A}\boldsymbol{\gamma}) \\ & + \zeta(\mathrm{tr}\,K - 1), \\ \text{subj. to } & \mathbf{0}_M \le \boldsymbol{\gamma} \le \boldsymbol{c}, \\ & \boldsymbol{\gamma} \in \Re_+^M, \zeta \in \Re, \quad K \in \boldsymbol{S}_+^N, \end{aligned} \tag{20}$$

For solving the minimization problem inside, we set the derivative with respect to K to zero. Then we have

$$\operatorname{tr}(K\mathcal{A}\boldsymbol{\gamma}) = -\operatorname{tr}(K\log K) - (1+\zeta)\operatorname{tr}(K)$$
$$\operatorname{tr} K = \exp(-\zeta - 1)\operatorname{tr}(\exp(-\mathcal{A}\boldsymbol{\gamma}))\,.$$

After putting them back into the objective function, we get the dual function

$$\exp(-\zeta - 1)\operatorname{tr}(\exp(-\mathcal{A}\boldsymbol{\gamma})) - \zeta \qquad (21)$$

Vanishing the derivative with respect to ζ, we obtain the problem given in Eq. (10).

(Received September 20, 2005)
(Accepted March 20, 2006)

(Editor in Charge: *Ei Banno*)

Tsuyoshi Kato received the B.E., M.E., and Ph.D. degree from Tohoku University, Sendai, Japan, respectively, in 1998, 2000, and 2003. From 2003 to 2005, he was with the National Institute of Advanced Industrial Science Technology (AIST) as a postdoctoral fellow in Computational Biology Research Center (CBRC) at Tokyo. Since 2005, he has been an assistant professor at Graduate School of Frontier Sciences, the University of Tokyo, and he is also a collaborative research fellow of CBRC. His current scientific interests include bioinformatics and statistical pattern recognition. He is a member of IEICEJ.

Wataru Fujibuchi received his Ph.D. degree at the department of biophysics from Kyoto University in 1998. From 1999–2002 he worked as an invited researcher at the NCBI, USA. Now he is hired as a Research Scientist of National Institute of Advanced Industrial Science and has a current position of Visiting Associate Professor, at the Research Institute of IT-Bio, Waseda University. He is an author of Cell Montage database. Research interests: sequence analysis of promoter functions, microarray data analysis, prediction of genetic networks from microarray, integrative analysis of cell features.

Kiyoshi Asai received B.S., M.S. and Ph.D. degrees in Mathematical Engineering from the University of Tokyo, in 1983, 1985 and 1995. He worked in Electrotechnical Labolatory (ETL) from 1985 to 2001, and has been working in Computatinal Biology Research Center (CBRC/AIST) since 2001. His contributions are mainly on hidden Markov models (HMMs) and the other stochastic models and on their applications to biological sequences. His current primary position is Professor in Department of Computational Biology, Graduate School of Frontier Science, the University of Tokyo since 2003.