$$\mathcal{I}_+ \equiv \{i \in \mathbb{N}_\ell | y_i = +1\}, \qquad \mathcal{I}_- \equiv \{i \in \mathbb{N}_\ell | y_i = -1\}.$$

Note that $\mathcal{I}_+ \cup \mathcal{I}_- = \mathbb{N}_\ell$. We now focus on the sets expressed as:

$$\mathcal{V}_+ \equiv \left\{ v_+ \in \mathbb{R}^d \,|\, 0 \le \exists \alpha_i \le \mu_+, v_+ = \sum_{i \in \mathcal{I}_+} \alpha_i x_i, \sum_{i \in \mathcal{I}_+} \alpha_i = 1 \right\},$$

$$\mathcal{V}_- \equiv \left\{ v_- \in \mathbb{R}^d \,|\, 0 \le \exists \alpha_i \le \mu_-, v_- = \sum_{i \in \mathcal{I}_-} \alpha_i x_i, \sum_{i \in \mathcal{I}_-} \alpha_i = 1 \right\},$$

where $\mu_+ \in \mathbb{R}$ and $\mu_- \in \mathbb{R}$ are the predetermined parameters.

Figure 14.1 shows the examples of convex sets, $\mathcal{V}_+$ and $\mathcal{V}_-$, with different values of μ_+ and μ_-. As shown in the figure, we can obtain a variety of classification boundaries by varying the values of μ_+ and μ_-. If we put $\mu_+ = 1/|\mathcal{I}_+|$ and $\mu_- = 1/|\mathcal{I}_-|$, the sets are reduced to $\mathcal{V}_+ = \{m_+\}$ and $\mathcal{V}_- = \{m_-\}$ (Figure 14.1a). If we put $\mu_+ \ge 1$ and $\mu_- \ge 1$, then $\mathcal{V}_+$ and $\mathcal{V}_-$ are the convex hull of the positive training set and the negative training set, respectively (Figure 14.1d). The boundary is the same as that of hard margin SVM [41].

Since every point is represented by using $\alpha \in \mathbb{R}^\ell$ as $v_+ = \sum_{i \in \mathcal{I}_+} \alpha_i x_i$ and $v_- = \sum_{i \in \mathcal{I}_-} \alpha_i x_i$, we wish to find α that represents the closest points. The square Euclidean distance between the two points can be expressed as:

$$\|v_+ - v_-\|^2 = \left\| \sum_{i \in \mathcal{I}_+} \alpha_i x_i - \sum_{i \in \mathcal{I}_-} \alpha_i x_i \right\|^2 = \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle.$$

To find the closest points, we minimize the distance with respect to α that satisfies:

$$\sum_{i \in \mathcal{I}_+} \alpha_i = \sum_{i \in \mathcal{I}_-} \alpha_i = 1, \quad \text{and}$$

$$\forall i \in \mathcal{I}_+ : 0 \le \alpha_i \le \mu_+, \quad \forall i \in \mathcal{I}_- : 0 \le \alpha_i \le \mu_-.$$

We can rearrange the first condition to $\sum_{i=1}^{\ell} \alpha_i = 2$ and $\sum_{i=1}^{\ell} y_i \alpha_i = 0$. By introducing a predetermined constant ν to rescale the variables by $\nu/2$ and setting $\mu_+ = \mu_- = 2/\nu\ell$, the minimization problem in (14.1) to find the closest points can be rewritten as:

$$\begin{aligned} &\min \quad \sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \qquad \text{wrt} \quad \alpha \in \mathbb{R}^\ell, \\ &\text{subj to} \quad \sum_{i=1}^{\ell} \alpha_i = \nu, \quad \sum_{i=1}^{\ell} y_i \alpha_i = 0, \qquad \forall i \in \mathbb{N}_\ell : 0 \le \alpha_i \le \frac{1}{\ell}. \end{aligned} \tag{14.2}$$

This formulation is well known as the ν-SVM classifier [42], which is a variant of SVM. The algorithm of the original SVM classifier [41] is given by:

$$\begin{aligned} &\min \quad \sum_{i=1}^{\ell}\sum_{j=1}^{\ell}\alpha_i\alpha_j y_i y_j \langle x_i, x_j\rangle - 2\sum_{i=1}^{\ell}\alpha_i \quad \text{wrt} \quad \alpha \in \mathbb{R}^{\ell}, \\ &\text{subj to} \quad \sum_{i=1}^{\ell} y_i\alpha_i = 0, \qquad \forall i \in \mathbb{N}_{\ell} : 0 \le \alpha_i \le C. \end{aligned} \tag{14.3}$$

The original SVM classifier requires a predetermined parameter C instead of ν. The decision function produced by ν-SVM can be produced by the original SVM classifier with a suitable choice of C [10].

14.2.2 Kernel Matrix

We now express the formulation in Equation (14.3) by using a matrix notation. Suppose we are given $n(>\ell)$ samples and the first ℓ samples are labeled. We train SVM to predict the labels of the remaining $(n-\ell)$ samples. We use an $n \times n$ matrix K to store the values of the inner-product among input vectors:

$$K_{ij} = \langle x_i, x_j\rangle \quad \text{for} \quad i,j = 1,\ldots,n. \tag{14.4}$$

We call $K \in \mathbb{S}^n$ a *kernel matrix* and partition it as:

$$K = \begin{bmatrix} K^{\text{tra}} & K^{\text{tra,tst}} \\ (K^{\text{tra,tst}})^T & K^{\text{tst}} \end{bmatrix} \tag{14.5}$$

where K^{tra} is an $\ell \times \ell$ symmetric matrix, K^{tst} is $(n-\ell) \times (n-\ell)$ and symmetric, and $K^{\text{tra,tst}}$ is $\ell \times (n-\ell)$. The sub-matrix $K^{\text{tra}} \in \mathbb{S}^{\ell}$ corresponds to the kernel matrix of ℓ labeled samples, and is the data inputted to the SVM algorithm. The matrix form of the optimization problem in Equation (14.3) is expressed as:

$$\begin{aligned} &\min \quad \alpha^T D_y K^{\text{tra}} D_y \alpha - 2\alpha^T 1_{\ell} \quad \text{wrt} \quad \alpha \in \mathbb{R}^{\ell}, \\ &\text{subj to} \quad y^T\alpha = 0, \qquad 0_{\ell} \le \alpha \le C1_{\ell} \end{aligned} \tag{14.6}$$

where $D_y \in \mathbb{S}^{\ell}$ is a diagonal matrix with the i-th diagonal element y_i. The vector $\alpha = [\alpha_1,\ldots,\alpha_{\ell}]^T$ is the variable to be optimized. Note that input vectors themselves are no longer necessary for SVM learning once the values of the inner-products are computed. In other words, the theory of SVM learning can be applied so long as there exists a set of ℓ vectors that produce the symmetric matrix K. Let us examine an example of the kernel matrix. Can a symmetric matrix:

$$K = \begin{bmatrix} 2 & 2 & 4 \\ 2 & 10 & 12 \\ 4 & 12 & 16 \end{bmatrix}$$

be produced by a set of vectors? The answer is yes. Generally, a symmetric matrix can be produced by different sets of vectors. The set of vectors:

$$x_1 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad x_2 = \begin{bmatrix} -3 \\ 1 \end{bmatrix}, \quad x_3 = \begin{bmatrix} -4 \\ 0 \end{bmatrix}$$

produces the matrix K, and the set:

$$x_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

also produces K. One can easily check this using Equation (14.4). Such vectors are called *feature vectors*. On the other hand, there is no set of vectors that produce the following symmetric matrix:

$$K = \begin{bmatrix} 2 & 2 & 4 \\ 2 & 10 & 12 \\ 4 & 12 & 5 \end{bmatrix}.$$

These observations pose the question of how to check whether a symmetric matrix could be applied to SVM learning. This can be done by computing the eigenvalues of the matrix. If all the eigenvalues are non-negative, the matrix can be an input of SVM algorithm. Such a symmetric matrix is said to be *positive semidefinite*, and a formal definition is given as follows:

Definition 14.1 (positive semidefinite, strictly positive definite)
A symmetric matrix $K \in \mathbb{S}^n$ is said to be positive semidefinite if K holds:

$$\forall c \in \mathbb{R}^\ell : c^T K c \geq 0.$$

If $c^T K c > 0$ for all non-zero $c \in \mathbb{R}^\ell$, we say that $K \in \mathbb{S}^n$ is strictly positive definite.

The following two theorems rationalize why we can use the eigenvalues to check whether a symmetric matrix is positive semidefinite and whether there exists a vector set producing the kernel matrix.

Theorem 14.1

A symmetric matrix is positive semidefinite if and only if all the eigenvalues are non-negative.

Theorem 14.2

A symmetric matrix $K \in \mathbb{S}^n$ is positive semidefinite if and only if there exists a set of ℓ vectors that produces K *using* Equation (14.4).

14.2.3 Polynomial Kernel and RBF Kernel

We have already seen Equation (14.4) which is an algorithm producing a positive semidefinite matrix. Equation (14.4) is called the *linear kernel*. There are many other choices to obtain a positive semidefinite matrix. The widely used polynomial kernel and RBF kernel are defined respectively by:

$$K_{ij}^{\text{poly}} = (c^2 + \langle x_i, x_j \rangle)^p, \quad K_{ij}^{\text{rbf}} = \exp\left(-\frac{D^2(x_i, x_j)}{2\sigma^2}\right) \tag{14.7}$$

where $p \in \mathbb{N}$ and $\sigma \in \mathbb{R}$ are constants and are called the degree and the width, respectively. The function $D(\cdot, \cdot)$ gives the Euclidean distance:

$$D(x_i, x_j) = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2}, \tag{14.8}$$

where x_{ik} and x_{jk} are the k-th element of x_i and x_j, respectively. The following two theorems ensure that both kernels always produce a kernel matrix that can be inputted to the SVM algorithm.

Theorem 14.3

Any kernel matrix produced by the polynomial kernel is positive semidefinite.

Theorem 14.4

Any kernel matrix produced by the RBF kernel is positive semidefinite.

Indeed, there exists a mapping function $\phi(\cdot)$ of the input vectors such that the kernel matrix coincides with the inner-products among the feature vectors generated by the mapping function:

$$K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle. \qquad \text{for} \quad \forall i, \forall j \in \mathbb{N}_n.$$

There are several advantages of using the polynomial kernels and the RBF kernels instead of the linear kernel. One advantage of using the polynomial kernels is to incorporate terms of p-th order into a feature vector. Let us examine a simple case of $p = 2$ and $c = 0$. If we define a mapping function:

$$\phi(x) = [x_1 x_1, \ldots, x_d x_1, x_d x_2, \ldots, x_d x_{d-1}, x_1 x_d, \ldots x_d x_d]^T,$$

we have $\langle \phi(x_i), \phi(x_j) \rangle = \langle x_i, x_j \rangle^2$. The derivation is omitted but straightforward.

14.2.4 Pre-process of Kernels

When we analyze data statistically, we often perform pre-processing of the data to remove irrelevant information. We introduce two kinds of pre-processing for analysis with SVM.

14.2.4.1 Normalization

Many studies make the norm of all the feature vectors $\|\phi(x)\|$ unit because norms have little information for classification in many cases. That transformation is called normalization. Any feature vector except zero can be normalized by $\phi(x)/\|\phi(x)\|$. Note that this operation transforms the norm in a unit, although the direction is not changed. This transformation can also be performed only by using the kernel matrix.

The normalized kernel matrix K^{new} is given by:

$$K_{ij}^{\text{new}} = \left\langle \frac{\phi(x_i)}{\|\phi(x_i)\|}, \frac{\phi(x_j)}{\|\phi(x_j)\|} \right\rangle = \frac{\langle \phi(x_i), \phi(x_j) \rangle}{\|\phi(x_i)\| \|\phi(x_j)\|} = \frac{K_{ij}}{\sqrt{K_{ii} K_{jj}}}. \tag{14.9}$$

14.2.4.2 SVD Denoising

A particular drawback of the microarray techniques is that running microarray experiments can be technically rather error prone. Microarray devices may contain dust and scratches that may lead to failure of hybridization and image analysis of some spots that represent gene expression levels. Therefore, the microarray data frequently contain noisy values that may seriously disturb subsequent statistical analysis.

For noise reduction, the approach based on principal component analysis (PCA) is often used in many analytical studies, including microarray analysis. PCA is a tool to extract informative subspaces from the dataset. The subspace is called the principal subspace. We project each feature vector to the principal subspace to eliminate the components in the remaining non-informative subspace. Principal subspace is computed by singular value decomposition (SVD), which is a factorization of a matrix given in the following theorem.

Theorem 14.5

Every matrix $X \in \mathbb{R}^{d \times n}$ can be factorized by two orthonormal matrices $U \in \mathbb{R}^{d \times d}$ and $V \in \mathbb{R}^{n \times n}$ and a diagonal matrix $S \in \mathbb{R}^{d \times n}$ such that:

$$X = USV^T, \tag{14.10}$$

where the diagonal matrix forms:

$$S = \left[\operatorname{diag}\{s_1, \ldots, s_d\}, \quad \mathbf{O}_{d \times (n-d)} \right],$$

when $d \leq n$; otherwise:

$$S = \begin{bmatrix} \operatorname{diag}\{s_1, \ldots, s_n\} \\ \mathbf{O}_{(d-n) \times n} \end{bmatrix}.$$

The term $\operatorname{diag}\{s_1, \ldots, s_r\}$ denotes an $r \times r$ symmetric diagonal matrix with $s_1 \geq s_2 \geq \cdots \geq s_r \geq 0$.

The factorization in Equation (14.10) is termed singular value decomposition, and s_i are singular values. Each column of U and V is called a left singular vector and a right singular vector, respectively. The k-dimensional principal subspace is spanned by the first k left singular vectors $u_1, \ldots, u_k$. The projection of $x \in \mathbb{R}^d$ onto the principal subspace is given by $(U')^T x$ where $U' \equiv [u_1, \ldots, u_k]$. Substituting the projections into Equation (14.4), we obtain the kernel matrix as:

$$K^{svd} = \sum_{i=1}^{k} s_i^2 v_i v_i^T.$$

The value of s_i^2 is equal to the i-th eigenvalue, and v_i coincides with the i-th eigenvector, as seen in the proof of Theorem 2.2.

However, this approach has a drawback. If the true distribution of data without noise were available, we would be able to obtain the exact principal subspace of the true distribution and, therefore, this approach would work well. Typically, however, we know neither the true distribution nor the true principal subspace in advance. Hence, we have to resort the contaminated data themselves to obtain the principal subspace. In this regard, the principal subspace can still be contaminated, and the resulting projections are not often well-denoised. We will discuss an alternative approach to noise reduction in the next section.

14.3 Metrization Kernels: Kernels for Microarray Data

This section introduces three metrization kernels that are produced from distances among data. The distance designed heuristically for microarray data is often non-metric. Those kernels we review in this section are always valid even if the distance is non-metric. SVM performance depends on the quality of a kernel matrix. Some classes of kernel matrices can be explained as a similarity matrix. One class is normalized kernel matrices because the values are the cosine of the angles among the feature vectors. We can say that the RBF kernel is also regarded as a tool that generates a similarity matrix because of its definition. The kernel is defined by a monotonically increasing function, $\exp(\cdot)$, of the negative Euclidean distance between input vectors. This leads to an additional perspective that the RBF kernel is a transformation from a distance matrix to a valid kernel matrix. This motivates us to devise another distance specialized for gene expression to obtain improved kernel matrices.

14.3.1 Partial Distance (or kNND)

The main issue addressed herein is how to handle noisy and missing values that exist in a large portion of a gene expression profile consisting of heterogeneous data. To effectively eliminate such spurious values without removing the entire gene, we devised the following distance. Assume that we have a gene expression table with d genes and n samples where a sample contains $\nu(\times 100)\%$ of noisy genes on average. In such a case only $(1-\nu)$ of genes in that sample contain no noise. Therefore, for any pair of samples, the ratio of common genes not containing noise is expected to be $(1-\nu)^2$. Based on this observation, we devised the following distance:

$$D_p(x_i, x_j) \equiv \min_{\forall \mathcal{I}, s.t. \mathcal{I} \subseteq \mathbb{N}_d, |\mathcal{I}| = d_p} \sqrt{\sum_{k \in \mathcal{I}} (x_{ik} - x_{jk})^2}, \tag{14.11}$$

where $d_p < d$ is a predetermined constant. $|\mathcal{I}|$ represents the cardinality of $\mathcal{I}$. The distance can be computed efficiently as follows: First, we compute the one-dimensional

Euclidean distances $d_h = (x_{ih} - x_{jh})^2$ for $\forall h \in \mathbb{N}_d$. Then we select $k = \lfloor (1-\nu)^2 d \rfloor$ of one-dimensional Euclidean distances d_h from the smallest ones. Finally, we take the sum of the selected d_h as the distance between x_i and x_j. We call this distance the partial distance [19], or the k-nearest neighbor distance (kNND) [15]. For instance, if a sample with $d = 100$ genes contains $\nu = 15\%$ of noisy values, $k = \lfloor (1-0.15)^2 \times 100 \rfloor = 72$ of the smallest distance genes out of the 100 genes are only considered in computing the partial distance between samples.

To classify microarray data, we need a kernel matrix. We consider building a kernel matrix from the partial distances. The RBF kernel is a well-known kernel that is computed from distances among samples. In the previous section we chose Euclidean distances as $D(\cdot, \cdot)$ in Equation (14.7). Generally, the RBF kernel produces a positive semidefinite matrix if and only if $-D^2(\cdot, \cdot)$ is conditionally positive semidefinite [4] (the definition is not shown). The negative squared Euclidean distance generates a conditionally positive semidefinite matrix, but $-D_p^2(\cdot, \cdot)$ does not. Hence, we have to employ another approach to metrization from the partial distances to a kernel matrix.

14.3.2 Maximum Entropy Kernel

We here describe an algorithm called the maximum entropy (ME) kernel [15, 49] to construct a kernel matrix from the partial distances. The algorithm was originally devised to represent an undirected graph, such as an enzyme network or a protein–protein interaction network [49].

Unlike the kernels described in the previous section, the ME kernel does not have any predefined functions. Instead, we obtain the ME kernel in matrix form, K, by basically maximizing the von Neumann entropy defined by:

$$H(K) = -\mathrm{tr}(K \log K - K)$$

with respect to a strictly positive definite matrix K subject to the distance constraints:

$$\forall (i,j) \in \mathcal{E} \quad : \quad \|\phi(x_i) - \phi(x_j)\| \leq D_{ij},$$

where $\mathcal{E} \subseteq \mathbb{N}_n \times \mathbb{N}_n$ is a set of pairs, and D_{ij} is the given upper bound of the distance for pair (i,j). Owing to the distance constraints, the kernel matrix is obtained such that a particular pair of feature vectors must not be distant. The distance constraints are constructed from the partial distance defined in Equation (14.11): $D_{ij} = GD_p(x_i, x_j)$ where G is a constant. Since the partial distance is designed for nearby pairs, we remove the distance constraints for distant pairs. To do this, we design $\mathcal{E}$, a set of pairs to form the distance constraints, from the edges of k-nearest neighbor graph[1)] [33]. In addition to the distance constraints, we restrict the trace of the kernel matrix to be unit to avoid the unlimited divergence of K.

1) The k-nearest neighbor graph is a graph in which an edge is established if a node of the edge is in k nearest neighbor of the other node.

To obtain the kernel matrix we have to solve the optimization problem. However, it cannot be solved analytically. An efficient numerical algorithm is detailed in Appendix 14 A.

14.3.3 Other Distance-Based Kernels

We present here two other metrization kernels to obtain a kernel matrix from distance matrix D_{ij}. Both approaches are originally devised to convert a non-positive semidefinite similarity matrix $S \in \mathbb{S}^n$ into a kernel matrix. The first approach is to take $S^T S$ as a new kernel matrix. The kernel is sometimes called empirical kernel mapping (EKM) [40]. The second approach is to subtract the smallest negative eigenvalue of the similarity matrix S from its diagonal. We call it the Saigo kernel [39]. We obtain a similarity matrix from distance matrix D_{ij} via $S_{ij} = \exp(-D_{ij}/\sigma^2)$.

14.4 Applications to Cancer Data

In this section we compare the classification performances of the six kernels in various cancer data and discuss the differences between metrization (ME, EKM, and Saigo) kernels and standard vectorial data (linear, RBF, and polynomial) kernels. Again, note that the RBF kernel also uses Euclidean distance as the metric of sample (dis-)similarities but cannot use the partial distance (PD) since it violates the positive semidefiniteness of kernels. Figure 14.2 shows a schematic view of the entire analysis

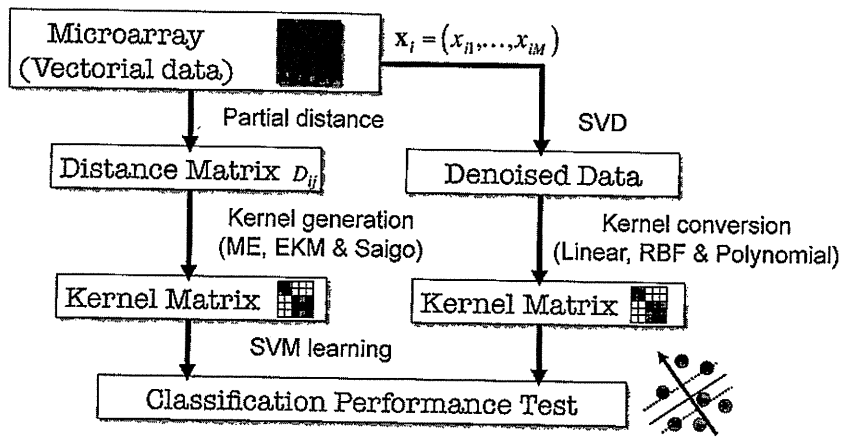

Figure 14.2 Schematic view of the entire process of microarray classification in the metrization (ME, EKM, and Saigo) and vectorial (linear, RBF, and polynomial) kernels. In the metrization kernels, the microarray vectorial data are first converted into a partial distance matrix D_{ij}, generating an optimal kernel matrix that is guaranteed to be positive semidefinite. In the vectorial kernels, the microarray data are first SVD-denoised and directly converted into kernels. Then, the SVM learns the classification boundary from kernel matrices and classifies test samples.

process. We use three examples of cancer datasets: *heterogeneous* human kidney data of normal and renal clear carcinoma tissues, *homogeneous* acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) data with artificial noise, and *heterogeneous* squamous cell carcinoma metastasis in the human head and neck regions.

14.4.1 Leave-One-Out Cross Validation

In the SVM classification analysis, various schemes are available to evaluate accuracies of predictions. In this chapter we simply adopt the standard leave-one-out cross-validation (LOOCV) procedure where each sample is alternatively excluded from the N data and the SVM trained with the remaining $N-1$ samples predicts the excluded one. The exclusion of $1/m \times N$ of data alternatively for use in prediction is generally called "m-fold cross-validation test." All accuracies reported in this chapter are calculated with the following formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}},$$

where TP, FP, TN, and FN are true positive, false positive, true negative, and false negative frequencies, respectively, in the classification. There are various types of equations to evaluate binary classification performances. For example, there are measurements called "sensitivity" and "specificity" that are frequently used to evaluate the prediction power from more specific aspects:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}.$$

The graph obtained when we plot sensitivity against false positive rate (i.e., 1−Specificity) with various SVM boundary thresholds is called a *receiver operating characteristic* (ROC) curve. The area under an ROC curve is also often used for comparison of prediction performances.

14.4.2 Data Normalization and Classification Analysis

Before testing the performance, all the raw data should be properly normalized by being first log-transformed and then scaled to mean 0 and standard deviation 1 (i.e., Z-normalization) in each sample and then each gene. Practically, many genes have a large number of missing values because heterogeneous data are combined; thus, we might need to estimate those values with the rest of the data beforehand. However, in this chapter, since we do not focus on the missing value estimation issues, we will adopt a simple imputation method that all the missing values are replaced with the mean value, that is, 0. Input genes that show high correlation to class labels, or *feature genes*, are selected by the standard two sample t-statistics [38] in each iteration of the LOOCV

test. The distance constraint matrices (D_{ij}) are also generated from the same feature genes. If a sample contains missing values, we again adopt a simple imputation method; we replace the one-dimensional Euclidean distance $(x_{ik}-x_{jk})^2$ with the mean value, that is, 2, if x_{ik} or x_{jk} is missing. Once a dataset is ready, the six kernels are tested with SVMs to analyze their classification performance with various numbers of feature genes and various parameters, as in the next section. The maximum accuracy among the tested parameters for each number of feature genes is recorded as the accuracy for each kernel.

14.4.3 Parameter Selection

Since classification accuracies depend on the parameters in the kernel-SVM method, we need to test various parameter values to obtain the best performance possible. In this chapter, for all the six (linear, polynomial, RBF, EKM, Saigo, and ME) kernels tested here, seven SVM parameters, $C = 10^{-3},10^{-2},10^{-1},1,10,10^2,10^3$, are tested. For the polynomial kernel, $D = 1,2,3,4,5,6,7,8,9,10$ are tested. For the RBF, EKM, and Saigo kernels, $\sigma = 10^{-10},10^{-9},10^{-8},10^{-7},10^{-6},10^{-5},10^{-4},10^{-3},10^{-2},10^{-1},1$ are tested. In the ME kernel, we use only one parameter G that magnifies the distance constraints D_{ij} to adjust the trade-off between over-learning and generalization of classification models (for details, see Reference [19]). The parameter G has to be chosen carefully. When $G \to 0$, typically $K \to 11^T/N$. When $D_{ij} > 2/N$ for $\forall i, \forall j$, $K \to I/N$. The two are somewhat extreme cases. However, if the value of G is positive but too small, SVM will not be able to find the hyperplane separating the positive class from the negative one clearly. Conversely, if the value of G is too large, the so-called diagonal dominant problem [43] ensues. We test the parameter in the range of $G = 2^{-5},2^{-4},2^{-3},2^{-2},2^{-1},1,2,2^2,2^3,2^4,2^5$. To ensure fairness of comparison, it is also important that the total number of parameter combinations in the ME kernel be equal to those in the RBF, EKM, and Saigo kernels in the study.

14.4.4 Heterogeneous Kidney Carcinoma Data

Data of human renal normal tissues and renal clear carcinoma tissues were collected from the public gene expression database GEO–Gene Expression Omnibus [2]. This dataset consists of ten platforms, two of which are spotted DNA/cDNA arrays and eight are variations of Affymetrix-type oligonucleotide arrays. To uniformly analyze the array data from different platforms, we convert as many probe names as possible into standard UniGene (see Reference [1]) identifiers and combine all the data. The total number of UniGenes in the integrated table is as large as 54 674, all of which contain missing values in some platforms; that is, there are no genes common to all platforms. The total number of normal and carcinoma tissue data is 100 (62 normal and 38 carcinoma) and classification analysis between normal and carcinoma is performed.

Figure 14.3 plots the results of the LOOCV test of 100 samples against various numbers (8–296; increasing 8 genes at each step due to computational limitations) of

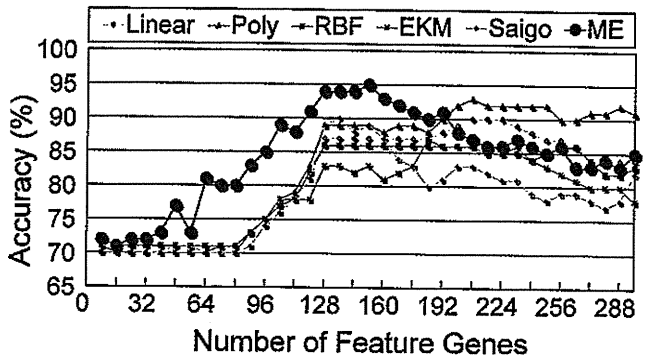

Figure 14.3 Classifications of heterogeneous renal carcinoma data with vectorial and metrization kernels. In most cases, the ME kernel shows much better performance than the linear, polynomial, and RBF kernels and the other two distance-based kernels for various numbers of feature genes. (Modified from Fujibuchi and Kato [15].)

feature genes. The figure shows that the accuracy increases with increasing number of feature genes, plateaus at some region, and decreases, well characterizing typical classification curves. Clearly, the ME kernel performs much better in all cases than the other five kernels for small numbers of feature genes (8–192). In fact, the ME kernel records maximum accuracies of 95.0 (89.5/98.4 sensitivity/specificity)% for 152 feature genes and its accuracies are superior to those of the other five kernels in 64.9% of the tested points (8–296) of feature genes.

14.4.5
Problems in Training Multiple Support Vector Machines for All Sub-data

In the above example of renal carcinoma data, we mix all of the ten sub-data together to train SVMs and predict test samples. As an alternative approach, using vectorial data kernels, it is theoretically possible to train multiple SVMs for all distinct sub-data contained in the composite dataset. However, this approach has practical difficulties in that (i) there are too many heterogeneous sub-data, (ii) some sub-data contain only a few samples, and (iii) some sub-data contain all positive (or negative) samples. The SVMs cannot be trained properly with only a few samples or data with one-sided (positive or negative) labels. In addition, if we do not know the origin (i.e., platform) of the test samples, it would be difficult to determine which SVMs should be used for the classification. Thus, it is very useful to apply the ME kernel to the mixed data because it is much simpler yet quite flexible in this regard.

14.4.6
Effects of Partial Distance Denoising in Homogeneous Leukemia Data

Acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) data for cancer subtype classification have been reported by Golub *et al.* [16] and are often recognized as gold-standard data for microarray classification analysis. There are

72 samples (47 AML and 25 ALL), all of which are quite homogeneous and of good quality, and are thus suitable for artificial noise experiments. To assess the denoising ability of the PD-based ME kernel, we first replace $\nu_{add} \times 100\%$ of original data in a gene expression profile with artificial white noise. The noise is added according to a normal distribution model by $N(0, (2\sigma_{gene})^2)$; a mean of 0 and a standard deviation of twice that of each gene value distribution in the original dataset. Then, we extract 50 feature genes from the training dataset for each iteration of the LOOCV test by the standard t-test.

As the control experiments use linear and RBF kernels, the standard singular value decomposition (SVD) is applied to reduce noise immediately after artificial noise is introduced. In the SVD denoising, three levels of noise removals by different cumulative proportions, 85, 90, and 95%, of eigenvalues are explored. For the ME kernel, the PD denoising method with the following noise level settings is applied. First, we define the total noise level as the sum of the raw noise and the above artificially added noise as $\nu_{raw} + \nu_{add}$, where the raw noise that is assumed to internally exist in the original data is arbitrarily set at $\nu_{raw} = 0.05$. If 10% artificial noise is added, the total noise level is $\nu_{raw} + \nu_{add} = 0.05 + 0.1 = 0.15$ and, according to Equation (14.11), $(1-0.15)^2 \times 100 = 72.3\%$ of the nearest distance genes out of the feature gene set are considered in calculating the PDs between samples.

We repeat the above random noise-adding test ten times and average the highest accuracies among various parameter combinations. Figure 14.4 shows the results. The artificial noise added is within the range of 0–50%. The accuracies decrease gradually with increasing noise levels (10–50%) for the vectorial kernels; for example, the accuracies of the RBF kernel decrease in the order of 96.2, 95.9, 91.0, 82.5, and 79.5%. SVD denoising boosts these accuracies to 98.0, 96.6, 93.2, 91.0, and 86.5%, respectively. Linear and polynomial kernels also show similar accuracies to the RBF kernel when SVD denoising is used.

Interestingly and surprisingly, the three PD-distance-based methods show high accuracies; for example, the PD-ME kernel has an accuracy of 97.8% even at 20%

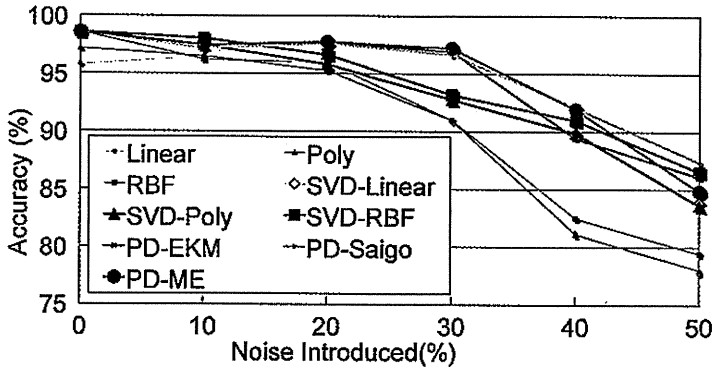

Figure 14.4 AML/ALL classification with artificial noise. The accuracies of standard linear and RBF kernels decrease with increasing noise levels, even with SVD denoising applied, while those of ME and other distance-based kernels with PD denoising are sustained at high levels at 10–30% noise levels. (Modified from Fujibuchi and Kato [15].)

noise level and maintains high accuracies of 97.2 and 92.0% at 30–40% noise levels. The EKM and Saigo kernels using PD-distance also show similar accuracies to the PD-ME kernel. To confirm the superior denoising ability of the PD-based method, results of intensive analysis of the same data with various parameters can be obtained from the author's web site [19].

14.4.7
Heterogeneous Squamous Cell Carcinoma Metastasis Data

We further analyze the total performance of the six kernels with a more practical problem – heterogeneous human squamous cell carcinoma metastasis data. The data consist of four GEO datasets (GSE2280, GSE3524, GSE9349, and GSE2379) from three different platforms (GPL96, GPL201, and GPL91). GSE2280 and GSE3524 are from the same platform (GPL96) but they are from different authors [31, 47]. The four datasets contain 14/8, 9/9, 11/11, and 15/19 metastasis/non-metastasis samples, respectively, and the size of each dataset is too small and not suitable for SVM classification if analyzed separately. However, combining all of the four datasets, we obtain as many as 49 metastasis and 47 non-metastasis samples, making it possible to carry out the SVM classification analysis.

Figure 14.5 shows the results of the LOOCV test for a total of 96 samples against various numbers (1–100; increasing one gene at each step) of feature genes with six different kernels with corresponding denoising methods, namely, SVD-linear, SVD-polynomial, SVD-RBF, PD-EKM, PD-Saigo, and PD-ME. In the PD-ME kernel, five different noise levels, $\nu = 0$ (no noise), 0.05, 0.1, 0.15, and 0.2 are evaluated. In the SVD denoising, five noise removal levels, 80, 85, 90, 95, and 100 (no noise) % of cumulative proportions, which are equal to the number of parameters of the PD denoising experiment, are tested.

The results indicate that the accuracy of the PD-ME kernel mostly exceeds those of the other kernels. The accuracies increase and plateau at around 20–80 feature genes.

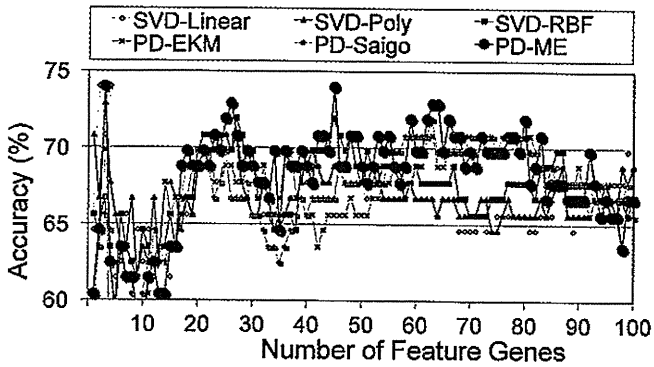

Figure 14.5 Squamous cell carcinoma metastasis classification. Prediction of metastasis by SVMs is performed with gene expression data of squamous cell carcinoma of the human head and neck regions. Classification accuracies of six kernels with corresponding denoising methods are compared.

Note that it is very important to provide robust prediction accuracies in real cancer diagnosis; the regions of 1–20 and 80–100 feature genes give too variant or too low accuracies for use in prediction. The result also indicates that the PD-ME kernel shows relatively stable and high accuracies compared to the other kernels for the proper numbers of feature genes (20–80). The top accuracy rate that the PD-ME kernel performs best among the six kernels in the 20–80 feature gene region is 33 points, which is $33/(80-20+1) = 54.1\%$. An overall maximum accuracy of 74.0 (75.5/72.3 sensitivity/specificity)% is observed for the PD-ME kernel at 45 feature genes, in the 20–80 feature gene region. This accuracy is obtained with the $\nu = 0.15$ denoising parameter.

14.4.8 Advantages of ME Kernel

One of the most remarkable properties of the ME kernel is that the generated kernel matrices always hold positive semidefiniteness, even when the distance matrices for input to the optimization algorithm violate the triangle inequalities at the initial point. This allows us to arbitrarily choose genes from among a set of feature genes to build the distance matrices in a distance-by-distance fashion. Utilizing this property, we introduce the PD denoising method for the distance-based kernels that show better performance than the linear, polynomial, and RBF kernels for leukemia data, even though the data are pre-denoised by SVD. This is quite important in a situation where there are few or heterogeneous samples where SVD may not work properly for denoising because the quality of the eigenvalue decomposition depends on the number of homogeneous samples. Since the PD denoising method only concerns the set of genes between sample pairs, it seems quite robust with regard to the number of samples or the degree of heterogeneity.

Furthermore, the results of kidney carcinoma and squamous cell carcinoma metastasis data in Figures 14.3 and 14.5, respectively, clearly show that the accuracies of the ME kernel exceed those of the other two distance-based kernels, EKM and Saigo. From these observations, the entropy maximization process may work favorably for "heterogeneous" data and allow SVMs to find the discriminant boundaries more easily than the other two distance-based methods, EKM and Saigo.

14.5 Conclusion

Through the analysis presented here, it becomes quite clear that combining similar but distinct data in the microarray analysis may enhance the realistic diagnosis of cancer or other diseases. As shown in our example of metastasis prediction for oral squamous cell carcinoma, each dataset contains only 18–34 samples, which is not suitable for training good SVM predictors. When the datasets are combined, however, the PD-ME kernel demonstrates higher and more robust classification performance than the other kernels, such as linear, polynomial, and RBF kernels

regardless of SVD denoising, and even than the other two distance-based kernels, EKM and Saigo.

One weak point of the ME kernel is its scalability. The ME kernel is given by solving a maximization problem. As the solution cannot be given in a closed form, we have to resort to an iterative algorithm to achieve the kernel matrix. The major reason for the heavy computation is the eigendecomposition of an n by n matrix that is required at each iteration. The eigendecomposition takes $O(n^3)$ computation, disabling us from using the ME kernel when the number of samples is huge [18]. Alternatively, we have employed the steepest descent algorithm [5], which is one of the simplest methods for optimization in this textbook. There are some other smart algorithms that may be able to find a better solution. One is the LBFGS (limited-memory BFGS) formula [24], a derivative of the Newton algorithm [30] that needs a Hessian matrix[2] at each iteration. Although the Newton method is very promising because it usually provides a better solution, computation of the Hessian matrix is time-consuming. The LBFGS algorithm [14, 30], which requires and approximates the Hessian matrix at each iteration by using a compact storage, updates the Hessian matrix efficiently. Such techniques may allow us to convert a large-scale dataset into a proper ME kernel matrix. Moreover, if we could devise a new technique to compute the ME kernel for much larger datasets, the ME kernel will benefit from the semi-supervised setting [54] where unlabeled sample data are mixed with labeled ones in learning. The ME kernel is basically designed to use nearest-neighbor graphs [20], pushing unrelated data points away from related cluster of data. As some samples in a class often form clusters in the data space, even unlabeled sample data may help labeled data to establish clusters, which also improves classification accuracies [11].

In the kernel design field, notably, the trade-off between generalization and specialization is always a problem. For example, to obtain better biological results, the creation of specialized kernels to solve specific biological problems may practically be a good solution. However, too specifically designed kernels lose flexibility and thus cannot be applied to many other problems. Therefore, it will be a major task to learn how to create substantial kernels that would be applicable to various problems in various fields, including biomedical analysis.

Although this chapter has shown the use of SVM as an application of the metrization kernels, the SVM is not the only existing algorithm for kernel-based classification; rather, we can use various kernel methods, such as the kernel Fisher discriminant (KFD) [27] and the relevance vector machine (RVM) [46], as well as variants of the SVM. The KFD solves a linear system to obtain values of the model parameters. The coefficient matrix of the linear system is the sum of the kernel matrix and its scaled identity matrix. A naïve approach to solve the linear system is LU decomposition that requires $O(n^3)$ computation. In the case of the ME kernel, a slightly cleverer approach that utilizes the eigendecomposition of the kernel matrix is available. The eigendecomposition can be executed during kernel generation, where the time complexity is only $O(n^2)$. The total complexity is not changed because

2) A Hessian matrix is a matrix whose elements are the values of the second derivatives.

generating a kernel matrix requires $O(n^3)$. However, only $O(n^2)$ computation is required in the situation that the same kernel matrix is needed for different discriminant tasks [21, 22].

RVM [46] is formulated by a probabilistic model, allowing us to obtain posterior class probabilities that offer a confidence level of the prediction results. Besides classification, kernel matrices can be used for a wide variety of applications, including clustering [55], regression [46], data visualization [28], and novelty detection [42]. Future work remains in the evaluation of the ME kernel in the above situations to explore its possibilities.

Recently, several new cell types, such as induced pluripotent stem (iPS) cells [45] and cancer stem cells (CSCs) [37], have been either created or found. In these research fields, it becomes increasingly important to characterize the features of cells by computational methods before initiating medical treatments to patients. For example, some iPS cells created from various parts of the human body do not have strong multipotency or proliferation ability and sometimes, even worse, have tumorigenesis characteristics. Thus, cell typing using excellent computational methods, such as the kernel-based discriminant analysis for quality control, is required to realize regenerative medicine using iPS cells. In the future, the number of human cell types, regardless of healthy or diseased, to be discriminated is expected to increase. Kernel or kernel-based methods are expected to make immense contributions to a wide variety of biomedical research areas that require accurate and complex cell typings.

14.A Appendix

Proof of Theorem 14.1

Any symmetric matrix $K \in \mathbb{S}^n$ has an eigendecomposition:

$$K = \sum_{i=1}^{n} t_i v_i v_i^T, \tag{14.A.1}$$

where $t_i \in \mathbb{R}$ is the i-th eigenvalue and $v_i \in \mathbb{R}^n$ is the corresponding eigenvector: $\langle v_i, v_j \rangle = \delta_{ij}$ where δ_{ij} is the Kronecker delta. For a d-dimensional arbitrary vector $c \in \mathbb{R}^n$:

$$c^T K c = \sum_{i=1}^{n} t_i c^T v_i v_i^T c = \sum_{i=1}^{n} t_i \langle c, v_i \rangle^2. \tag{14.A.2}$$

Since $\langle c, v_i \rangle^2 \geq 0$ for $\forall i$, the quadratic form $c^T K c$ is non-negative for any c if all the eigenvalue is non-negative. Conversely, suppose at least one of the eigenvalues t_k is negative. If we put $c = v_k$, then:

$$c^T K c = t_k \langle v_k, v_k \rangle^2 + \sum_{i \neq k} t_i \langle v_k, v_i \rangle^2 = t_k < 0 \tag{14.A.3}$$

Hence, Theorem 14.1 is established.

Proof of Theorem 14.2

We first prove the necessary condition. From Theorem 14.1, all the eigenvalues of K are non-negative. Letting $t_i \in \mathbb{R}$ be the i-th eigenvalue and $v_i \in \mathbb{R}^n$ be the corresponding eigenvector, K is produced by the set of column vectors in the matrix:

$$X = \left[\sqrt{t_1}v_1, \ldots, \sqrt{t_n}v_n\right]^T.$$

We next show that the sufficient condition is established. Consider the $d \times n$ matrix whose columns are vectors $X = [x_1, \ldots, x_n]$ producing the kernel matrix K. Denote the SVD of X by $X = USV^T$ where $U \in \mathbb{R}^{d\times d}$ and $V \in \mathbb{R}^{n\times n}$ are *orthonormal* and $S \in \mathbb{R}^{d\times n}$ is *diagonal* whose diagonal elements are $\{s_i\}_{i=1}^{\min(d,n)}$. From the definition, $\forall i : s_i \geq 0$. Substituting SVD into X, we obtain the eigendecomposition of the kernel matrix as:

$$K = X^TX = VS^TU^TUSV^T = VS^TSV^T = \sum_{i=1}^{\min(d,n)} s_i^2 v_i v_i^T. \tag{14.A.4}$$

Notice that the eigenvalues are s_i^2, which are non-negative. Hence, K is positive semidefinite.

Optimization Algorithm for ME Kernel

We here describe a numerical optimization algorithm to obtain the ME kernel. Let M be the number of pairs in $\mathcal{E}$ and denote $\mathcal{E} = \{(i_k, j_k)\}_{k=1}^M$. Furthermore, for simplicity of notation, we define $U_k \in \mathbb{S}^n$ by:

$$U_k \equiv E_{i_k,i_k} + E_{j_k,j_k} - E_{i_k,j_k} - E_{j_k,i_k} - D_{i_k,j_k} I_n \quad \text{for} \quad k \in \mathbb{N}_M.$$

Then, for any kernel matrix $K \in \mathbb{S}^n$ such that $\mathrm{tr}(K) = 1$, the distance constraints can be rewritten as:

$$\forall k \in \mathbb{N}_M : \quad \|\phi(x_{i_k}) - \phi(x_{j_k})\|^2 - D_{i_k j_k} = \mathrm{tr}(U_k K) \leq 0.$$

There does not always exist a kernel matrix that satisfies all the constraints. To keep the optimization problem feasible, we introduce a slack variable $x \in \mathbb{R}_+^M$ and relax the constraints as $\mathrm{tr}(U_k K) \leq \xi_k$ for $\forall k \in \mathbb{N}_M$. The L1-norm of the slack variable is added to the objective function as a penalty. Then the optimization problem is expressed as:

$$\begin{aligned} &\min \quad \mathrm{tr}(K \log K) + \lambda \|\xi\|_1 \\ &\mathrm{wrt} \quad K \in S_{++}^n, \quad \xi \in \mathbb{R}_+^M \\ &\text{subj to} \quad \mathrm{tr}(K) = 1, \qquad \forall k \in \mathbb{N}_M : \mathrm{tr}(U_k K) \leq \xi_k, \end{aligned} \tag{14.A.5}$$

where λ is constant. Since this is a convex problem [6], gradient-based algorithms can easily attain to the optimal solution. An implementation is to solve the dual problem [6] instead of the primal problem given in Equation (14.A.5). The dual problem is described by:

$$\max \quad -\log \mathrm{tr}(\exp(-\mathcal{U}\alpha)) \quad \text{wrt} \quad \alpha \in \mathbb{R}_+^M \quad \text{subj to} \quad \alpha \leq \lambda 1_M \qquad (14.A.6)$$

where α is a dual variable vector [6] and the operator $\mathcal{U}$ performs $\mathcal{U}\alpha = \sum_{k=1}^{M} \alpha_k U_k$. For optimization, the steepest descent method is used. If we denote the objective function of the problem in Equation (14.A.6) by J, the derivatives are given by:

$$\frac{\partial J}{\partial \alpha_k} = \frac{\mathrm{tr}(U_k \exp(-\mathcal{U}\alpha))}{\mathrm{tr}(\exp(-\mathcal{U}\alpha))} \qquad \text{for} \quad \forall k \in \mathbb{N}_M .$$

When the values of some dual variables violate the constraints in (14.A.6) they are forced back into the feasible region. Since the optimization problem is convex, the optimal solution can always be attained from any initial values. Once we obtain the dual optimal solution, we can recover the primal optimal solution as follows:

$$K = \frac{\exp(-\mathcal{U}\alpha)}{\mathrm{tr}(\exp(-\mathcal{U}\alpha))}.$$

References

1 NCBI UniGene project at ncbi. http://www.ncbi.nlm.nih.gov/unigene/, 2006.

2 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. (2005) NCBI GEO: mining millions of expression profiles-database and tools. *Nucleic Acids Res.*, 33 (Database issue), D562–D566.

3 Bennett, K.P. and Bredensteiner, E.J. (2000) Duality and geometry in SVM classifiers, in *Proceedings of the 17th International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc., pp. 57–64.

4 Berg, C., Christensen, J.P.R., and Ressel, P. (1984) *Harmonic Analysis on Semigroups*, Springer-Verlag, New York.

5 Bertsekas, D.P. (2004) *Nonlinear Programming*, Athena Scientific, Belmont, Mass.

6 Boyd, S. and Vandenberghe, L. (2004) *Convex Optimization*, Cambridge University Press.

7 Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Vilo, J., Abeygunawardena, N., Holloway, E., Kapushesky, M., Kemmeren, P., Lara, G.G., Oezcimen, A., Rocca-Serra, P., and Sansone, S.A. (2003) Arrayexpress-a public repository for microarray gene expression data at the ebi. *Nucleic Acids Res.*, 31 (1), 68–71.

8 Burges, C.J.C. (1998) A tutorial on support vector machines for pattern recognition, in *Data Mining and Knowledge Discovery*, vol. 2, Kluwer Academic Publishers, pp. 121–167.

9 Campbell, C. (2001) An introduction to kernel methods, in *Radial Basis Function Networks 1: Recent Developments in Theory and Applications*, Physica Verlag Rudolf Liebing KG, Vienna, pp. 155–192.

10 Chang, C.-C. and Lin, C.-J. (2002) Training ν-support vector regression: theory and algorithms. *Neural. Comput.*, 14, 1959–1977.

11 Chapelle, O., Schölkopf, B., and Zien, A. (2006) *Semi-Supervised Learning*, MIT Press, Cambridge, MA.

12 Crisp, D.J. and Burges, C.J.C. (2000) A geometric interpretation of ν-svm classifiers, in *Advances in Neural Information Processing Systems 12* (eds S.A. Solla, T.K. Leen, and K-.R. Müller), MIT Press.

13 Cristianini, N. and Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*,

Cambridge University Press, Cambridge, UK.
14 Fletcher, R. (1987) *Practical Methods of Optimization*, Wiley-Interscience.
15 Fujibuchi, W. and Kato, T. (2007) Classification of heterogeneous microarray data by maximum entropy kernel. *BMC Bioinformatics*, 8, 267.
16 Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., and Lander, E.S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286 (5439), 531–537.
17 Hastie, T., Tibshirani, R., and Friedman, J.H. (2003) *The Elements of Statistical Learning*, Springer.
18 Kashima, H., Ide, T., Kato, T., and Sugiyama, M. (2009) Recent advances and trends in large-scale kernel methods. *IEICE T. Inf. Syst.*, 92D, 1338–1353.
19 Kato, T., Fujibuchi, W., and Asai, K. (2006) Kernels for noisy microarray data. CBRC Technical Report, AIST-02-J00001-8. http://www.net-machine.net/~kato/pdf/t-kato-cbrctr2006a.pdf.
20 Kato, T., Kashima, H., and Sugiyama, M. (2008) Robust label propagation on multiple networks. *IEEE T. Neural Network.*, 20, 35–44.
21 Kato, T., Kashima, H., Sugiyama, M., and Asai, K. (2009) Conic programming for multi-task learning. *IEEE T. Knowl. Data Eng.* (in press).
22 Kato, T., Okada, K., Kashima, H., and Sugiyama, M. A transfer learning approach and selective integration of multiple types of assays for biological network inference. *Int. J. Knowledge Discovery Bioinformatics (IJKDB)*. (in press).
23 Kondor, R. and Lafferty, J. (2002) Diffusion kernels on graphs and other discrete structures, in *Proceedings 19th International Conference on Machine Learning (ICML) [ICML 2002], San Francisco, CA, USA* (eds C. Sammut and A.G. Hoffmann), Morgan Kaufmann, pp. 315–322.
24 Liu, D.C. and Nocedal, J. (1989) On the limited memory method for large scale optimization. *Math. Program. B*, 45, 503–528.
25 Liu, H., Li, J., and Wong, L. (2005) Use of extreme patient samples for outcome prediction from gene expression data. *Bioinformatics*, 21 (16), 3377–3384.
26 Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002) Text classification using string kernels. *J. Machine Learn. Res.*, 2, 419–444.
27 Mika, S., Rätch, G., Weston, J., and Schölkopf, B. (1999) Fisher discriminant analysis with kernels, in *Neural Networks for Signal Processing IX* (eds Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas), IEEE.
28 Mika, S., Schölkopf, B., Smola, A., Müller, K.R., Scholz, M., and Ratsch, G. (1999) Kernel PCA and de-noising in feature spaces, in *Advances in Neural Information Processing Systems 11* (eds M.S. Kearns, S.A. Solla, and D.A. Cohn), MIT Press, pp. 536–542.
29 Nilsson, B., Andersson, A., Johansson, M., and Fioretos, T. (2006) Cross-platform classification in microarray-based leukemia diagnostics. *Haematologica*, 91 (6), 821–882.
30 Nocedal, J. and Wright, S.J. (2006) *Numerical Optimization*, Springer, New York.
31 O'Donnell, R.K., Kupferman, M., Wei, S.J., Singhal, S., Weber, R., O'Malley, B.Jr., Cheng, Y., Putt, M., Feldman, M., Ziober, B., and Muschel, R.J. (2005) Gene expression signature predicts lymphatic metastasis in squamous cell carcinoma of the oral cavity. *Oncogene*, 24 (7), 1244–1251.
32 Okutsu, J., Tsunoda, T., Kaneta, Y., Katagiri, T., Kitahara, O., Zembutsu, H., Yanagawa, R., Miyawaki, S., Kuriyama, K., Kubota, N., Kimura, Y., Kubo, K., Yagasaki, F., Higa, T., Taguchi, H., Tobita, T., Akiyama, H., Takeshita, A., Wang, Y.H., Motoji, T., Ohno, R., and Nakamura, Y. (2002) Prediction of chemosensitivity for patients with acute myeloid leukemia, according to expression levels of 28 genes selected by genome-wide complementary DNA microarray analysis. *Mol. Cancer Ther.*, 1 (12), 1035–1042.
33 Preparata, F.P. and Shamos, M.I. (1985) *Computational Geometry: An Introduction*, Springer.

34 Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S., and Golub, T.R. (2001) Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl. Acad. Sci. USA*, 98 (26), 15149–15154.

35 Rhodes, D.R., Yu, J., Shanker, K., Deshpande, N., Varambally, R., Ghosh, D., Barrette, T., Pandey, A., and Chinnaiyan, A.M. (2004) Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression. *Proc. Natl. Acad. Sci. USA*, 101 (25), 9309–9314.

36 Rhodes, D.R., Yu, J., Shanker, K., Deshpande, N., Varambally, R., Ghosh, D., Barrette, T., Pandey, A., and Chinnaiyan, A.M. (2004) Oncomine: a cancer microarray database and integrated data-mining platform. *Neoplasia*, 6 (1), 1–6.

37 Rosen, J.M. and Jordan, C.T. (2009) The increasing complexity of the cancer stem cell paradigm. *Science*, 324 (5935), 1670–1673.

38 Rosner, B. (2000) *Fundamentals of Biostatistics*, 5th edn, Duxbury, Pacific Grove, CA.

39 Saigo, H., Vert, J.-P., Ueda, N., and Akutsu, T. (2004) Protein homology detection using string alignment kernels. *Bioinformatics*, 20 (11), 1682–1689.

40 Schölkopf, B., Weston, J., Eskin, E., Leslie, C., and Noble, W.S. (2002) A kernel approach for learning from almost orthogonal patterns, in *13th European Conference on Machine Learning, Helsinki, Finland* (eds S. Thrun, L. Saul, and B. Schölkopf), Springer, pp. 511–528.

41 Schölkopf, B. and Smola, A.J. (2002) *Learning with Kernels*, MIT Press, Cambridge, MA.

42 Schölkopf, B., Smola, A.J., Williamson, R.C., and Bartlett, P.L. (2000) New support vector algorithms. *Neural. Comput.*, 12, 1207–1245.

43 Scholköpf, B., Weston, J., Eskin, E., Leslie, C., and Noble, W.S. (2002) A kernel approach for learning from almost orthogonal patterns, in *Proceedings of ECML 2002i, 13th European Conference on Machine Learning, Helsinki, Finland*, Springer, pp. 511–528.

44 Staunton, J.E., Slonim, D.K., Coller, H.A., Tamayo, P., Angelo, M.J., Park, J., Scherf, U., Lee, J.K., Reinhold, W.O., Weinstein, J.N., Mesirov, J.P., Lander, E.S., and Golub, T.R. (2001) Chemosensitivity prediction by transcriptional profiling. *Proc. Natl. Acad. Sci. USA*, 98 (19), 10787–10792.

45 Takahashi, K., Tanabe, K., Ohnuki, M., Narita, M., Ichisaka, T., Tomoda, K., and Yamanaka, S. (2007) Induction of pluripotent stem cells from adult human fibroblasts by defined factors. *Cell*, 131 (5), 861–872.

46 Tipping, M.E. (2001) Sparse Bayesian learning and the relevance vector machine. *J. Machine Learn. Res.*, 1, 211–244.

47 Torunera, G.A., Ulgera, C., Alkana, M., Galanted, A.T., Rinaggioe, J., Wilkf, R., Tiang, B., Soteropoulosa, P., Hameedh, M.R., Schwalba, M.N., and Dermody, J.J. (2004) Association between gene expression profile and tumor invasion in oral squamous cell carcinoma. *Cancer Genet. Cytogenet.*, 154 (1), 27–35.

48 Tsuda, K., Kin, T., and Asai, K. (2002) Marginalized kernels for biological sequences. *Bioinformatics*, 18 (Suppl. 1), S268–S275.

49 Tsuda, K. and Noble, W.S. (2004) Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, 20 (Suppl. 1), i326–i333.

50 van't Veer, L.J., Dai, H., van de Vijver, M.J., He, Y.D., Hart, A.A.M., Mao, M., Peterse, H.L., van der 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. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415 (6871), 530–536.

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

52 Vert, J.P., Tsuda, K., and Schölkopf, B. (2004) A primer on kernel methods, in *Kernel Methods in Computational Biology*