Simultaneous Learning of Localized Multiple Kernels and Classifier with Weighted Regularization

Naoya Inoue¹ and Yukihiro Yamashita²

¹ Planning & Development Division, Planex Communications Inc. 2-11-9,
Ebisu-Nishi, Shibuya-ku, Tokyo 150-0021 Japan
n708i@yy.ide.titech.ac.jp

² Graduate School of Science and Engineering, Tokyo Institute of Technology,
2-12-1-S6-19, O-okayama, Meguro-ku, Tokyo 152-8553, Japan
yamasita@ide.titech.ac.jp

Abstract. Kernel classifiers have demonstrated their high performance for many classification problems. For the proper selection of kernel functions, multiple kernel learning (MKL) has been researched. Furthermore, the localized MKL (LMKL) enables to set the weights for the kernel functions at each point. However, the training of the weight functions for kernel functions is a complex nonlinear problem and a classifier can be trained separately after the weights are fixed. The iteration of the two processes are often necessary. In this paper we propose a new framework for MKL/LMKL. In the framework, not kernel functions but mappings to the feature space are combined with weights. We also propose a new learning scheme to train simultaneously weights for kernel functions and a classifier. We realize a classifier by our framework with the Gaussian kernel function and the support vector machine. Finally, we show its advantages by experimental results.

1 Introduction

Kernel classifiers such as the support vector machine (SVM) [8], the kernel Fisher discriminant (KFD) [5], etc. have demonstrated their high performance for many classification problems. By mapping an input vector into a high dimensional feature space, nonlinear classification functions in the input space are provided even if they are linear in the feature space. However, to realize such excellent performances, the proper selection of kernel functions is very important. For the purpose the multiple kernel learning (MKL) has been intensively researched [6,1,4,7,3]. Their kernel function is given as a weighed combination of simple kernel functions. In MKL, the weights for kernel functions are spatially uniform. Because the spacial distribution of samples may not be uniform, the best kernel function should depend on its position. To solve this problem, Gönen and Alpaydin proposed the localized multiple kernel learning (LMKL) that can change the weights for kernel functions at each point [2]. However, because the weights are necessary for an unknown input pattern, we have to train the weight

functions for kernel functions before training SVM. It is a complex nonlinear problem. Therefore, their proposed learning process consists of the iteration of 2 processes, training of weight functions for kernel functions and training of SVM.

In this paper, we propose a new framework for MKL or LMKL. A kernel function is given by a weighted combination of kernel functions, and weights for kernel functions and coefficients of classifiers are separated in the original framework, whereas a nonlinear mapping to the feature space is given by a weighted combination of feature mappings, and combined weights for nonlinear mappings and classifiers are introduced in the proposed framework. We also propose a learning process to train the combined weights. We call this learning framework the simultaneous LMKL (SLMKL).

We apply the framework to SVM with the Gaussian kernel function (GKF). This classifier is called SLMKL-SVM. The problem to train the combined weights for SLMKL-SVM is given as a standard convex quadratic problem that can be calculated by using a optimization package such as CPLEX. We also show the advantages of SLMKL-SVM over SVM by experimental results.

We explain the framework of SLMKL in Section 2, and SLMKL-SVM in Section 3. We show experimental results in Section 4, and we conclude this paper and provide future works in Section 5.

2 Framework for SLMKL

Let $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ be a set of samples, where $\mathbf{x}_n$ is a sample pattern and y_n is its label (± 1). The inner product of $\mathbf{x}$ and $\mathbf{z}$ is denoted by $\langle \mathbf{x}, \mathbf{z} \rangle$. In this paper, we consider a linear discriminant model for a binary problem.

In case of the standard kernel method, we fix a Mercer kernel function $k(\mathbf{x}, \mathbf{z})$ and the discriminant function is given by

$$d(\mathbf{x}) = \sum_{n=1}^N \alpha_n k(\mathbf{x}_n, \mathbf{x}) + \theta, \quad (1)$$

where α_n and θ are coefficients and a threshold, respectively. From Mercer's theorem, we can consider there exists a feature mapping $\Phi(\mathbf{x})$ such that $k(\mathbf{x}, \mathbf{z}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle$. Then, the eq.(1) can be written as

$$d(\mathbf{x}) = \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + \theta, \quad \mathbf{w} = \sum_{n=1}^N \alpha_n \Phi(\mathbf{x}_n). \quad (2)$$

In MKL, we prepare several kernel functions $k_j(\mathbf{x}, \mathbf{z})$ ($j = 1, 2, \dots, J$) and construct a kernel function as

$$k(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^J \eta_j k_j(\mathbf{x}, \mathbf{z}), \quad (3)$$

where η_i ($j = 1, 2, \dots, J$) are weights for kernel functions. The discriminant function is also given by eq.(1). In this framework, α_i and η_i are separately trained and the problem to obtain η_i is very complex and nonlinear.

In LMKL, the position dependent weight functions $\eta_j(\mathbf{x})$ ($j = 1, 2, \dots, J$) are used. The kernel function is given by

$$k(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^J \eta_j(\mathbf{x}) k_j(\mathbf{x}, \mathbf{z}) \eta_j(\mathbf{z}), \quad (4)$$

The discriminant function is given by

$$d(\mathbf{x}) = \sum_{n=1}^N \alpha_n \sum_{j=1}^J \eta_j(\mathbf{x}_n) k_j(\mathbf{x}_n, \mathbf{x}) \eta_j(\mathbf{x}) + \theta, \quad (5)$$

The weights for kernel functions can be chosen at every point. However, it seems to be very difficult to obtain $\eta_j(\mathbf{x})$ from training data because $\mathbf{x}$ is an unknown data in the classification stage so that another training process is necessary to obtain $\eta_j(\mathbf{x})$. Therefore, its learning process consists of the iteration of 2 processes, training of $\eta_j(\mathbf{x})$ and training of α_n and θ .

In the proposing SLMKL, we assume that we can prepare $\Phi_j(\mathbf{x})$ ($j = 0, 1, 2, \dots, J$) of which inner product can be calculated analytically. Their kernel functions are denoted by

$$k_{i,j}(\mathbf{x}, \mathbf{z}) \equiv \langle \Phi_i(\mathbf{x}), \Phi_j(\mathbf{z}) \rangle. \quad (6)$$

Although the existence of inner product seems to be a strong condition, in Section 4, we show the inner product of GKF's of which kernel parameters are different. Because GKF is widely used in many recognition problems, the proposing framework can be used widely.

The model of $\mathbf{w}$ in eq.(2) is defined by J feature mappings as

$$\mathbf{w} = \sum_{n=1}^N \sum_{j=1}^J \alpha_{j,n} \Phi_j(\mathbf{x}_n). \quad (7)$$

The parameters $\alpha_{j,n}$ express both weights for kernel functions and coefficients for a classifier. Since it is difficult to fix weights for feature mappings for an unknown input pattern, we used only Φ_0 for the feature mapping into which an unknown input pattern is substituted. Then, the discriminant function is given by

$$d(\mathbf{x}) = \langle \mathbf{w}, \Phi_0(\mathbf{x}) \rangle + \theta = \sum_{n=1}^N \sum_{j=1}^J \alpha_{j,n} k_{j,0}(\mathbf{x}_n, \mathbf{x}) + \theta. \quad (8)$$

The advantages of this frame work are as follows. **(1)** The weights for feature mappings can be changed at each sample point individually. **(2)** It does not need a continuous weight functions for feature mappings whereas LMKL has $\eta_j(\mathbf{x})$. **(3)** The coefficients $\alpha_{j,n}$ can express simultaneously weights for kernel functions and coefficients for a classifier and can be trained similarly to the original classifier.

3 SLMKL-SVM

The criterion of SVM is given as minimizing

$$\|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n \quad (9)$$

under the condition that for $n = 1, 2, \dots, N$,

$$y_n d(\mathbf{x}_n) - 1 + \xi_n \geq 0, \quad \xi_n \geq 0, \quad (10)$$

where C is a constant. The first term of eq.(9) comes from minimizing the margin $1/\|\mathbf{w}\|$. It can be considered as a regularization term. The second term is the hinge loss.

In this section, we consider the simplest case of SLMKL in order to investigate basic features of SLMKL. Let $J = 2$ and we use two feature mappings Φ_1 and Φ_2 . We use Φ_2 for Φ_0 that is the feature mapping into which an unknown input pattern is substituted. Then, $\mathbf{w}$ can be expressed by $\mathbf{w} = \mathbf{w}_1 + \mathbf{w}_2$, where

$$\mathbf{w}_j = \sum_{n=1}^N \alpha_{j,n} \Phi_j(\mathbf{x}_n) \quad (j = 1, 2). \quad (11)$$

However, even if we substitute eqs.(8) and (11) to eqs.(9) and (10), respectively, the result of the training is reduced to a standard SVM. The representer theorem [9] ensures that $\mathbf{w}$ has to be spanned only by $\{\Phi_2(\mathbf{x}_n)\}_{n=1}^N$. That implies $\alpha_{1,n} = 0$ for all n and the classifier is a standard SVM.

We have another problem in the framework of SLMKL with respect to the ratio of weights between for $\alpha_{1,n}$ and for $\alpha_{2,n}$. The main term in the discriminant function or the hinge loss is

$$\langle \mathbf{w}, \mathbf{x} \rangle = \sum_{n=1}^N \sum_{i=1}^2 \alpha_{i,n} k_{i,2}(\mathbf{x}_n, \mathbf{x}). \quad (12)$$

In the training stage, sample points are substituted into $\mathbf{x}$. The dominant weights for $\alpha_{1,n}$ and $\alpha_{2,n}$ are given by $k_{1,2}(\mathbf{x}_n, \mathbf{x}_n)$ and $k_{2,2}(\mathbf{x}_n, \mathbf{x}_n)$, respectively. For example, we assume that GKF in eq.(16) is used and we let $M = 20$ and $\sigma_1 = 3\sigma_2$. Then, the former is approximately 10^{-8} and the latter is 1. On the other hand, in the regularization term

$$\|\mathbf{w}\|^2 = \sum_{m=1}^N \sum_{n=1}^N \sum_{i=1}^2 \sum_{j=1}^2 k_{i,j}(\mathbf{x}_m, \mathbf{x}_n) \alpha_{i,m} \alpha_{j,n}, \quad (13)$$

their dominant weights for them are $k_{1,1}(\mathbf{x}_n, \mathbf{x}_n)$ and $k_{2,2}(\mathbf{x}_n, \mathbf{x}_n)$, respectively, and both are 1. Accordingly, the regularization for $\alpha_{1,n}$ is much stronger than that for $\alpha_{2,n}$.

We explain this problem in the point of view of linear algebra. Let S_j be the subspace spanned by $\{\Phi_j(\mathbf{x}_n)\}_{n=1}^N$ ($j = 1, 2$). When $k_{1,2}(\mathbf{x}_n, \mathbf{x}_n)$ is small, S_1 and S_2 are nearly orthogonal. The vector $\mathbf{w}$ is included in $S_1 + S_2$, and $\mathbf{w}$ is evaluated with vectors in $S_1 + S_2$ in the regularization term. However, it is evaluated only with vectors in S_2 in the hinge loss term. Then, in the term, the weight for $\alpha_{1,n}$ becomes smaller than that for $\alpha_{2,n}$.

In order to solve this problem, we have to weaken the regularization for $\alpha_{1,n}$ and we introduce the following five new weighted regularization terms.

- (1) Projected regularization (PrR): Let P_2 be the orthogonal projection operator onto S_2 . The regularization term is defined by $\|P_2 \mathbf{w}\|^2$. By projecting $\mathbf{w}$ onto S_2 in the regularization, the norm of projection of a vector in S_1 becomes be comparable with the values of inner products with a vector in S_2 .
- (2) Training sample regularization (TSR): The regularization term is given by $\sum_{n=0}^N |\langle \mathbf{w}, \Phi_{\sigma_2}(\mathbf{x}_n) \rangle|^2$. Because the sum of squared norms of $\langle \mathbf{w}, \Phi_2(\mathbf{x}_n) \rangle$, which are the same terms in the hinge loss, is used for the regularization, the strength of regularization is balanced.
- (3) Peak Regularization (PkR): $\alpha_{j,n}$ in the regularization term is replaced by $k_{j,2}(\mathbf{x}_n, \mathbf{x}_n) \alpha_{j,n}$. It weakens the regularization of $\alpha_{1,n}$ by multiplying $k_{1,2}(\mathbf{x}_n, \mathbf{x}_n)$ whereas that of $\alpha_{2,n}$ does not change it because $k_{2,2}(\mathbf{x}_n, \mathbf{x}_n) = 1$. The regularization term is given by

$$\sum_{m=1}^N \sum_{n=1}^N \sum_{i=1}^2 \sum_{j=1}^2 k_{i,2}(\mathbf{x}_m, \mathbf{x}_m) k_{j,2}(\mathbf{x}_n, \mathbf{x}_n) k_{i,j}(\mathbf{x}_m, \mathbf{x}_n) \alpha_{i,m} \alpha_{j,n}. \quad (14)$$

- (4) Sum of kernel function regularization (SKFR): $\alpha_{j,n}$ in the regularization term is replaced by $\sum_{m=1}^N k_{j,2}(\mathbf{x}_n, \mathbf{x}_m) \alpha_{j,n}$. The strengths of the regularizations for $\alpha_{1,n}$ and $\alpha_{2,n}$ are changed by $\sum_{m=1}^N k_{1,2}(\mathbf{x}_n, \mathbf{x}_m)$ and $\sum_{m=1}^N k_{2,2}(\mathbf{x}_n, \mathbf{x}_m)$, respectively. PkR uses the value of a kernel function with $\mathbf{x}_n$ itself for $\alpha_{j,n}$ but SKFR uses sum with all samples.
- (5) Square root of sum of kernel function regularization (SSKFR):
 $\alpha_{j,n}$ in the regularization term is replaced by $\sqrt{\sum_{m=1}^N k_{j,2}(\mathbf{x}_n, \mathbf{x}_m)} \alpha_{j,n}$. Since the change of weights in SKFR seems to be large, we use the square root of SKFR although it seems to be heuristic.

With each regularization term, the criterion can be transformed to a constrained quadratic optimization problem.

4 Experiments

We show experimental results of a toy problem and 13 types of UCI datasets used in [5]. We use the Gaussian kernel function (GKF) for a feature mapping. We prepare two kernel parameters σ_1 and σ_2 . Then, the feature mapping with σ_j from a point $\mathbf{x}$ to a function is defined by

$$(\Phi_j(\mathbf{x}))(z) = \left(\frac{2}{\pi \sigma_j^2} \right)^{\frac{M}{2}} \exp(-\|\mathbf{z} - \mathbf{x}\|^2 / \sigma_j^2). \quad (15)$$

Fig. 1. Results for artificial data

for $j = 1, 2$. From eq.(6), the kernel function for SLMKL is given by

$$k_{i,j}(\mathbf{x}, \mathbf{z}) = \left(\frac{2\sigma_i\sigma_j}{\sigma_i^2 + \sigma_j^2} \right)^M \exp \left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{\sigma_i^2 + \sigma_j^2} \right). \quad (16)$$

We let $\sigma_1^2 = r\sigma^2$ and $\sigma_2^2 = (2-r)\sigma^2$. If $r = 1$, it reduces to the standard SVM. We set $r = 1.5$ and $r = 1.8$ for SLMKL-SVM.

In order to see the basic performance of SLMKL-SVM, we conducted an experiment with artificial data. As shown in Fig. 1, the domain is a two-dimensional region $D = \{(x, y) | x \in [0, 1], y \in [0, 0.6]\}$. The original boundary (dashed line) consists of two kinds of sinusoidal functions. It is more complex in the right section of D . Furthermore, samples (small square or cross points) are denser in the region $[0.4, 1] \times [0.1, 0.5]$. The error rate is evaluated by 201×201 uniform samples in D . The parameters (σ^2 and C) are set to provide the best recognition rate for each classifier. The error rates for SVM, and SLMKL-SVM with PrR, TSR, PkR, SKFR, and SSKFR are 12.95%, 11.18%, 14.07%, 11.42%, 10.79%,

Table 1. Properties of datasets

Dataset name	# of training patterns	# of test patterns	# of realizations	dimension of data
banana	400	4900	100	2
breast cancer	200	77	100	9
diabetis	468	300	100	8
flare solar	666	400	100	9
german	700	300	100	20
heart	170	100	100	13
image	1300	1010	20	18
ringnorm	400	7000	100	20
splice	1000	2175	20	60
thyroid	140	75	100	5
titanic	150	2051	100	3
twonorm	400	7000	100	20
waveform	400	4600	100	21

Table 2. Error rates for UCI datasets

DN	SVM	SLMKL-SVM $r = 1.5$				
		PrR	TSR	PkR	SKFR	SSKFR
Ba	11.53±0.66	10.43±0.42	11.21±4.25	10.47±0.50	10.71±0.53	10.91±0.63
Br	26.04±4.74	26.43±4.77	25.70±4.32	25.48±4.53	27.49±4.74	27.60±4.81
Di	23.53±1.73	23.97±1.65	24.15±4.54	23.49±1.64	24.09±1.77	23.27±1.70
Fl	32.43±1.82	32.33±1.80	33.70±1.98	32.37±1.78	32.86±3.50	32.36±1.78
Ge	23.71±2.20	23.47±2.10	24.19±2.21	23.54±2.23	23.68±2.16	23.49±2.37
He	15.95±3.26	15.65±3.28	16.63±7.29	15.96±3.31	15.79±3.19	15.53±3.37
Im	2.96±0.60	3.10±0.53	2.89±0.48	3.04±0.51	3.63±0.88	3.68±1.52
Ri	1.66±0.12	1.47±0.11	30.31±21.77	1.48±0.11	1.72±0.12	1.58±0.10
Sp	10.88±0.66	10.80±0.67	10.79±0.68	11.00±0.66	11.06±0.70	11.09±0.75
Th	4.80±2.19	4.31±2.21	7.47±3.20	4.97±2.25	4.72±2.25	4.60±2.40
Ti	22.42±1.02	22.77±1.11	23.06±5.34	22.75±1.12	22.77±1.18	22.36±1.00
Tw	2.96±0.23	2.39±0.12	2.95±0.34	2.41±0.13	2.42±0.13	2.49±0.16
Wa	9.88±0.43	9.98±0.49	10.52±4.94	9.99±0.52	10.07±0.40	10.19±0.47

(Values in bold font indicates the best result in the row.)

and 12.23%, respectively. The calculated boundaries (solid lines) of SVM and SLMKL-SVM with SKFR are shown for their best parameters.

In Fig. 1 (b), a small square point expresses training data $\mathbf{x}_n$ where $\langle \mathbf{w}_1, \Phi_2(\mathbf{x}_n) \rangle > \langle \mathbf{w}_2, \Phi_2(\mathbf{x}_n) \rangle$ and a small cross point expresses the opposite. We can see large and small kernel parameters are selected in smooth and complicated region, respectively, by the training.

We also conducted experiments using the 13 UCI datasets listed in Table 1. For example, the 'banana' dataset has 100 realizations and each realization has 400 training and 4900 test patterns of which dimension is two.

Table 3. Error rates for UCI datasets

DN	SVM	SLMKL-SVM $r = 1.8$				
		PrR	TSR	PkR	SKFR	SSKFR
Ba	11.53±0.66	10.41±0.44	10.80±0.50	10.69±0.55	10.69±0.53	10.70±0.54
Br	26.04±4.74	25.56±4.71	26.23±5.67	25.74±4.71	27.58±4.78	27.38±4.76
Di	23.53±1.73	23.38±1.67	23.37±1.81	23.85±1.59	23.60±1.76	23.67±1.79
Fl	32.43±1.82	32.37±1.78	33.42±1.51	32.36±1.78	32.59±2.85	32.36±1.79
Ge	23.71±2.20	23.66±2.19	24.15±2.13	23.73±2.17	23.55±2.33	23.70±2.14
He	15.95±3.26	15.26±3.16	15.62±3.09	16.00±3.23	15.57±3.22	16.10±3.28
Im	2.96±0.60	3.08±0.56	2.85±0.45	3.05±0.54	3.14±0.62	3.25±0.58
Ri	1.66±0.12	1.45±0.10	28.45±22.68	1.45±0.10	1.91±0.20	1.63±0.12
Sp	10.88±0.66	10.77±0.71	10.84±0.61	10.77±0.71	11.14±0.71	11.11±0.71
Th	4.80±2.19	4.44±2.22	6.55±3.12	4.83±2.26	4.64±2.22	4.36±2.06
Ti	22.42±1.02	22.43±1.02	22.46±1.07	24.91±8.47	23.00±4.81	22.69±1.56
Tw	2.96±0.23	2.45±0.14	3.00±0.23	2.40±0.12	2.54±0.16	2.45±0.13
Wa	9.88±0.43	10.60±2.30	9.73±0.45	10.38±0.47	10.49±0.43	10.55±0.35

(Values in bold font indicates the best result in the row.)

Table 4. p -value of t -test against SVM

DN	$r = 1.5$					$r = 1.8$				
	PrP	TSR	PkR	SKFR	SSKFR	PrP	TSR	PkR	SKFR	SSKFR
Ba	0.000	0.230	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Br	0.718	0.299	0.198	0.984	0.989	0.238	0.601	0.328	0.988	0.976
Di	0.966	0.897	0.434	0.987	0.144	0.268	0.263	0.912	0.611	0.712
Fl	0.349	1.000	0.407	0.860	0.392	0.407	1.000	0.392	0.681	0.393
Ge	0.319	0.971	0.410	0.592	0.352	0.565	0.964	0.655	0.424	0.618
He	0.260	0.801	0.509	0.364	0.187	0.066	0.233	0.543	0.205	0.626
Im	0.775	0.347	0.670	0.995	0.969	0.736	0.263	0.685	0.816	0.931
Ri	0.000	1.000	0.000	1.000	0.000	0.000	1.000	0.000	1.000	0.040
Sp	0.356	0.341	0.711	0.790	0.817	0.312	0.424	0.312	0.875	0.846
Th	0.059	1.000	0.705	0.400	0.270	0.126	1.000	0.538	0.305	0.073
Ti	0.989	0.879	0.984	0.987	0.338	0.527	0.606	0.998	0.879	0.924
Tw	0.000	0.404	0.000	0.000	0.000	0.000	0.889	0.000	0.000	0.000
Wa	0.936	0.900	0.947	0.999	1.000	0.999	0.009	1.000	1.000	1.000

The parameters σ and C were selected by 5-fold cross validation as explained in [5]. The averaged error rate and its standard deviation over all realizations are shown in Tables 2 and 3. Their p -values of t -test against SVM are shown in Table 4. The results of SVM are referred from [5].

We can see the advantage of the proposed method especially for 'banana', 'ringnorm', 'twonorm' and 'waveform'. However, because the result for 'ringnorm' with TSR is very bad, we have to research on the combination of mappings to the feature space and regularization in the criterion.

5 Conclusion

In this paper, we proposed a new framework for MKL or LMKL. In the framework, not kernel functions but feature mappings are linearly combined. We also proposed the SLMKL-SVM with weighted regularizations. We conducted experiments and showed its advantages.

For future work, we have to research on feature mappings of which inner product can be obtained analytically and investigate regularization terms to improve its performance.

References

1. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the smo algorithm. In: ICML (July 2004)
2. Gönen, M., Alpaydin, E.: Localized multiple kernel learning. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, pp. 352–359. ACM, New York (2008), <http://doi.acm.org/10.1145/1390156.1390201>
3. Gönen, M., Alpaydin, E.: Multiple kernel learning algorithm. *Journal of Machine Learning Research* 12, 2211–2268 (2011)
4. Lanckriet, G.R.G., Bie, T.D., Cristianini, N., Jordan, M.I., Noble, W.S.: A statistical framework for genomic data fusion. *Some Fine Journal* 20(16), 2626–2635 (2004)
5. Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Müller, K.R.: Fisher discriminant analysis with kernels. In: *Neural Networks for Signal Processing IX*, pp. 41–48. IEEE (1999)
6. Pavlidis, P., Weston, J., Cai, J., Grundy, W.N.: Gene functional classification from heterogeneous data. In: *Proceedings of the Fifth Annual International Conference on Computational Biology, RECOMB 2001*, pp. 249–255. ACM, New York (2001), <http://doi.acm.org/10.1145/369133.369228>
7. Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large scale multiple kernel learning. *J. Mach. Learn. Res.* 7, 1531–1565 (2006), <http://dl.acm.org/citation.cfm?id=1248547.1248604>
8. Vapnik, V.N.: *Statistical Learning Theory*. Wiley, New-York (1998)
9. Wahba, G.: *Spline Models for Observational Data* (CBMS-NSF Regional Conference Series in Applied Mathematics). Society for Industrial and Applied Mathematics, Philadelphia (1990)