Neural Process Lett (2015) 42:745–762
DOI 10.1007/s11063-014-9392-3

A Novel Multiple Kernel Learning Method Based on the Kullback–Leibler Divergence

Zhizheng Liang · Lei Zhang · Jin Liu

Published online: 25 October 2014

Abstract In this paper we develop a novel multiple kernel learning (MKL) model that is based on the idea of the multiplicative perturbation of data in a new feature space in the framework of uncertain convex programs (UCPs). In the proposed model, we utilize the Kullback–Leibler divergence to measure the difference between the estimated kernel weights and ideal kernel weights. Instead of directly handling the proposed model in the primal, we obtain the optimistic counterpart of its Langrage dual in terms of the theory of UCPs and solve it by using the alternating optimization. In the case of a varying parameter, the proposed model gives the solution path from a robust combined kernel to some combined kernel corresponding to the initially ideal kernel weights. In addition, we also give a simple strategy to select the initial kernel weights as the ideal kernel weights if any prior knowledge of kernel weights is not available. Experimental results on several data sets show that the proposed model can obtain competitive performance with some of the state-of-the-art MKL algorithms.

Keywords MKL · Uncertain convex programs · Robust counterparts · Optimistic counterparts · KL divergence · Data classification

1 Introduction

Multiple kernel learning (MKL) [1–4] has gained some successful applications in many areas such as scene classification and bioinformatics. Unlike single kernel leaning algorithms where the kernel parameters within a kernel are optimized, the aim of MKL is to search for a linear

Z. Liang · L. Zhang · J. Liu
Department of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China

Z. Liang (✉)
School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China
e-mail: liang@cumt.edu.cn

or nonlinear combination of predefined kernels by optimizing some performance measures. The predefined kernels in MKL generally encode different information from multiple diverse sources and thus MKL gives the enough flexibility in dealing with data with multiple sources. In addition to that, MKL which obtains an optimal kernel from multiple kernels also provides a possible strategy of kernel selection in kernel-based learning methods. Consequently, MKL has an advantage in achieving good generalization performance in many learning tasks.

With the advance in MKL, many MKL algorithms have been proposed during the past several years. In the original work of Lanckriet et al.[5], MKL is formulated as a semidefinite programming (SDP) problem. In [6], the SDP problem is further reduced to a second-order cone programming (SOCP) problem [7]. However, these two MKL algorithms are only suitable for handling small-scale or medium-scale data sets due to the high computational cost of SDP or SOCP. In order to improve the efficiency of MKL, some fast MKL algorithms [8–16] have been proposed. Among them, the alternating optimization [14] may be one of the most widely used methods, where the kernel weights and the coefficients of the span of samples are alternately updated. In [8], Sonnenburg et al. proposed a semi-infinite linear programming (SILP) problem where a cutting plane method is used to update kernel weights. In order to avoid the instability of the solution in SILP, Rakotomamonjy et al. [10] used a reduced gradient method to learn kernel weights. Obviously one advantage of the alternating optimization in solving MKL is that the existing SVM solvers can be directly utilized in some cases. In addition to finding fast algorithms for solving MKL, some researchers also designed novel MKL models where kernel weights are usually controlled by different regularizers. Kloft et al. [4] developed L_p-norm MKL which generalizes L_1-norm MKL. In order to make the trade-off between the orthogonal information and sparse kernel weights, a generalized MKL model by imposing an elastic-net-type constraint on kernel weights is proposed in [12]. In [17], the entropy of kernel weights is used as the regularizer to control kernel weights in MKL. Overall, these models further contribute to the development of MKL.

Recently, there has also been an increasing interest in developing uncertain convex programs (UCPs) [18,19]. UCPs can deal with data uncertainty by using an uncertainty set consisting of every possible value of some parameters [20–23]. UCPs are associated with the robust counterpart of the convex program and the optimistic counterpart of its uncertain Lagrange dual. In [19], it is shown that strong duality between the robust counterpart and the optimistic counterpart holds under the Slater-type strict feasibility condition. In [20], Jeyakumar and Li pointed out that strong duality between the robust counterpart and the optimistic counterpart holds for uncertain polyhedral convex programming problems. In [21], Li et al. further relaxed the condition of strong duality between the robust counterpart and the optimistic counterpart, in which robust strong duality holds for partially finite convex program problems.

In UCPs, data uncertainty may be introduced if the input data is contaminated by noise or measurement errors. Data uncertainty may also be modeled as two typical forms: the additive perturbation and the multiplicative perturbation. The additive perturbation is that the data is contaminated by additive noise or errors while the multiplicative perturbation is that the data is contaminated by multiplicative noise or errors. In fact, we find that the kernel weights in L_1-norm MKL take values in some sets. If we regard the kernel weights in L_1-norm MKL as uncertain parameters in UCPs and think that data is perturbed by uncertain parameters in the multiplicative form, classical L_1-norm MKL will become the optimistic counterpart of an SVM-based optimization problem. This new explanation for L_1-norm MKL allows us to develop new MKL models which are based on the robust counterpart of SVM-based

optimization problems. Consequently, in this paper we propose a novel MKL model from the viewpoint of UCPs. Note that the solution of the proposed model is feasible no matter how the data is perturbed in an uncertainty set. We also introduce the KL divergence to measure data uncertainty in the new feature space. Thus our model may yield good performance in some classification problems, compared with some MKL models. As often done in UCPs, we resort to solving the optimistic counterpart of the Langrage dual of our model if strong duality holds. It is interesting to note that the proposed model can be transformed into a convex optimization problem. But we adopt the alternating optimization to solve it in this paper. In addition, a strategy for setting initially ideal kernel weights in the proposed model is also given. As shown in the experiments, the proposed model gives the enough flexibility to control kernel weights by varying an important parameter and achieves comparable performance with previous MKL methods.

The rest of this paper is organized as follows. In Sect. 2, we briefly review L_1-norm MKL and UCPs. In Sect. 3, we develop an optimization model and give an effective algorithm to solve this model. Moreover, we also discuss the problem of selecting the initially ideal kernel weights. In Sect. 4, some experiments are done to evaluate the proposed model. The final section is the conclusion.

2 Brief Review of L_1-Norm MKL and UCPs

2.1 L1-Norm Multiple Kernel Learning

Let $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1,\ldots,n}$ be a set of training samples, where $\boldsymbol{x}_i \in R^m$ and $y_i \in \{-1, 1\}$. In MKL, a group of mappings $\psi_j : R^m \to \boldsymbol{H}_j (j = 1, \ldots, d)$ are given, each associated with a kernel $\boldsymbol{K}_j$ in a reproducing kernel Hilbert space (RKHS). L_1-norm MKL based on the margin maximization can be formulated as [4]

$$\min_{\tilde{\boldsymbol{w}}, b, \theta} f(\tilde{\boldsymbol{w}}, b, \boldsymbol{\theta}) := \frac{1}{2} \sum_{k=1}^{d} \|\tilde{\boldsymbol{w}}_k\|^2 + C \sum_{i=1}^{n} \left| 1 - y_i \left(\sum_{k=1}^{d} \left\langle \tilde{\boldsymbol{w}}_k, \sqrt{\theta_k} \boldsymbol{\psi}_k(\boldsymbol{x}_i) \right\rangle + b \right) \right|_+$$
$$\text{s.t.} \sum_{k=1}^{d} \theta_k = 1, \theta_k \geq 0, \tag{1}$$

where $|z|_+ = z$ if $z \geq 0$ and 0 otherwise, $\psi_k(x_i)$, $\tilde{\boldsymbol{w}}_k \in H_k$, d is the number of RKHSs, b is a bias, and C is a regularization parameter. Applying $\tilde{\boldsymbol{w}}_k = \boldsymbol{w}_k \sqrt{\theta_k}$ to Eq. (1), one can obtain

$$\min_{\boldsymbol{w}, b, \theta} f(\boldsymbol{w}, b, \boldsymbol{\theta}) := \frac{1}{2} \sum_{k=1}^{d} \theta_k \|\boldsymbol{w}_k\|^2 + C \sum_{i=1}^{n} \left| 1 - y_i \left(\sum_{k=1}^{d} \langle \theta_k \boldsymbol{w}_k, \boldsymbol{\psi}_k(\boldsymbol{x}_i) \rangle + b \right) \right|_+$$
$$\text{s.t.} \sum_{k=1}^{d} \theta_k = 1, \theta_k \geq 0. \tag{2}$$

The objective function in Eq. (2) is non-smooth since it contains the form of $|z|_+$. By introducing extra variables, one can transform Eq. (2) into a smooth objective function under proper constraints. In addition, Eq. (1) or Eq. (2) may also be converted to a convex optimization problem [4].

2.2 Uncertain Convex Programs

The uncertain convex optimization problem [18,19] can be described as

$$\begin{aligned} &\min_{\boldsymbol{x}}\{f(\boldsymbol{x}, \boldsymbol{\mu}_0)\} \\ &\text{s.t. } g_i(\boldsymbol{x}, \boldsymbol{\mu}_i) \leq 0, i = 1, \ldots, d, \end{aligned} \tag{3}$$

where f and g_i are convex functions with respect to the variable $\boldsymbol{x}$, and the uncertain parameters $\boldsymbol{\mu}_i (i = 0, \ldots, d)$ lie in convex compact uncertainty sets such as polytopes and ellipsoids. The variable $\boldsymbol{x}$ is robust feasible for Eq. (3) if it satisfies the constraints for every possible value of $\boldsymbol{\mu}_i (i = 0, \ldots, d)$ in an uncertainty set. The robust counterpart of Eq. (3) is defined as

$$\begin{aligned} &\min_{\boldsymbol{x}} \max_{\boldsymbol{\mu}_0}\{f(\boldsymbol{x}, \boldsymbol{\mu}_0)\} \\ &\text{s.t.} \max_{\boldsymbol{\mu}_i} g_i(\boldsymbol{x}, \boldsymbol{\mu}_i) \leq 0, \quad i = 1, \ldots, d. \end{aligned} \tag{4}$$

The Lagrange dual of Eq. (3) is formulated as

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}, \boldsymbol{\mu}_0) + \sum_{i=1}^{d} \lambda_i g_i(\boldsymbol{x}, \boldsymbol{\mu}_i) \right\}. \tag{5}$$

The optimistic counterpart of Eq. (5) is defined as

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu}_0, \ldots \boldsymbol{\mu}_d} \min_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}, \boldsymbol{\mu}_0) + \sum_{i=1}^{d} \lambda_i g_i(\boldsymbol{x}, \boldsymbol{\mu}_i) \right\}. \tag{6}$$

In general, the optimal value of the objective function in Eq. (4) is not smaller than that of the objective function of Eq. (6). It is found in [20,21] that the strong duality between Eqs.(4) and (6) holds under proper conditions. That is, the optimal value of the objective function in Eq. (4) equals that of the objective function of Eq. (6) under certain conditions. In such a case, the optimization problem of Eq. (4) can be transformed into that of Eq. (6) and thus one solves Eq. (6) instead of Eq. (4) if directly solving Eq. (4) is difficult.

3 Our Proposed Method

In this section, we first introduce an optimization model based on the idea of UCPs. Then we give its optimistic counterpart of its Lagrange dual and solve it by using the alternating optimization. In addition, we also discuss how to set the initially ideal kernel weights.

3.1 Problem Formulation

From Eq. (1), one can observe that each sample is mapped into a new feature space by a composite map $\psi_{\boldsymbol{\theta}} = \sqrt{\theta_1}\boldsymbol{\psi}_1 \times \cdots \times \sqrt{\theta_d}\boldsymbol{\psi}_d$. It is observed that the projected data $\sqrt{\theta_k}\boldsymbol{\psi}_k(x_i)(i = 1, \ldots, n)$ will vary as the parameters $\theta_k (i = 1, \ldots, d)$ change. Thus $\sqrt{\theta_k}\boldsymbol{\psi}_k(x_i)$ can be considered as the multiplicative perturbation of $\boldsymbol{\psi}_k(x_i)$ in terms of the parameter θ_k. This actually corresponds to data uncertainty in the new feature space. This is different from the additive perturbation in [21] where the data is perturbed in the original feature space rather than in the mapping feature space. Based on this point, we regard Eq. (1) as the optimistic counterpart of $f(\tilde{\boldsymbol{w}}, b, \boldsymbol{\theta})$, where $\tilde{\boldsymbol{w}}$ and b are unknown variables and $\boldsymbol{\theta}$ is

an uncertain parameter taking values in a simplex. This also gives a new explanation for Eq. (1) in terms of the uncertain parameter $\boldsymbol{\theta}$ and thus Eq. (1) falls into the framework of UCPs. When the optimal $\boldsymbol{\theta}$ is obtained from Eq. (1), one can achieve an optimal combined kernel in terms of $\boldsymbol{\theta}$ and each kernel in RKHSs. In such a case, we refer to the optimal combined kernel from Eq. (1) as the best combined kernel in an uncertainty set. From the theory of UCPs, the best combined kernel from Eq. (1) may yield too optimistic results on the training set. Since the projected data can be regarded as data uncertainty in the feature space, we also give the following robust counterpart of $f(\tilde{\boldsymbol{w}}, b, \boldsymbol{\theta})$ in the framework of UCPs.

$$\begin{aligned} &\min_{\tilde{\boldsymbol{w}},b}\max_{\boldsymbol{\theta}} f(\tilde{\boldsymbol{w}}, b, \boldsymbol{\theta}) := \frac{1}{2}\sum_{k=1}^{d}\|\tilde{\boldsymbol{w}}_k\|^2 + C\sum_{i=1}^{n}\left|1 - y_i\left(\sum_{k=1}^{d}\left\langle\tilde{\boldsymbol{w}}_k, \sqrt{\theta_k}\boldsymbol{\psi}_k(\boldsymbol{x}_i)\right\rangle + b\right)\right|_+ \\ &\text{s.t.}\sum_{k=1}^{d}\theta_k = 1, \theta_k \geq 0. \end{aligned} \tag{7}$$

A solution $(\tilde{w}, b)$ is robust feasible if it satisfies Eq. (7) for every possible realization of uncertain parameters. From Eq. (7), one obtains an optimal combined kernel in terms of $\boldsymbol{\theta}$ and each kernel in RKHSs. In contrast to the optimal combined kernel in Eq. (1), we refer to the optimal combined kernel from Eq. (7) as the robust combined kernel. Thus Eq. (7) explores the performance of a novel MKL model in the case of the robust combined kernel. Substituting $\tilde{\boldsymbol{w}}_k = \boldsymbol{w}_k\sqrt{\theta_k}$ into Eq. (7), one can obtain

$$\begin{aligned} &\min_{\boldsymbol{w},b}\max_{\boldsymbol{\theta}} f(\boldsymbol{w}, b, \boldsymbol{\theta}) := \frac{1}{2}\sum_{k=1}^{d}\theta_k\|\boldsymbol{w}_k\|^2 + C\sum_{i=1}^{n}\left|1 - y_i\left(\sum_{k=1}^{d}\left\langle\theta_k\boldsymbol{w}_k, \boldsymbol{\psi}_k(\boldsymbol{x}_i)\right\rangle + b\right)\right|_+ \\ &\text{s.t.}\sum_{k=1}^{d}\theta_k = 1, \theta_k \geq 0. \end{aligned} \tag{8}$$

The model of Eq. (8) is different from that of Eq. (2) since the latter is to search for the best combined kernel in an uncertainty set and the former is to explore the robust combined kernel in an uncertainty set. Thus from the viewpoint of data perturbation, Eq. (8) may be much more suitable for dealing with the data that belong to an uncertainty set than Eq. (2). Note that $f(\boldsymbol{w}, b, \boldsymbol{\theta})$ in Eq. (8) is a convex function with respect to the variable $\boldsymbol{\theta}$ for fixed $\boldsymbol{w}$ and b. However, it is a maximization problem with respect to the variable $\boldsymbol{\theta}$. Specifically, the inner layer optimization in Eq. (8) is a concave optimization problem which is generally NP-hard [24]. Due to this concave optimization problem, one may obtain the optimal solution $\boldsymbol{\theta}$ from extreme points [24]. That is, the optimal solution $\boldsymbol{\theta}$ may be obtained from one of extreme points of a simplex. Assume that the kth element of $\boldsymbol{\theta}$, i.e. θ_k, takes one and other elements of $\boldsymbol{\theta}$ are zeros, which gives a very sparse solution of $\boldsymbol{\theta}$. What's worse, if the kernel matrix corresponding to the kth kernel is also positive semi-definite, not positive definite, e.g. a kernel matrix whose elements are 1, one may obtain $\|\boldsymbol{w}_k\|^2 = 0$. In such a case, the combined kernel is the kth kernel and the margin that equals $\frac{2}{\|\boldsymbol{w}_k\|}$ becomes the infinity. This causes all the training samples to be classified as one class and yields too pessimistic results, which shows that severe underfitting occurs. In order to avoid this case, one possible strategy is to adopt d strictly positive definite kernel matrices with good class separability. However, selecting the kernels with good class separability is not straightforward. To this end, we provide a strategy to handle this case. Assume that d ideal kernel weights $q_k(k = 1, \ldots, d)$ are given. Considering that the KL divergence measures data uncertainty in an uncertainty set and also measures the difference between $\theta_k(k = 1, \ldots, d)$ and $q_k(k = 1, \ldots, d)$, we add the KL divergence to Eq. (8) and propose the following optimization model.

$$\min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\zeta}} \max_{\theta} f(\boldsymbol{w}, b, \boldsymbol{\theta}, \boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{v}) := \frac{1}{2}\sum_{k=1}^{d}\theta_k \|\boldsymbol{w}_k\|^2 + C\sum_{i=1}^{n}\xi_i - \gamma\sum_{k=1}^{d}\theta_k \log\frac{\theta_k}{q_k}$$
$$\text{s.t. } y_i\left(\sum_{k=1}^{d}\left\langle\theta_k\boldsymbol{w}_k, \boldsymbol{\psi}_k(\boldsymbol{x}_i)\right\rangle + b\right) \geq 1-\zeta_i, \zeta_i \geq 0, i = 1, \ldots, n,$$
$$\sum_{k=1}^{d}\theta_k = 1, \theta_k \geq 0, \tag{9}$$

where $q_k (k = 1, \ldots, d)$ are the ideal kernel weights reflecting the importance of kernel functions, $\sum_{k=1}^{d} q_k = 1$, and γ is a nonnegative parameter which controls the trade-off between the robust combined kernel and some combined kernel. It is observed that the inner layer optimization in Eq. (9) is not a concave optimization problem due to the introduction of the KL divergence. Based on the theory of UCPs, we give the following optimistic counterpart of the uncertain Lagrange dual in terms of Eq. (9) [20].

$$\max_{\boldsymbol{\theta},\boldsymbol{\alpha},\boldsymbol{v}} \min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\zeta}} f(\boldsymbol{w}, b, \boldsymbol{\theta}, \boldsymbol{\zeta}, \boldsymbol{\alpha}, \boldsymbol{v}) := \frac{1}{2}\sum_{k=1}^{d}\theta_k \|\boldsymbol{w}_k\|^2 + C\sum_{i=1}^{n}\xi_i - \gamma\sum_{k=1}^{d}\theta_k \log\frac{\theta_k}{q_k}$$
$$+\sum_{i=1}^{n}\alpha_i\left(1-\zeta_i - y_i\left(\sum_{k=1}^{d}\left\langle\theta_k\boldsymbol{w}_k, \boldsymbol{\psi}_k(\boldsymbol{x}_i)\right\rangle + b\right)\right) - \sum_{i=1}^{n}\zeta_i v_i$$
$$\text{s.t.} \sum_{k=1}^{d}\theta_k = 1, \theta_k \geq 0, \alpha_i \geq 0, v_i \geq 0, \zeta_i \geq 0, i = 1, \ldots, n. \tag{10}$$

Since $\theta_k (k = 1, \ldots, d)$ are in a compact set which is finitely generated from d kernel matrices, Eq. (9) belongs to the robust counterpart of partially finite convex programs [20,21]. By introducing an extra variable in the objective function in Eq. (9), one can resort to Corollary 3.3 in [20] to show that the constraint qualification in UCPs for Eq. (9) is satisfied and that the strong duality between Eq. (9) and Eq. (10) holds. To be specific, the optimal value of the objective function of Eq. (9) equals that of the objective function of Eq. (10). In such a case, one can solve Eq. (10) since directly solving Eq. (9) is not easy. In the following subsection, we will continue to discuss how to solve Eq. (10).

3.2 Optimization Strategy

It is not difficult to verify that Eq. (10) is equivalent to the following optimization problem.

$$\max_{\boldsymbol{\alpha},\boldsymbol{\theta}} f(\boldsymbol{\alpha}, \boldsymbol{\theta}) := -\frac{1}{2}\sum_{k=1}^{d}\theta_k \boldsymbol{\alpha}^T diag(\boldsymbol{y}) K_k diag(\boldsymbol{y})\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\boldsymbol{e} - \gamma\sum_{k=1}^{d}\theta_k \log\frac{\theta_k}{q_k}$$
$$\text{s.t.} \sum_{i=1}^{n}\alpha_i y_i = 0, \quad C \geq \alpha_i \geq 0, i = 1, \ldots, n, \sum_{k=1}^{d}\theta_k = 1, \theta_k \geq 0, \tag{11}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ and $\boldsymbol{e} = (1, \ldots, 1)^T$. Without loss of generality, we assume that the first l samples in the training set belong to the positive class. Let $\boldsymbol{\beta}^k = diag(\boldsymbol{y})\theta_k\boldsymbol{\alpha} (k = 1, \ldots, d)$. Applying some operations to Eq. (11), one can obtain

$$\max_{\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_d,\boldsymbol{\theta}} f(\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_d,\boldsymbol{\theta}) := -\frac{1}{2}\sum_{k=1}^{d}\frac{1}{\theta_k}(\boldsymbol{\beta}^k)^T K_k \boldsymbol{\beta}^{(k)} + \sum_{k=1}^{d}\boldsymbol{y}^T\boldsymbol{\beta}^k - \gamma\sum_{k=1}^{d}\theta_k \log\frac{\theta_k}{q_k}$$

$$\text{s.t.} \sum_{k=1}^{d}\boldsymbol{e}^T\boldsymbol{\beta}^k = 0, \quad C \geq \sum_{k=1}^{d}\beta_s^k \geq 0 \quad (s = 1,\ldots,l)$$

$$-C \leq \sum_{k=1}^{d}\beta_t^k \leq 0, \quad (t = l+1,\ldots,n), \sum_{k=1}^{d}\theta_k = 1, \theta_k \geq 0. \tag{12}$$

It is not hard to find that the objective function in Eq. (12) with respect to $(\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_d,\boldsymbol{\theta})$ is concave, so Eq. (12) is a convex optimization problem. But we find that directly solving Eq. (12) may meet some computational problems. On the one hand, if kernel weights approach zeros, computing the weighted kernels may cause the computational problem since they lie in the denominator in Eq. (12). On the other hand, the number of unknown variables in Eq. (12) is $dn + d$, which is greater than that of unknown variables in Eq. (11). Thus solving Eq. (12) may be computationally expensive if there are too many kernels or training samples. Due to these disadvantages in solving Eq. (12), in this paper we will focus on solving Eq. (11) instead. It is of interest to note that Eq. (11) is a bi-convex optimization problem. That is, for fixed $\boldsymbol{\alpha}$, optimizing $\boldsymbol{\theta}$ is a convex optimization problem; for fixed $\boldsymbol{\theta}$, optimizing $\boldsymbol{\alpha}$ is a convex optimization problem. It is found from Eq. (11) that optimizing $\boldsymbol{\alpha}$ involves solving an SVM problem for fixed $\boldsymbol{\theta}$. Thus any existing SVM software can be used to obtain $\boldsymbol{\alpha}$. For fixed $\boldsymbol{\alpha}$, one can obtain the solution $\boldsymbol{\theta}$ in terms of the method in [25]. However, it is of interest to note that $\boldsymbol{\theta}$ has the following closed-form solution from Eq. (11) (Appendix).

$$\theta_k = \frac{q_k e^{\frac{-\sum_{i,j}\alpha_i\alpha_j y_i y_j K_k(\boldsymbol{x}_i,\boldsymbol{x}_j)}{2\gamma}}}{\sum_{s=1}^{d} q_s e^{\frac{-\sum_{i,j}\alpha_i\alpha_j y_i y_j K_s(\boldsymbol{x}_i,\boldsymbol{x}_j)}{2\gamma}}}, (k = 1,\ldots,d). \tag{13}$$

For the sake of completeness, we briefly summarize the algorithm for solving Eq. (11) in the following, which is an alternative to Eq. (9).

Algorithm 1: The solution to Eq. (11)

1: given $q_k (k = 1,\cdots,d)$, C, and γ,
2: **for** $t = 1, 2, \cdots$ do
  2.1: Solve the dual SVM problem based on Eq. (11) in fixed $\boldsymbol{\theta}$;
  2.2: Update the kernel weights based on Eq. (13);
  2.3: if the stopping criterion is met, then break;
3: end **for**

3.3 Analysis and Discussion

From Algorithm 1, one can observe that it involves alternately solving the SVM problem and updating kernel weights until a stopping criterion is met. The stopping criterion in Algorithm 1 may be based on the maximal number of iterations or the relative change of the two consecutive objective function values in Eq. (11). Hence our algorithm for solving Eq. (11) is relatively easy to implement but effective. It is also noted that the overall complexity of our algorithm strongly depends on that of SVMs since updating kernel weights needs a relatively

low computational cost. In general, the faster the SVM algorithm one uses, the more efficient our method. Note that we use the alternating optimization method to solve Eq. (11). Since Eq. (11) is not a convex optimization problem, one generally obtains locally optimal solutions of Eq. (11). However, in the experiments, we find that our algorithm gives better performance than other MKL algorithms in some cases.

From Eq. (13), one sees that the smaller that $\sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{K}_k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the larger the kernel weight θ_k. If one regards $\frac{2}{\|\boldsymbol{w}_k\|}$ as the margin of the kth kernel, $\sqrt{\sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{K}_k(\boldsymbol{x}_i, \boldsymbol{x}_j)}$ is inversely proportional to the margin of the kth kernel. From this point, one knows that the larger the margin of the kth kernel, the bigger the kernel weight θ_k. Thus our model prefers to take the large weight for the kernel with the large margin. However, it should be pointed out that this strategy is reasonable only for the case that data is linearly separable in each feature space. In practice, it turns out that the data is linearly separable in each implicitly high-dimensional space if one adopts Gaussian kernels and each sample in the training set is different. If data is not linearly separable in each feature space, the large margin corresponding to a single kernel does not mean the large margin for the combined kernel. Moreover, $\sum_{i=1}^{n} \xi_i$ in Eq. (10) also affects the number of misclassified samples. Thus the large margin for some kernel which has low class separability, e.g. the infinite margin, may result in too many misclassified training samples. By introducing ideal weights, we expect to alleviate the effect of this kind of kernels. From Eq. (13), one can see that θ_k will take a small value or zero if the initially ideal kernel weight q_k is relatively small or zero. That is, if some kernel with low class separability or poor classification performance has a small weight, then its updating weight also takes a small value from Eq. (13).

In addition, one can observe from Algorithm 1 that the parameter γ should be set in advance and it will directly affect the optimal value of the objective function in Eq. (11). In the following, we give several remarks on the parameter γ.

Remark 1 Let $I(\boldsymbol{\alpha}^*) = \{i = \arg\max_j f(\boldsymbol{\alpha}^*, \theta_j^*)\}$,where $\boldsymbol{\alpha}^*$ and θ_j^* are the optimal solution of Eq. (11) in given C and $\gamma = 0$. If $\gamma \to 0^+$ and q_k $(k = 1, \ldots, d)$ are equal, it is not difficult to derive that the limit of θ_k in Eq. (13) is zero if $k \notin I(\boldsymbol{\alpha}^*)$ and is a nonzero constant if $k \in I(\boldsymbol{\alpha}^*)$. This also shows that $\theta_k (k = 1, \ldots, d)$ may be sparse if γ takes sufficiently small values. From Eq. (7), one can see that this corresponds to the robust combined kernel. Specifically, the robust combined kernel is selected from an uncertainty set in such a case.

Remark 2 If $\gamma \to +\infty$ and q_j $(j = 1, \ldots, d)$ are not equal to zero, one obtains $\theta_j = q_j$ from Eq. (13). Substituting θ_j of Eq. (13) into $\gamma \sum_{j=1}^{d} \theta_j \log \frac{\theta_j}{q_j}$, one can verify that $\gamma \sum_{j=1}^{d} \theta_j \log \frac{\theta_j}{q_j}$ approaches zero if $\gamma \to +\infty$, i.e., $\lim_{\gamma \to \infty} \gamma \sum_{j=1}^{d} \theta_j \log \frac{\theta_j}{q_j} = 0$. It is also noted that one obtains $\theta_j = 1/d$ $(j = 1, \ldots, d)$ if the kernel weights $q_j = 1/d$ $(j = 1, \ldots, d)$ and $\gamma \to +\infty$. This shows that in such a case Eq. (11) degenerates into SVMs with the uniform kernel weights. In other words, SVMs with uniform kernel weights are an extreme case of Eq. (11).

Remark 3 It is clear that if the parameter γ varies from zero to infinity, one can obtain the solution path from the robust combined kernel to some combined kernel from ideal kernel weights. For example, if one can take ideal kernel weights by solving Eq. (2), one can obtain the solution path from the robust combined kernel to the best combined kernel, which builds a bridge between Eqs.(11) and (2) by the parameter γ.

3.4 The Setting of Initially Ideal Kernel Weights

In this subsection, we will show how to set the initially ideal kernel weights in the proposed model in order to obtain good classification performance. The parameters $q_j (j = 1, \ldots, d)$ in the proposed model generally provide prior information (distribution) of kernel weights and should approach ideal kernel weights. In general, ideal weights should be that they take small values for the kernels with poor classification performance or low class separability and vise versa. Indeed, ideal kernel weights may be obtained by solving Eq. (1). However, in real applications, we expect to obtain ideal kernel weights in advance. To this end, in what follows we will provide a kind of ideal kernel weights based on class separability, which is independent of known MKL methods. Assume that $\boldsymbol{K_j} (j = 1, \ldots, d)$ are d kernel matrices obtained from the training samples. We regard the ith column of $\boldsymbol{K_j}$ as the projection of x_i in the jth kernel space. Based on this idea, we can obtain the within-class scatter matrix and the between-class scatter matrix in each kernel space as follows.

$$\boldsymbol{S}_b^j = \sum_{i=1}^{c} n_i (\boldsymbol{\mu}_i^j - \bar{\boldsymbol{\mu}}^j)(\boldsymbol{\mu}_i^j - \bar{\boldsymbol{\mu}}^j)^T, \tag{14}$$

$$\boldsymbol{S}_w^j = \sum_{i=1}^{c} \sum_{x_s \in class\ i} (\boldsymbol{K}_j(:, s) - \boldsymbol{\mu}_i^s)(\boldsymbol{K}_j(:, s) - \boldsymbol{\mu}_i^s)^T, \tag{15}$$

where c is the number of classes of the training samples, $\boldsymbol{K}_j(:, s)$ denotes the sth column of $\boldsymbol{K}_j$, n_i is the number of samples in class i, $\boldsymbol{\mu}_i^j = \frac{1}{n_i} \sum_{x_s \in class\ i} \boldsymbol{K}_j(:, s)$, and $\bar{\boldsymbol{\mu}}^j = \frac{1}{n} \sum_{s=1}^{n} \boldsymbol{K}_j(:, s)$. From Eqs.(14) and (15), we define the discriminant power of the jth kernel as follows.

$$dp_j = \frac{tr(\boldsymbol{S}_b^j)}{tr(\boldsymbol{S}_w^j)}, \quad j = 1, \ldots, d. \tag{16}$$

From Eq. (16), we find that $dp_j (j = 1, \ldots, d)$ take nonnegative values. According to the discriminant power of each kernel from Eq. (16), we set the relative discirmniant power(RDP) of kernels as initial kernel weights $q_j (j = 1, \ldots, d)$,denoted by

$$q_j = \frac{dp_j}{\sum_{i=1}^{d} dp_i} j = 1, \ldots, d. \tag{17}$$

It is observed from Eq. (17) that the stronger the discriminant power of some kernel, the larger the corresponding kernel weight. Thus in our implementation we prefer to set large weights to the kernel with strong discriminant power. Note that if q_j is zero, then θ_j will always be zero from Eq. (13). In other words, if the discriminant power of some kernel is zero, we will not consider this kernel in our scheme. In addition, one can find from Eq. (17) that setting kernel weights by Eq. (17) can also reduce the scaling problem of kernel matrices. That is, if one kernel is proportional to another kernel, both kernels obtain the same ideal weights.

4 Experimental Results

In this section, we conduct a series of experiments to evaluate the performance of the proposed model. For comparison purposes, we evaluate the following MKL methods.

(1) Uniform MKL (UMKL): We use the average kernel combination of predefined kernels. The parameter $C \in \{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ and we implement UMKL by using LibSVM [26] in Matlab version.
(2) L_1-norm MKL(L_1-MKL): The parameter C is set as done in UMKL and we implement L_1-MKL(Algorithm 1 in [8])in terms of Matlab.
(3) L_p-norm MKL(L_p-MKL)[4]: The parameter $p \in \{32/31, 16/15, 8/7, 4/3, 2, 4, 8, 16, 1024, 10240\}$ and C is set as done in UMKL. We implement L_p-MKL (Algorithm 1 in [4]) in terms of Matlab.
(4) Nesterov's method for smooth MKL (NMKL): The parameter $\lambda \in \{1/16, 1/4, 1, 4, 16, 64, 256, 1024, 10240, 10^8\}$ and C is set as done in UMKL. We implement NMKL (Algorithm 2 in [17]) in terms of Matlab.
(5) Variable sparsity kernel learning (VSKL) [1]: The parameter q in [1] is set as p in L_p-MKL and C is set as done in UMKL. Considering the fact that parameter $\boldsymbol{\gamma}$ in VSKL seems to be similar to the parameter $\boldsymbol{\theta}$ in our paper due to the maximum problem, we only consider kernels in different groups and do not consider kernels within a group due to the minimization problem in our implementation of VSKL (Algorithm 1 in [1]) in terms of Matlab. This makes the kernels in different groups not be sparse, which is pointed out in VSKL and is also observed in our experiments.
(6) Uncertainty-Set SVM (USSVM) [22]: The parameter C is set as done in UMKL and the parameter κ in [22] is set to 1. We use all the predefined kernels in our paper as the base kernels and do not consider the nominal kernel. We implement USSVM (SOCP) based on Mosek optimization software [27] in Matlab version.
(7) Ours(U): Initial kernel weights in our model are equal. The parameters C and γ in Eq. (11) are taken from $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. We implement Eq. (11) by using LibSVM [26] in Matlab version. The stopping criterion in our algorithm is that the maximal number of iterations is 100 or the relative change of the two consecutive objective function values is smaller than 0.001.
(8) Ours(D):Initial kernel weights in our model are obtained from Eq. (17) and other settings are similar to those of Ours(U):

We use the following predefined kernels in all the methods we implement, and also deal with each kernel with sphere normalization as done in [4] in order to obtain good classification performance.

(a) Gaussian kernels with different widths $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ on all features.
(b) Polynomial kernels of degree 2–7 on all features.

4.1 The Experiments on the Artificial Data

In this set of experiments, we design a two-class problem in $\Re^{20}$. For this data set, each feature of the first 10 dimensions of data points from the first class is sampled from the Gaussian distribution with mean 0.5 and variance 2, each feature of the first 10 dimensions of data points from the second class is sampled from the Gaussian distribution with mean -0.5 and variance 2, and the other 10 dimensions of data points for two classes are sampled from the Gaussian distribution with mean 0 and the identity covariance matrix. Obviously this is not a linearly separable problem. In order to reduce the randomness of the choice of data points, we report the experimental results over 30 trials. For each trial, we sample 100 data points from each class and thus there are 200 samples. We randomly select 50 % of the data points we use for training and use the rest for testing. Note that the parameters of each method are determined by the five-fold cross-validation on the training set. Figure 1

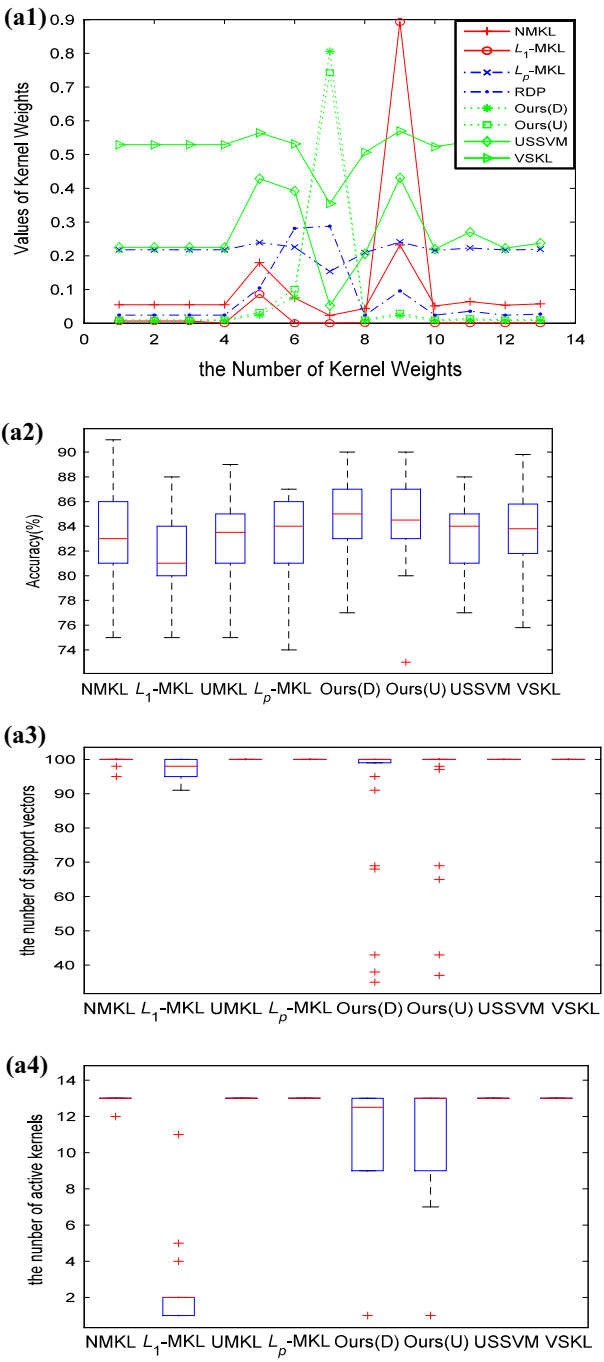

Fig. 1 Experimental results on simulated data. **a1** The values of kernel weights of each method. **a2** The average classification accuracy of each method. **a3** The average number of support vectors of each method. **a4** The average number of active kernels of each method

(a1) shows the boxplot of the value of kernel weights of different algorithms, (a2) shows the average classification accuracy of different algorithms on the testing set, (a3) shows that the average number of support vectors of different algorithms, and (a4) shows the average

number of active kernels of different algorithms. Considering the fact that the kernel weights of MKL may take relatively small values, we set the kernel weights whose values are greater than 10^{-10} as active kernels.

(1) From Fig. 1(a1), one can observe from RDP computed by Eq. (17) that the 7th kernel has the strongest discriminant power. Note that the 7th optimal kernel weight of Ours(D) is slightly greater than that of Ours(U). It is clear that the 7th optimal kernel weight obtained by our model takes the largest value even if we set the initial kernel weights to be equal. This shows that the assumption that a large weight is assigned to the kernel with strong discriminant power is reasonable. However, for other MKL methods, we find that the 7th optimal kernel weight does not take the largest value, which shows that the mechanism of selecting the optimal kernels in our model is different from that of selecting the optimal kernels in previous MKL methods.
(2) From Fig. 1(a2), one can see that Ours(D) is slightly better than Ours(U) and the performance of our method is superior to that of other MKL methods.
(3) From Fig. 1(a3), it is found that the number of support vectors of our model is smaller than that of support vectors of other MKL methods. This is possibly because our models assign large weights to those kernels with good separability. It is also noted that the number of support vectors of L_2-MKL, USSVM, VSKL, and UMKL is almost the same.
(4) From Fig. 1(a4), one can see that L_1-MKL has the least number of active kernels among all the methods and our model obtains sparse kernel weights in some trials.

Overall, the experimental results show that our model can obtain good performance in terms of classification errors and the number of support vectors although it cannot obtain the sparsest kernels on this data set.

4.2 The Experiments on the Benchmark Data Sets

In this subsection, we continue to perform some experiments on some benchmark data sets such as Australian(690 samples/14 attributes/2 classes), Breast(683/9/2), Diabetes(768/8/2), German(1000/24/2), Ionosphere(351/34/2), Heat(270/13/2), and Sonar(178/60/2) from the UCI repository [28] to evaluate the performance of different MKL methods. These data sets have been widely used in testing and evaluating the performance of some machine learning algorithms.

In order to measure the performance of each method, the five-fold cross-validation is performed and the average classification accuracy of the five-fold cross-validation is reported. For each train-test pair, another five-fold cross-validation is performed on the training set to choose the parameters of each method. Table 1 shows the average classification accuracy of each method and its standard deviation in the parenthesis on the testing set. The bold numbers denote the best performance of MKL methods on each data set. Table 2 shows the average number of support vectors of each method and its standard deviation in the parenthesis. The bold numbers denote the fewest support vectors of MKL methods on each data set. Table 3 shows the average number of active kernels of each method and its standard deviation in the parenthesis. The bold numbers denote the fewest active kernels of MKL methods on each data set. Note that the average ranks (AR) from the Friedman test [29] provide a relatively fair comparison of different algorithms on multiple data sets. So we compute the average ranks of all the algorithms in Tables 1 and 2, which are shown in the last line of each table.

(1) From Table 1, one can see that L_p-MKL obtains the best performance on the Diabetes and Ionosphere data sets and VSKL is superior to other methods on the Breast data set. Note that USSVM does not perform well on these data sets. This may be because we

Table 1 The average classification accuracy of each method and its standard deviation in the parenthesis on seven UCI data sets

Methods	L_1-MKL	L_p-MKL	NMKL	UMKL	USSVM	VSKL	Ours(U)	Ours(D)
Australian	85.49(3.69)	86.07(3.47)	78.59(16.0)	85.64(3.69)	85.49(3.92)	85.64(3.96)	**86.08(3.31)**	**86.08(2.80)**
Breast	96.63(1.60)	96.63(1.60)	96.92(1.66)	96.63(1.60)	96.48(2.09)	**97.06**(1.86)	96.63(1.32)	96.92(1.09)
Diabetes	77.07(3.26)	**77.33**(3.34)	76.81(2.86)	76.94(2.89)	74.60(2.48)	75.90(3.14)	76.42(4.04)	76.42(4.01)
German	73.60(2.07)	74.20(1.03)	74.60(1.51)	74.10(1.51)	72.90(3.83)	74.10(1.63)	75.40(4.68)	**76.30**(4.17)
Ionosphere	95.43(1.87)	**95.72**(2.02)	**95.72**(2.02)	95.43(2.56)	94.57(3.56)	96.00(2.35)	95.43(2.56)	95.43(1.87)
Heart	82.22(5.35)	81.48(7.29)	81.11(7.79)	80.74(5.94)	80.37(6.19)	80.74(5.94)	83.33(6.14)	**83.70(6.19)**
Sonar	87.54(5.08)	87.06(4.87)	87.54(4.52)	88.49(3.87)	85.09(4.58)	88.97(3.91)	88.49(3.87)	**90.89(3.03)**
AR	5.00	3.85	4.50	4.92	7.71	3.85	3.57	2.57

Table 2 The average number of support vectors of each method and its standard deviation in the parenthesis on seven UCI data sets

Methods	L_1-MKL	L_p-MKL	NMKL	UMKL	USSVM	VSKL	Ours(U)	Ours(D)
Australian	471.8(50.4)	370.6 (29.4)	442.0(80.4)	365.2(24.3)	534.4(37.4)	360.8(4.91)	343.4(41.2)	**225.8(30.5)**
Breast	173.0(46.7)	135.2(16.7)	138.4(15.1)	123.6(3.78)	249.0(76.8)	123.8(7.49)	95.6(32.1)	**72.4(15.6)**
Diabetes	547.8(33.1)	403.2 (16.7)	472.8(86.0)	381.2(8.67)	560.9(82.6)	386.6(11.5)	353.8(37.1)	**340.4(38.8)**
German	737.0(17.3)	645.2(45.9)	669.0(74.8)	625.2(44.9)	788.6(24.9)	601.4(9.60)	**436.4(20.6)**	437.6(19.5)
Ionosphere	192.8(16.0)	182.6 (5.94)	192.0(16.9)	186.2(4.96)	248.6(45.7)	185.4(4.50)	180.0 (5.78)	**121.4(11.1)**
Heart	198.4(9.44)	176.4(12.5)	171.2(10.1)	161.6(6.80)	214.8(2.68)	159.4(6.50)	**96.2(30.8)**	96.6(30.6)
Sonar	157.2(3.11)	149.8(3.11)	154.4(4.77)	147.0(2.23)	166.4(0.54)	146.8(0.92)	144.8(2.58)	**127.2(6.76)**
AR	7.00	4.85	5.85	3.85	8.00	3.42	1.71	1.28

Table 3 The average number of active kernels of each method and its standard deviation in the parenthesis on seven UCI data sets

Methods	L_1-MKL	L_p-MKL	NMKL	UMKL	USSVM	VSKL	Ours(U)	Ours(D)
Australian	**3.8(3.56)**	13(0)	6(6.40)	13(0)	13(0)	13(0)	13(0)	8.4(4.66)
Breast	**5.8(4.96)**	13(0)	13(0)	13(0)	13(0)	13(0)	8.2(6.57)	**5.8(6.57)**
Diabetes	**4.0(0.70)**	13(0)	13(0)	13(0)	13(0)	13(0)	8.8(5.84)	7.0(5.65)
German	3.80(0.83)	13(0)	13(0)	13(0)	13(0)	13(0)	**1.0(0)**	1.4(0.54)
Ionosphere	5.8(2.94)	13(0)	13(0)	13(0)	13(0)	13(0)	13(0)	13(0)
Heart	4.4(3.71)	13(0)	13(0)	13(0)	13(0)	13(0)	**1.0(0)**	1.4(0.89)
Sonar	2.8(0.44)	13(0)	13(0)	13(0)	13(0)	13(0)	13(0)	13(0)

only use L_2-norm constraints in USSVM. It is also found from Table 1 that Ours(D) is not worse than Ours(U), which shows that introducing prior knowledge for kernels is beneficial. In addition, from the average ranks of all the algorithms, it is found that Ours(D) has the smallest average rank. Thus from the viewpoint of the classification accuracy, our method can obtain better performance than other MKL methods in most cases.

(2) From Table 2, it is interesting to note that the number of support vectors of our model is generally smaller than that of support vectors of other MKL methods, which is consistent with the conclusion on the artificial data. This further shows that our model is different from previous MKL models. In general, the number of active kernels or support vectors affects the classification speed of the learning algorithms. The fewer they are, the faster the classification speed of the learning algorithms. It is found that Ours(D) gives the least number of support vectors in 5 out of 7 data sets and Ours(U) gives the least number of support vectors in 2 out of 7 data sets. It is observed that USSVM needs to store more support vectors than other MKL methods. One also observes that on the German and Heart data sets, the number of support vectors obtained by Ours(U) and Ours(D) is almost equal. From average ranks of all algorithms, we find that our model performs the best in terms of the number of support vectors.

(3) From Table 3, in terms of the number of active kernels, we find that L_1-MKL selects fewer kernels than other MKL methods on most data sets. For UMKL, L_p-MKL, USSVM and VSKL, the number of active kernels is completely equal to that of the kernels we uses. Note that NMKL does not yield sparse kernel weights since we select the best performance by tuning its parameters. Since we only consider non-sparse combinations of different groups and does not impose L_1-norm regularization on each group, VSKL does not yield sparse kernel weights. One can also see from Table 3 that our model gives sparse kernel weights on Breast, Diabetes, German, and Heart data sets. It is noted that introducing prior knowledge of kernels also affects the number of active kernels since the number of support vectors in Ours(D) and Ours(U) is not the same. In addition, it is interesting to note that our model even obtains the fewest active kernels on the Heart and German data sets, i.e., it is much sparser than L_1-MKL in obtaining kernel weights. It shows that our model can give sparse or non-sparse kernel weights for different types of data sets. Note that sparse kernel weights are obtained by setting kernel weights whose values are not greater than 10^{-10} to be zero. In fact, from theoretical analysis, kernel weights are not truly sparse and some kernel weights may take very small values due to the application to the KL divergence.

Overall, in terms of the classification accuracies, the number of active kernels, and the number of support vectors, the experimental results from Tables 1, 2, 3 show that our model is a good trade-off among these performance indexes. In addition, if the prior knowledge of each kernel is provided or one wants to use MKL with relatively fewer support vectors, ours(D) is more preferable than other MKL methods since ours(D) can obtain good performance in the general case. If one wants to use MKL with relatively fewer kernels, L_1-norm MKL is more preferable than other MKL methods.

5 Conclusions

This paper proposes a novel MKL model for data classification, which utilizes the idea of UCPs. As a trade-off between a robust combined kernel and some combined kernel from ideal kernel weights, we adopt the KL divergence in the model and solve the optimistic counterpart of its Langrage dual of the proposed model instead of the robust counterpart. It is noted that our model generally obtains the close-form solution of kernel weights. It is also found that we can obtain the solution path from a robust combined kernel to some combined kernel from ideal kernel weights by tuning a parameter in the proposed model. Experimental results on several data sets demonstrate that our model can obtain better performance than other MKL models in some cases.

Acknowledgments This work is partially supported by the National Natural Science Foundation of P.R.China (61003169, 61303182).

Appendix: The derivation of Eq. (13)

$$\begin{aligned}
&\max_{\boldsymbol{\alpha},\boldsymbol{\theta}} f(\boldsymbol{\alpha},\boldsymbol{\theta}) := -\frac{1}{2}\sum_{k=1}^{d}\theta_k \boldsymbol{\alpha}^T diag(\mathbf{y}) K_k diag(\mathbf{y})\boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{e} - \gamma \sum_{k=1}^{d}\theta_k \log\frac{\theta_k}{q_k} \\
&\text{s.t.} \sum_{i=1}^{n}\alpha_i y_i = 0, C \geq \alpha_i \geq 0, i = 1,\ldots,n, \\
&\sum_{k=1}^{d}\theta_k = 1, \theta_k \geq 0,
\end{aligned} \tag{18}$$

From Eq. (18), for fixed $\alpha_i (i = 1,\ldots,n)$, $f(\boldsymbol{\alpha},\boldsymbol{\theta})$ is strict concave with respect to θ. Thus maximizing $f(\boldsymbol{\alpha},\boldsymbol{\theta})$ over the simplex yields a unique solution. We define the following partial Lagrangian function.

$$L(\boldsymbol{\theta}) := -\frac{1}{2}\sum_{k=1}^{d}\theta_k \boldsymbol{\alpha}^T diag(\mathbf{y}) K_k diag(\mathbf{y})\boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{e} - \gamma \sum_{k=1}^{d}\theta_k \log\frac{\theta_k}{q_k} + \lambda\left(1 - \sum_{k=1}^{d}\theta_k\right) \tag{19}$$

Setting the derivative of $L(\boldsymbol{\theta})$ with respect to θ_k to zero gives

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \theta_k} = -\frac{1}{2}\boldsymbol{\alpha}^T diag(\mathbf{y}) K_k diag(\mathbf{y})\boldsymbol{\alpha} - \gamma \log\theta_k + \gamma \log q_k - \lambda = 0. \tag{20}$$

From (20), one has

$$-\frac{1}{2}\boldsymbol{\alpha}^T diag(\mathbf{y}) K_k diag(\mathbf{y})\boldsymbol{\alpha} - \lambda = \gamma \log \frac{\theta_k}{q_k} \tag{21}$$

From Eq. (21), one has

$$q_k e^{-\frac{1}{2\gamma}\boldsymbol{\alpha}^T diag(\mathbf{y}) K_k diag(\mathbf{y})\boldsymbol{\alpha} - \frac{\lambda}{\gamma}} = \theta_k. \tag{22}$$

Note that $\sum_{k=1}^{d} \theta_k = 1$. We have

$$\sum_{k=1}^{d} q_k e^{-\frac{1}{2\gamma}\boldsymbol{\alpha}^T diag(\mathbf{y}) K_k diag(\mathbf{y})\boldsymbol{\alpha} - \frac{\lambda}{\gamma}} = \sum_{k=1}^{d} \theta_k = 1. \tag{23}$$

$$e^{-\frac{\lambda}{\gamma}} \sum_{k=1}^{d} q_k e^{-\frac{1}{2\gamma}\boldsymbol{\alpha}^T diag(\mathbf{y}) K_k diag(\mathbf{y})\boldsymbol{\alpha}} = 1. \tag{24}$$

Substituting Eq. (24) into Eq. (22), one has

$$\theta_k = \frac{q_k e^{\frac{-\sum_{i,j} \alpha_i \alpha_j y_i y_j K_k(\mathbf{x_i}, \mathbf{x_j})}{2\gamma}}}{\sum_{s=1}^{d} q_s e^{\frac{-\sum_{i,j} \alpha_i \alpha_j y_i y_j K_s(\mathbf{x}_i, \mathbf{x}_j)}{2\gamma}}}, (k = 1, \ldots, d). \tag{25}$$

From Eq. (25), one can see that the non-negativity of θ can be automatically guaranteed.

References

1. Aflalo J, Ben-Tal A, Bhattacharyya C, Nath JS, Raman S (2001) Variable sparsity kernel learning. J Mach Learn Res 12:565–592
2. Gonen M, Alpaydm E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12:2211–2268
3. Bach FR (2008) Consistency of group Lasso and multiple kernel learning. J Mach Learn Res 9:1179–1225
4. Kloft M, Brefeld U, Sonnenburg S, Zien A (2011) L_p-norm multiple kernel learning. J Mach Learn Res 12:953–997
5. Lanckriet GRG, Cristianini N, Bartlett P, Jordan MI (2004) Learning the kernel matrix with semidefinite programming. J Mach Learn Res 5:27–72
6. Bach FR, Lanckriet GRG, Jordan MJ (2004) Multiple kernel learning, conic duality and SMO algorithms. In: Proceedings of the 21st, ICML, ACM
7. Alizadeh F, Goldfarb D (2003) Second-order cone programming. Math Program Ser B 95:3–51
8. Sonnenburg S, Ratsch G, Schafer C, Scholkopf B (2006) Large scale multiple kernel learning. J Mach Learn Res 7:1531–1565
9. Szafranski M, Grandvalet Y, Rakotomamonjy A (2010) Composite kernel learning. Mach Learn 79(1–2):73–103
10. Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y (2008) Simple MKL. J Mach Learn Res 9:1125–1179
11. Xu Z, Jin R, Ye J, King I, Lyu M (2010) Simple and efficient multiple kernel learning by group Lasso. In: The 27th International Conference on Machine Learning, pp 1175–1182
12. Yang H, Xu X, Ye J, King I, Lyu M (2011) Efficient sparse generalized multiple kernel learning. IEEE Trans Neural Netw 22(3):433–446
13. Vishwanathan SVN, Sun Z, Ampornputn N, Varma M (2010) Multiple kernel learning and the SMO algorithm. In: Proceedings of NIPS, pp 2361–2369
14. Bertsekas DP (1999) Nonlinear programming, 2nd edn. Athena Scientific, Belmont
15. Liang ZZ, Xia S, Zhou Y, Zhang L (2013) Training Lp norm multiple kernel learning in the primal. Neural Netw 46:172–182
16. Xu X, Tsang I, Xu D (2013) Soft margin multiple kernel learning. IEEE Trans Neural Netw Learn 24(5):749–761
17. Xu Z, Jin R, Zhu S, Lyu M, King I (2010) Smooth optimization for effective multiple kernel learning. In: The 24th AAAI Conference on Artificial Intelligence, pp 637–642

18. Beck A, Ben-Tal A (2009) Duality in robust optimization: primal worst equals dual best. Oper Res Lett 37:1–6
19. Ben-Tal A, Ghaoui LE, Nemirovski A (2009) Robust optimization. Princeton series in applied mathematics. Princeton, Princeton University Press
20. JeyakumarV Li G (2011) Strong duality in robust convex programming: complete characterizations. SIAM J Optim 20(6):3384–3407
21. Li G, Jeyakumar V, Lee GM (2011) Robust conjugate duality for convex optimization under uncertainty with applications to data classification. Nonlinear Anal Theory Methods Appl 74(6):2327–2341
22. Ben-Tal A, Bhadra S, Bhattacharyya C, Nemirovski A (2012) Efficient methods for robust classification under uncertainty in kernel matrices. J Mach Learn Res 13:2923–2954
23. Jeyakumar V, Li G (2012) Support vector machine classifier with uncertain knowledge sets via robust optimization. Optimization, pp 1–18
24. Goyal V, Ravi R (2013) An FPTAS for minimizing a class of quasi-concave functions over a convex set. Oper Res Lett 41(2):191–196
25. Duchi J, Shalev-Shwartz S, Singer Y, Chandra T (2008) Efficient projections onto the l1-ball for learning in high dimensions. In: ICML, pp 272–279
26. http://www.csie.ntu.edu.tw/cjlin/libsvm/. Accessed 10 April 2013
27. http://www.mosek.com/. Accessed 20 June 2013
28. Blake C, Merz C (1998) UCI repository of machine learning databases. http://archive.ics.uci.edu/ml. Accessed 9 July 2009
29. Nemsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–26