Efficient Sparse Generalized Multiple Kernel Learning

Haiqin Yang, *Student Member, IEEE*, Zenglin Xu, *Member, IEEE*, Jieping Ye, Irwin King, *Senior Member, IEEE*, and Michael R. Lyu, *Fellow, IEEE*

***Abstract*— Kernel methods have been successfully applied in various applications. To succeed in these applications, it is crucial to learn a good kernel representation, whose objective is to reveal the data similarity precisely. In this paper, we address the problem of multiple kernel learning (MKL), searching for the optimal kernel combination weights through maximizing a generalized performance measure. Most MKL methods employ the L_1-norm simplex constraints on the kernel combination weights, which therefore involve a sparse but non-smooth solution for the kernel weights. Despite the success of their efficiency, they tend to discard informative complementary or orthogonal base kernels and yield degenerated generalization performance. Alternatively, imposing the L_p-norm ($p > 1$) constraint on the kernel weights will keep all the information in the base kernels. This leads to non-sparse solutions and brings the risk of being sensitive to noise and incorporating redundant information. To tackle these problems, we propose a generalized MKL (GMKL) model by introducing an elastic-net-type constraint on the kernel weights. More specifically, it is an MKL model with a constraint on a linear combination of the L_1-norm and the squared L_2-norm on the kernel weights to seek the optimal kernel combination weights. Therefore, previous MKL problems based on the L_1-norm or the L_2-norm constraints can be regarded as special cases. Furthermore, our GMKL enjoys the favorable sparsity property on the solution and also facilitates the grouping effect. Moreover, the optimization of our GMKL is a convex optimization problem, where a local solution is the global optimal solution. We further derive a level method to efficiently solve the optimization problem. A series of experiments on both synthetic and real-world datasets have been conducted to show the effectiveness and efficiency of our GMKL.**

***Index Terms*— Grouping effect, kernel methods, level method, multiple kernel learning.**

I. Introduction

Kernel methods such as support vector machines (SVMs), kernel principal component analysis, etc. [1]–[4] have become useful tools in various applications including, e.g., pattern recognition [3], [4] and bioinformatics [5], [6]. To achieve good performance, one has to define a good kernel representation. The kernel matrix is specified by the inner product of data points mapped in a high-dimensional (possibly infinite dimensional) feature space. The kernel matrix defines the similarity among data and usually has to be learned from the data.

The problem of learning the optimal kernel matrix has received much attention in recent studies of machine learning [3], [7]–[11]. One of the important kernel learning techniques is multiple kernel learning (MKL), which was first introduced in [12]. In general, MKL searches for a linear combination of base kernel functions/matrices that maximizes a generalized performance measure. Typical measures for MKL include maximum margin classification errors [12], [13], kernel–target alignment [14], and Fisher discriminative analysis [15]. MKL methods have been shown to be usually outperformed by SVM with uniformly weighted kernels [12], [16]–[18].

Among various MKL methods, the L_1-MKL has shown its efficiency in learning the kernel weights. This method seeks the kernel weights in a simplex and thus yields a sparse solution. The sparsity of the selected kernels is helpful in identifying the appropriate combination of data sources or different feature subsets in real-world applications, such as genome fusion [5], splice site detection [19], image annotation [20], etc. However, when a problem contains kernels encoding orthogonal or correlation characterizations, the simplex solution space may discard useful information and thus result in suboptimal generalization performance [18]. Alternatively, an MKL with the L_2-norm constraint on the kernel weights is proposed [17] and an MKL with the L_p-norm ($p > 1$) constraint on the kernel weights is further presented [18] to improve the L_1-MKL method. Unfortunately, these extensions lead to a non-sparse solution and may be sensitive to noise. They suffer poor interpretation ability and subsequently can lead to high computational and storage cost.

To avoid problems of the above two types of approaches, it is strongly desirable to keep the locally orthogonal information in the base kernels [16], [18], while at the same time, to yield a sparse solution. Clearly, one approach toward this objective is first to cluster the kernel matrices/functions into groups and then to identify the leading groups. In this way, the complementary or locally orthogonal information can be kept and sparse solutions can also be obtained. Similar methods, e.g., group lasso [21], fussed lasso [22], etc., have been introduced in statistics. Group lasso aims to find important

Manuscript received December 3, 2009; revised November 22, 2010; accepted December 23, 2010. Date of publication January 20, 2011; date of current version March 2, 2011. This work was substantially supported by two grants from the Research Grants Council of the Hong Kong SAR, China (Project CUHK4128/08E and Project CUHK4154/10E), and by funding from Google Focused Grant Project "Mobile 2014."

H. Yang, I. King, and M. R. Lyu are with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong (e-mail: hqyang@cse.cuhk.edu.hk; king@cse.cuhk.edu.hk; lyu@cse.cuhk.edu.hk).

Z. Xu was with Cluster of Excellence MMCI, Saarland University, Saarbrucken 66123, Germany, and the Max Planck Institute for Informatics, Saarbrucken 66123, Germany. He is now with the Department of Computer Science, Purdue University, West Lafayette, IN 47907 USA (e-mail: zlxu@mpi-inf.mpg.de).

J. Ye is with the Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: jieping.ye@asu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2010.2103571

1045–9227/$26.00 © 2011 IEEE

explanatory factors in predicting the response variable, where each explanatory factor can be represented by a group of derived input variables. In [23], Bach has shown that group lasso reduces to MKL when the Euclidean norms in group lasso are replaced by reproducing kernel Hilbert norms. The composite kernel learning [24] is an example of kernel learning approach based on the group lasso, where the kernels are hierarchically penalized. Despite their success, the group composition must be specified ahead as a prior knowledge. However, in some real-world problems, the prior knowledge on the composition of the group structure may not be available before learning. Moreover, the group penalization often involves high computation cost due to the projection to the hierarchical structure of the kernel weights.

To tackle the above problems, we propose a novel generalized multiple kernel learning (GMKL) model. Our model introduces the regularization with a linear combination of the L_1-norm and the squared L_2-norm on the kernel weights, i.e., a combination of lasso and ridge penalties on the kernel weights. This model generalizes the L_1-MKL and the L_2-MKL methods. More importantly, our GMKL not only enjoys sparse solution as the L_1-MKL but also encourages the grouping effect on the solution, where similar base kernels tend to be either in or out of the model altogether without specifying the group information in advance. Therefore, this demonstrates distinct advantages over the L_1-MKL [12], [25] or the L_2-MKL [17]. Furthermore, compared to group lasso-based approaches, the proposed approach relaxes the needs for the prior knowledge of the group structure of base kernels.

In summary, the contributions of this paper include the following.

1) A novel GMKL model is introduced that generalizes several previously proposed MKL models, including the L_1-MKL and the L_2-MKL. The proposed GMKL model can overcome the insufficiency of the L_1-MKL and the L_2-MKL.
2) Theoretical analysis of the GMKL on why it contains a sparse solution with the grouping effect is provided. This guarantees the favorite properties of the GMKL.
3) The GMKL is transformed into a convex-concave optimization problem. So the global optimal solution is guaranteed. An efficient method, i.e., the level method, is proposed to solve the GMKL and its convergence rate is provided. This solution enables the GMKL for its potential on solving large-scale datasets.
4) A series of experiments have been conducted both on synthetic and real-world datasets to demonstrate the effectiveness and efficiency of the GMKL.

The rest of this paper is organized as follows. In Section III, we outline the MKL framework and introduce the current research progress on extending this framework. In Section IV, we describe our proposed GMKL model and provide theoretical analysis of its properties. In Section V, we present the solution of the GMKL by the level method and provide its convergence analysis. In Section VI, we report the experimental results on both synthetic and real-world datasets. Finally, we conclude this paper in Section VII.

II. Notation

We first introduce some notations here. Bold capital letters, e.g., $\mathbf{K}$, indicate matrices. Bold small letters, e.g., $\mathbf{w}$ and $\boldsymbol{\alpha}$, indicate vectors. $\mathbf{1}_m$ ($\mathbf{0}_m$) is an m-dimensional vector with each element being 1 (0). Letters in calligraphic or blackboard bold fonts, e.g., $\mathcal{X}$, $\mathbb{R}$, indicate a set, where $\mathbb{R}^n$ denotes an n-dimensional real space. $\mathbf{z} \in \mathbb{R}^n_+$ means $\mathbf{z}$ is an n-dimensional vector with $z_i \geq 0$ for $i = 1, \ldots, n$. The operator $^\top$ denotes the transpose and $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{H}}$ defines the inner product of $\mathbf{x}$ and $\mathbf{y}$ in the space $\mathcal{H}$. $d(\mathcal{H})$ defines the dimension of the space $\mathcal{H}$. $\mathbf{X} \succeq \mathbf{0}$ denotes a matrix which is positive semidefinite. The operator $\circ$ defines the Hadamard or elementwise product.

III. MKL

In this section, we first introduce the basic concept of kernel methods. We then present the framework of MKL. Finally, the L_1-MKL and its extensions are further discussed.

A. Preliminaries

In supervised learning, a set of labeled data $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ is given, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ for some input space $\mathcal{X}$, and $y_i \in \mathcal{Y}$. For binary classification, $\mathcal{Y} = \{\pm 1\}$. For regression problems, $\mathcal{Y} = \mathbb{R}$. The objective of supervised learning is to find a hypothesis $f \in \mathcal{H}$ that can generalize well on unseen data. This is attained by minimizing the following regularized risk:

$$f^\star = \arg\min_f \; C\,\mathrm{R_{emp}}(f) + \Omega(f) \tag{1}$$

where $\mathrm{R_{emp}}(f) = 1/N \sum_{i=1}^N R(f(\mathbf{x}_i), y_i)$ is the empirical risk of hypothesis f with respect to a loss function, $R : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$, and $\Omega(f)$ is a regularization term. The positive constant term C is a tradeoff parameter balancing the regularization and the empirical risk.

For different problems, different (usually convex) loss functions $R(f(\mathbf{x}), y)$ are adopted. Typical examples include the hinge loss for classification in SVMs and ε-insensitive loss function for support vector regression [26].

In this paper, similar to previous kernel methods [3], the regularizer $\Omega(f)$ is $1/2\|\mathbf{w}\|_2^2$, corresponding to the squared L_2-norm on the function weights and the function f takes a linear form with parameters $\mathbf{w}$ and b as

$$f_{\mathbf{w},b}(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b, \quad \mathbf{w} \in \mathbb{R}^{d(\mathcal{H})}, \; b \in \mathbb{R} \tag{2}$$

where $\phi : \mathcal{X} \to \mathcal{H}$ defines a (possible nonlinear) feature mapping from the original input space to a Hilbert space $\mathcal{H}$. The feature mapping is usually implicitly defined by a Mercer kernel computing the inner product in $\mathcal{H}$ as $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}}$ [3].

The decision function can then be represented by

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i^\star \mathbf{K}(\mathbf{x}, \mathbf{x}_i) + b^\star \tag{3}$$

where the optimal parameter $\boldsymbol{\alpha}^\star$ and $b^\star$ are obtained by solving the dual of the optimization in (1).

B. MKL Framework

In the MKL framework, there are given Q base kernels. Each base kernel $\mathbf{K}_q$ implicitly represents a feature mapping, $\phi_q : \mathcal{X} \rightarrow \mathcal{H}_q$, in a reproducing kernel Hilbert space $\mathcal{H}_q$, for $q = 1, \ldots, Q$. The hypothesis in (2) is then extended to

$$f_{\hat{\mathbf{w}},b,\boldsymbol{\theta}}(\mathbf{x}) = \hat{\mathbf{w}}^\top \boldsymbol{\phi}_{\boldsymbol{\theta}}(\mathbf{x}) + b = \sum_{q=1}^{Q} \sqrt{\theta_q}\mathbf{w}_q^\top \phi_q(\mathbf{x}) + b \tag{4}$$

where the weight $\hat{\mathbf{w}}$ is defined as $\hat{\mathbf{w}} = (\mathbf{w}_1^\top, \ldots, \mathbf{w}_Q^\top)^\top$, consisting of a $\sum_{q=1}^{Q} d(\mathcal{H}_q)$-dimensional vector. The composite feature mapping is defined as $\boldsymbol{\phi}_{\boldsymbol{\theta}} = \sqrt{\theta_1}\phi_1 \times \cdots \times \sqrt{\theta_Q}\phi_Q$, and θ_q is the corresponding coefficient, or the kernel weights of the kernel $\mathbf{K}_q$, and needs to be learned from the data.

The objective of MKL is to seek the optimal kernel combination $\mathbf{K}_{\boldsymbol{\theta}} = \sum_{q=1}^{Q} \theta_q \mathbf{K}_q$ by minimizing the following optimization while imposing the Ivanov regularization on the kernel weights [12], [18]:

$$\min_{\hat{\mathbf{w}},b,\boldsymbol{\theta}\geq\mathbf{0}} \quad C\sum_{i=1}^{N} R(f_{\hat{\mathbf{w}},b,\boldsymbol{\theta}}(\mathbf{x}_i), y_i) + \frac{1}{2}\hat{\mathbf{w}}^\top\hat{\mathbf{w}} \tag{5}$$

$$\text{s.t.} \quad \mathcal{J}(\boldsymbol{\theta}) \leq 1 \tag{6}$$

where $\mathcal{J}(\boldsymbol{\theta})$ defines a regularizer on $\boldsymbol{\theta}$, which will be elaborated in the following subsections. It is noted that an MKL framework that seeks optimal kernels in a compact set by minimizing a regularized functional was also studied in [27] from a theoretical perspective. Reference [27] mainly studied the theoretical properties on the square loss with L_1-norm regularization on the functional. This is different from what we will propose in the next section.

In addition, we should note that the non-convexity of (5) can be resolved by applying the variable transformation $\mathbf{v}_q := \sqrt{\theta_q}\mathbf{w}_q$, as that in [18] and [28]. Hence, the objective in (5) becomes

$$\min_{\hat{\mathbf{v}},b,\boldsymbol{\theta}\geq\mathbf{0}} \quad C\sum_{i=1}^{N} R(f_{\hat{\mathbf{v}},b}(\mathbf{x}_i), y_i) + \frac{1}{2}\sum_{q=1}^{Q} \frac{\mathbf{v}_q^\top\mathbf{v}_q}{\theta_q} \tag{7}$$

where $\hat{\mathbf{v}} = (\mathbf{v}_1^\top, \ldots, \mathbf{v}_Q^\top)^\top$ and $f_{\hat{\mathbf{v}},b}(\mathbf{x}) = \sum_{q=1}^{Q} \mathbf{v}_q^\top \phi_q(\mathbf{x}) + b$. In (7), we use the convention that $(u/0) = 0$ if $u = 0$ and ∞ otherwise. If R is a convex function and the constraint (6) is convex, then (7) is convex. This result can be referred to [29, Ch. 3.1.5].

C. L_1-MKL

Common approaches in MKL [12], [13], [25] and [30] impose the L_1-norm constraint on the kernel weights for the kernel selection. That is, $\mathcal{J}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1$, or $\|\boldsymbol{\theta}\|_1 \leq 1$ in the condition of (6). We refer to this case as the L_1-MKL.

In [12], the L_1-MKL is first formulated into a semidefinite programming problem. Due to its effectiveness in learning an interpretable kernel representation, researchers have proposed various methods to speed up its computation. Methods such as second order cone programming [13], semi-infinite linear programming [30], gradient descent [25], and the extended level method [31] have been proposed to reduce the time consumption in seeking the optimal kernel combination weights.

An advantage of the L_1-MKL constraint on the kernel combination weights is that it provides the favorite property of sparsity, where the obtained kernels can be easily interpreted. However, it may also discard some useful information when two kernels are orthogonal [16] or yield non-unique solutions when two kernels are strongly correlated.

D. MKL Extensions

In order to tackle the deficiency of the L_1-MKL, researchers have extended the MKL models. They include the following.

1) The MKL model with the L_2-norm constraint on the kernel weights [17], i.e., $\mathcal{J}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2$, or $\|\boldsymbol{\theta}\|_2^2 \leq 1$ for the condition (6). Similarly, a multiple kernel ridge regression is proposed in [16], where the kernel weights is constrained in a ball around a positive mean.
2) The MKL with the L_p-norm $(p > 1)$ constraint on the kernel weights [18], [32]. This corresponds to $\mathcal{J}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_p^p$, or $\|\boldsymbol{\theta}\|_p^p \leq 1$ in (6). The L_p-MKL is more general and includes the L_2-MKL as its special case. An interleaved optimization strategy with second order approximation is proposed to solved the L_p-MKL [18].
3) The MKL model with mixed norm regularization on the kernel weights [33]. This model imposes a mixed norm regularization on the kernel weights, which yields structure sparsity on the solutions.
4) Other MKL extensions. These models reformulate the MKL problem by imposing mixed norm regularization on the function weights [34], or by introducing the elastic net-type regularization, i.e., a linear combination of the lasso penalty and the ridge penalty on the function weights [35]. These formulations correspond to modifying the regularizer to a block norm, i.e., a norm of the vector containing the individual kernel norms [13], [36]. Now, we discuss several MKL methods incorporating the elastic net-type regularization that may be similar to that in this paper. Longworth and Gales [37] included the squared L_2-norm regularization on the kernel weights while keeping the L_1-norm simplex constraint on the kernel weights. Shawe-Taylor [38] proposed a linear combination of the square of the sum of L_1-norms and the squared L_2-norm on the function weights to solve the novelty detection problem. In [35], an MKL model added the linear combination of the lasso penalty and the ridge penalty on the function weights, which includes the L_1-MKL and the uniformly weighted MKL (UW-MKL) as its special cases. However, they lack the analysis on the properties of the models, e.g., the grouping effect.

Among the above methods, the L_1-MKL yields a sparse solution, but cannot capture the complementary information on the kernels. For the L_p-MKL $(p > 1)$ models, they will yield non-sparse solutions. As indicated in [39], in the L_p $(p \geq 1)$ penalty family, only the lasso penalty $(p = 1)$ can produce sparse solutions. The non-sparsity of the solution has the weaknesses in interpreting the model and may be sensitive to the noise. In the following section, we will present our proposed GMKL model to tackle the above insufficiency problem of previously proposed MKL models.

IV. GMKL

In this section, we first introduce the formulation of the GMKL model. We then provide theoretical analysis on the properties of the GMKL model, i.e., explaining why it can produce sparse solutions while encouraging the grouping effect.

A. Formulation and Duality

Motivated by the fact that the L_1-MKL produces sparse solutions and the L_p-MKL ($p > 1$) can capture correlations among kernels, we propose a GMKL model incorporating a linear combined norm on the kernel weights as follows:

$$\min_{\boldsymbol{\theta}\in\Theta,\hat{\mathbf{v}},b} \quad C\sum_{i=1}^{N} R(f_{\hat{\mathbf{v}},b}(\mathbf{x}_i), y_i) + \frac{1}{2}\sum_{q=1}^{Q}\frac{\mathbf{v}_q^\top\mathbf{v}_q}{\theta_q} \tag{8}$$

where, more specially, we set $p = 2$, and the domain of $\boldsymbol{\theta}$ is

$$\Theta = \{\boldsymbol{\theta} \in \mathbb{R}_+^Q : \upsilon\|\boldsymbol{\theta}\|_1 + (1-\upsilon)\|\boldsymbol{\theta}\|_2^2 \leq 1\} \tag{9}$$

where the parameter υ, $0 \leq \upsilon \leq 1$, is a nonnegative constant to balance the two terms in the constraint. For this MKL extension, we have several remarks.

1) There are two main reasons that we adopt this elastic net-type regularization, i.e., a linear combination of the L_1-norm and the squared L_2-norm on the kernel weights. One is due to computational consideration. Through this setting, the optimization in (8) is a convex optimization problem given that the loss R is convex. More specifically, it is a quadratically constrained quadratic programming problem when the hinge loss or the ε-insensitive loss is used. The second reason is that, as discussed in Section IV-B, our GMKL enjoys the sparsity property as the L_1-MKL and encourages the grouping effect on the kernel weights similar to that of the elastic net on the model weights.
2) Our formulation generalizes previously proposed L_1-MKL and the L_2-MKL models. When $\upsilon = 0$, the constraint reduces to a ridge penalty on $\boldsymbol{\theta}$ and the model is equivalent to the L_2-MKL [17]. When $\upsilon = 1$, the constraint is a lasso constraint and the model is the L_1-MKL [12]. This motivates us to name our model as GMKL. When $\upsilon \in (0, 1)$, the constraint contains the characteristics of both the lasso and ridge penalty and the model includes several favorite properties, which will be introduced in Section IV-B.
3) The constraint can be further extended by combining the L_1-norm and the L_p-norm ($p > 1$) on the kernel weights and therefore generalizes previously proposed related MKL methods [18], [32]. When $\upsilon \in (0, 1)$, the extended constraint is strictly convex on $\boldsymbol{\theta}$ and contains similar properties of our GMKL formulation, see Section IV-B for more details.
4) Our proposed GMKL also generalizes the L_2-norm regularization proposed in [16]. A main difference is that the L_2-norm regularization in [16] introduces an L_2-ball with a predefined positive ball center. Actually, predefining a ball center is not necessary in practical applications and is not required in our model. Furthermore, the formulation in [16] lacks the properties of sparsity and the grouping effect.

Now, we derive the dual form of the optimization in (8) with respect to $\hat{\mathbf{w}}$, b by fixing $\boldsymbol{\theta}$. Here, we consider the classification problem where the hinge loss is adopted. Hence, the primal problem of GMKL is equivalent to

$$\begin{aligned}\min_{\boldsymbol{\theta}\in\Theta} \ \min_{\hat{\mathbf{v}},b,\boldsymbol{\xi}} \quad & C\sum_{i=1}^{N}\xi_i + \frac{1}{2}\sum_{q=1}^{Q}\frac{\mathbf{v}_q^\top\mathbf{v}_q}{\theta_q} \\ \text{s.t.} \quad & y_i\left(\sum_{q=1}^{Q}\mathbf{v}_q^\top\phi_q(\mathbf{x}_i) + b\right) \geq 1 - \xi_i, \xi_i \geq 0.\end{aligned}$$

Following the standard Lagrange multipliers method [26], [40], we construct the corresponding Lagrangian functional $\mathcal{L}(\hat{\mathbf{v}}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\gamma})$ of the minimization on the primal variables with fixed $\boldsymbol{\theta}$ as

$$\begin{aligned}\mathcal{L}(\cdot) = {} & C\sum_{i=1}^{N}\xi_i + \frac{1}{2}\sum_{q=1}^{Q}\frac{\mathbf{v}_q^\top\mathbf{v}_q}{\theta_q} - \sum_{i=1}^{N}\gamma_i\xi_i \\ & -\sum_{i=1}^{N}\alpha_i\left(y_i\left(\sum_{q=1}^{Q}\mathbf{v}_q^\top\phi_q(\mathbf{x}_i) + b\right) - 1 + \xi_i\right)\end{aligned} \tag{10}$$

where the multipliers satisfy $\boldsymbol{\alpha} \geq \mathbf{0}$ and $\boldsymbol{\gamma} \geq \mathbf{0}$.

Taking the partial derivative of the Lagrangian function with respect to the corresponding primal variables and setting them to zeros, we obtain

$$\frac{\partial\mathcal{L}}{\partial\mathbf{v}_q} = \mathbf{v}_q - \theta_q\sum_{i=1}^{N}\alpha_i y_i\phi_q(\mathbf{x}_i) = 0, \ q = 1, \ldots, Q \tag{11}$$

$$\frac{\partial\mathcal{L}}{\partial b} = \sum_{i=1}^{N}\alpha_i y_i = 0 \tag{12}$$

$$\frac{\partial\mathcal{L}}{\partial\xi_i} = C - \alpha_i - \gamma_i = 0, \ i = 1, \ldots, N. \tag{13}$$

From (11), we can obtain the dual form of (8) as follows:

$$\min_{\boldsymbol{\theta}\in\Theta} \ \max_{\boldsymbol{\alpha}\in\mathcal{A}} \ \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}) \tag{14}$$

where the objective function is defined as

$$\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \mathbf{1}_N^\top\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})^\top\left(\sum_{q=1}^{Q}\theta_q\mathbf{K}_q\right)(\boldsymbol{\alpha}\circ\mathbf{y}). \tag{15}$$

Correspondingly, constraints (12) and (13) with the conditions of $\boldsymbol{\alpha} \geq \mathbf{0}$ and $\boldsymbol{\gamma} \geq \mathbf{0}$ yield the domain of $\boldsymbol{\alpha}$ defined in the set of $\mathcal{A}$ as

$$\mathcal{A} = \{\boldsymbol{\alpha} \in \mathbb{R}_+^N, \ \boldsymbol{\alpha}^\top\mathbf{y} = 0, \ \boldsymbol{\alpha} \leq C\mathbf{1}_N\}. \tag{16}$$

The formulation in (14) is a convex-concave problem and its optimal solution is guaranteed to be the global optimal solution. Wrapping-based methods [25], [30] have been proposed to solve this kind of optimization problems. Especially, the maximization problem in (14) corresponds to a standard dual form of SVMs. Currently, solvers for SVMs are very efficient [41], [42] and can be directly adopted in our model.

B. Properties

Here, we present several properties for our GMKL model. First, we prove that the constraint on $\boldsymbol{\theta}$ is tight when the optimal solution is obtained. Second, we provide a theorem to show that the solution of the GMKL is sparse with the grouping effect. Third, we prove that our GMKL model introduces the grouping effect when two kernels are strongly correlated.

To simplify the analysis, we first define the optimal $(\boldsymbol{\theta}^\star, \boldsymbol{\alpha}^\star)$ as follows.

Definition 1: $(\boldsymbol{\theta}^\star, \boldsymbol{\alpha}^\star)$ is the optimal solution of (14). That is

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}\in\Theta} \ \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}^\star) \quad \boldsymbol{\alpha}^\star = \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \ \mathcal{D}(\boldsymbol{\theta}^\star, \boldsymbol{\alpha}).$$

Now, the following theorem shows that the optimal solution in (14) attains when the constraint of $\boldsymbol{\theta}$ is tight.

Theorem 1: Suppose the kernel matrices $\mathbf{K}_1, \ldots, \mathbf{K}_Q$ are positive semidefinite. Then the condition $\upsilon\|\boldsymbol{\theta}^\star\|_1 + (1 - \upsilon)\|\boldsymbol{\theta}^\star\|_2^2 = 1$ always holds.

Proof: First, we have the following two observations about the function $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}^\star)$.

1) The function $\mathcal{D}(\theta_q, \boldsymbol{\alpha}^\star)$ is a monotonically but not strictly decreasing function on θ_q, with fixed $\boldsymbol{\alpha}^\star$. It is because $\mathcal{D}(\theta_q, \boldsymbol{\alpha}^\star)$ is a linear function on θ_q, with each coefficient being nonpositive. That is, $\mathcal{D}(\theta_q, \boldsymbol{\alpha}^\star) = u_q\theta_q$, where the qth coefficient on θ_q is $u_q = -(1/2)(\boldsymbol{\alpha}^\star \circ \mathbf{y})^\top \mathbf{K}_q(\boldsymbol{\alpha}^\star \circ \mathbf{y})$. Obviously, u_q is nonpositive since the kernel matrix $\mathbf{K}_q$ is positive semidefinite.
2) The constraint function $\upsilon\|\boldsymbol{\theta}\|_1 + (1-\upsilon)\|\boldsymbol{\theta}\|_2^2$ is elementwise and it is an increasing function on each element of $\boldsymbol{\theta}$, or θ_q, with $\theta_q \geq 0$, for $q = 1, \ldots, Q$.

Hence, we can conclude that the optimal $\boldsymbol{\theta}^\star$ should be attained when the constraint of (9) is tight. Otherwise, we have the following two cases.

1) If there is an element, e.g., q, with $u_q < 0$, we can select θ_q and increase its value to make (9) tight. This again will further reduce the function value of $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}^\star)$, which is a better solution of (14), from the above first observation.
2) For all q, $u_q = 0$, we can select any θ_q and increase its value to make (9) tight, while keeping the same optimal objective function value. ■

We now turn to study the grouping effect of our GMKL. First, we note that the grouping effect is only derived from $\boldsymbol{\theta}$ and it is not related to the variable $\boldsymbol{\alpha}$. By the Lagrange multiplier method [29], we know that (14) is equivalent to the following minimization problem given the fixed $\boldsymbol{\alpha}^\star$ for some $\lambda \geq 0$:

$$\min_{\boldsymbol{\theta}\geq\mathbf{0}} \quad \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}^\star) + \lambda\left(\upsilon\|\boldsymbol{\theta}\|_1 + (1-\upsilon)\|\boldsymbol{\theta}\|_2^2\right). \tag{17}$$

We then have the following theorem stating one aspect of the grouping effect.

Theorem 2: Suppose $\lambda > 0$, $\mathbf{K}_i = \mathbf{K}_j$, $i, j \in \{1, \ldots, Q\}$, and $\boldsymbol{\theta}^\star$ is a minimizer of (17); then we have:

1) if $\upsilon \neq 1$, then
$$\theta_q^\star = \max\left\{0, \frac{1}{2(1-\upsilon)}\left(\frac{1}{2\lambda}(\boldsymbol{\alpha}^\star \circ \mathbf{y})^\top \mathbf{K}_q(\boldsymbol{\alpha}^\star \circ \mathbf{y}) - \upsilon\right)\right\} \tag{18}$$
and therefore $\theta_i^\star = \theta_j^\star$;
2) if $\upsilon = 1$, then $\tilde{\boldsymbol{\theta}}$ is another minimizer of (17) with
$$\tilde{\theta}_q = \begin{cases} \theta_q^\star & \text{if } q \neq i \text{ and } q \neq j \\ (\theta_i^\star + \theta_j^\star)\cdot\sigma & \text{if } q = i \\ (\theta_i^\star + \theta_j^\star)\cdot(1-\sigma) & \text{if } q = j \end{cases}$$
for any $\sigma \in [0, 1]$.

A detailed proof is given in Appendix I. There are some remarks about the above theorem.

1) Theorem 2 provides an explicit solution of $\boldsymbol{\theta}$ in (18). This is different from that of the elastic net in [39, Lemma 2].
2) Theorem 2 indicates that our GMKL can achieve the grouping effect and that the L_1-MKL does not have a unique solution when two kernels are the same. This analysis can be also extended to other regularizers with strictly convex property.
3) Equation (18) also indicates that our GMKL can yield sparse solutions when the second term in the bracket of (18) is less than 0. On the contrary, by setting $\upsilon = 0$ into (18), we can obtain $\theta_q^\star = 1/(2(1-\upsilon))\left((1/2)\lambda(\boldsymbol{\alpha}^\star \circ \mathbf{y})^\top \mathbf{K}_q(\boldsymbol{\alpha}^\star \circ \mathbf{y})\right)$ for the L_2-MKL. This also shows that the L_2-MKL yields the grouping effect, but usually yields non-sparse solutions on the kernel weights.

We further analyze the grouping effect when the given kernels are strongly correlated. Here, we define a ratio for two kernels to indicate the correlation of two kernels.

Definition 2: Let r_{ij} define the ratio of two kernels on given $\boldsymbol{\alpha}^\star$ as

$$r_{ij} = \frac{(\boldsymbol{\alpha}^\star \circ \mathbf{y})^\top \mathbf{K}_i(\boldsymbol{\alpha}^\star \circ \mathbf{y})}{(\boldsymbol{\alpha}^\star \circ \mathbf{y})^\top \mathbf{K}_j(\boldsymbol{\alpha}^\star \circ \mathbf{y})}.$$

If $r_{ij} \approx 1$, we say $\mathbf{K}_i$ and $\mathbf{K}_j$ are strongly correlated. Now, we can easily obtain the following theorem.

Theorem 3: Given two kernels $\mathbf{K}_i$ and $\mathbf{K}_j$, if $\upsilon \neq 1$, as r_{ij} approaches 1, we have $\theta_i^\star$ approaches $\theta_j^\star$.

Proof: Since $\upsilon \neq 1$, from (18) in Theorem 2, we note that $\theta_q^\star$ can be simplified as $\max\{0, t_q\}$, which is a continuous function of t_q, where $t_q = 1/(2(1-\upsilon))(1/(2\lambda)(\boldsymbol{\alpha}^\star \circ \mathbf{y})^\top \mathbf{K}_q(\boldsymbol{\alpha}^\star \circ \mathbf{y}) - \upsilon)$. Hence, we have

$$|\theta_i^\star - \theta_j^\star| \leq |t_i - t_j|, \ \text{ for } i, j = 1, \ldots, Q. \tag{19}$$

This inequality can be obtained by analyzing the following several cases.

1) When $\theta_i^\star$ and $\theta_j^\star$ are both positive, we have $t_i, t_j > 0$ and attain the equality in (19).
2) When $\theta_i^\star > 0$ and $\theta_j^\star = 0$, we have $|\theta_i^\star - \theta_j^\star| = |t_i| \leq |t_i - t_j|$. The inequality is due to the condition that $t_j \leq 0$.

TABLE I
COMPARISON BETWEEN GMKL AND OTHER MODELS

	L_1-MKL	L_2-MKL	GMKL	Lasso	Elastic net	Group Lasso
Sparsity	✓	×	✓	✓	✓	✓
Nonlinearity	✓	✓	✓	×	×	×
Grouping	×	✓	✓	×	✓	×

For the case of $\theta_i^\star = 0$ and $\theta_j^\star > 0$, we can derive the result similarly.

3) When $\theta_i^\star = \theta_j^\star = 0$, the inequality in (19) is satisfied for all t_i and t_j.

From (19), we have

$$|\theta_i^\star - \theta_j^\star| \le \frac{1}{2(1-\upsilon)}\frac{1}{2\lambda}(\boldsymbol{\alpha}^\star \circ \mathbf{y})^\top \mathbf{K}_j(\boldsymbol{\alpha}^\star \circ \mathbf{y})|r_{ij} - 1|. \quad (20)$$

Hence, as $r_{ij} \approx 1$, we have $|\theta_i^\star - \theta_j^\star| \approx 0$. That is, $\theta_i^\star$ approaches $\theta_j^\star$. ■

From (20), we note that the difference of two weights, i.e., $|\theta_i^\star - \theta_j^\star|$, is proportional to the ratio $|r_{ij} - 1|$, and inversely proportional to $2(1-\upsilon)$. The ratio indicates that, if two kernels are strongly correlated, the weights are nearly the same. The inversely proportional value $2(1-\upsilon)$ indicates that a smaller υ will yield closer solutions for the kernel weights. Meanwhile, we should note that the coefficient 2 is introduced due to the use of the L_2-norm on the kernel weights. If the L_p-norm is adopted, the ratio is inversely related to p. As p increases, e.g., approaches to infinity, it will lead to the same weights, i.e., UW-MKL, which is the same as the result in previous MKL models.

In summary, our GMKL contains the following properties.

1) In view of (18), we can see that out GMKL imposes the sparsity on the coefficients of the model. This surpasses those non-sparse MKL models [17], [18], which may be prone to the noise and have a larger computation/storage cost.
2) Theorems 2 and 3 state that the GMKL can provide the grouping effect, which retains more useful information from the data than the L_1-MKL.
3) Nonlinearity is embedded in the formulation of (8) and is represented by the kernels. Our GMKL can therefore capture more information of the data than those statistic models, e.g., the lasso [43] and the elastic net [39].

Table I summarizes the above arguments.

V. OPTIMIZING THE GMKL

Due to the efficiency in solving SVMs, the wrapping-based methods have been adopted to solve the MKL models, e.g., [10], [18], [25], [30], [31], and [44]. In the wrapping-based methods, the first step is to seek the optimal $\hat{\mathbf{w}}, b$ or the dual variable $\boldsymbol{\alpha}$ given a fixed $\boldsymbol{\theta}$ by an SVM solver. The second step is to update the kernel weights $\boldsymbol{\theta}$ to further decrease the objective value of (5) with fixed primal variables $\hat{\mathbf{w}}$ and b or fixed dual variable $\boldsymbol{\alpha}$. Many previously proposed MKL methods try to speed up the model in the second step. For example, a gradient method is proposed in [25] and [44], an semi-infinite linear program method is applied in [18] and [30], and the level method is introduced in [31].

Among these optimization methods, the level method, which is a cutting plane method derived from the family of bundle methods [45], has shown better success in solving machine learning and kernel learning methods. For example, it has been introduced to efficiently solve regularized risk minimization problems [46], the L_1-MKL [31], and neighborhood kernel learning [10]. Hence, in this paper, we adopt the level method to solve our GMKL of (14).

A. GMKL by the Level Method

The key part of the level method is to construct the corresponding lower bound and the upper bound of the objective function. First, we know that $\mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha})$ in (14) is convex on $\boldsymbol{\theta}$ and concave on $\boldsymbol{\alpha}$. According to von Neumann Lemma [47], for any optimal solution $(\boldsymbol{\theta}^\star, \boldsymbol{\alpha}^\star)$, we have

$$\begin{aligned}\mathcal{D}(\boldsymbol{\theta}^\star, \boldsymbol{\alpha}) \le \max_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{D}(\boldsymbol{\theta}^\star, \boldsymbol{\alpha}) &= \mathcal{D}(\boldsymbol{\theta}^\star, \boldsymbol{\alpha}^\star)\\ &= \min_{\boldsymbol{\theta}\in\Theta} \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}^\star) \le \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}^\star). \quad (21)\end{aligned}$$

The above property indicates that our model can easily obtain the corresponding lower bound and the upper bound.

Suppose $\{(\boldsymbol{\theta}^i, \boldsymbol{\alpha}^i)\}_{i=1}^t$ denote the solutions of (14) obtained in the last t iterations. We define the corresponding lower bound $\underline{\mathcal{D}}^t$ and the corresponding upper bound $\overline{\mathcal{D}}^t$ as follows:

$$\underline{\mathcal{D}}^t = \min_{\boldsymbol{\theta}\in\Theta} h^t(\boldsymbol{\theta}), \quad \overline{\mathcal{D}}^t = \min_{1\le i\le t} \mathcal{D}(\boldsymbol{\theta}^i, \boldsymbol{\alpha}^i) \quad (22)$$

where $h^t(\boldsymbol{\theta})$ corresponds to a cutting plane as follows:

$$h^t(\boldsymbol{\theta}) = \max_{1\le i\le t} \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}^i). \quad (23)$$

It is noted that the lower bound is the minimum value at the cutting plane and the upper bound is the minimum objective value attained at previous steps.

We can then define the level set as follows:

$$\mathcal{L}^t = \left\{\boldsymbol{\theta} \in \Theta : h^t(\boldsymbol{\theta}) \le \mathcal{V}^t = \tau\overline{\mathcal{D}}^t + (1-\tau)\underline{\mathcal{D}}^t = \underline{\mathcal{D}}^t + \tau\Delta^t\right\} \quad (24)$$

where $\tau \in (0, 1)$ is a given constant controlling the tradeoff of two bounds. The level set specifies the set of solution where the objective is bounded by the lower bound and the upper bound. The gap Δ^t between the upper bound and the lower bound at each step is defined as

$$\Delta^t = \overline{\mathcal{D}}^t - \underline{\mathcal{D}}^t \quad (25)$$

and measures the suboptimality for the solution $(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^t)$ at each step.

The final step in the level method is to project $\boldsymbol{\theta}^t$ onto the level set $\mathcal{L}^t$ to calculate a new solution $\boldsymbol{\theta}^{t+1}$. That is, we obtain $\boldsymbol{\theta}^{t+1}$ by solving the following quadratic optimization problem:

$$\begin{aligned}\min_{\boldsymbol{\theta}\in\Theta} \quad & \|\boldsymbol{\theta} - \boldsymbol{\theta}^t\|_2^2 \quad (26)\\ \text{s.t.} \quad & \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}^i) \le \mathcal{V}^t, \quad i = 1, \ldots, t.\end{aligned}$$

The intuition of the projection is to make the solution satisfy the level set conditions in a faster way and to require $\boldsymbol{\theta}$'s in

Algorithm 1 Level method for the GMKL

```
Given: predefined tolerant error $\delta > 0$.
Initialization: Let $t = 0$ and $\boldsymbol{\theta}^0 = c\mathbf{1}_Q$, where $c$ is the positive root of the quadratic equation: $(1 - \upsilon)c^2 + \upsilon c - (1/Q) = 0$.
repeat
    1) Solve the dual problem of the SVM with $\sum_{q=1}^{Q} \theta_q^t \mathbf{K}_q$ to get the optimal solution, $\boldsymbol{\alpha}$.
    2) Construct the cutting plane model, $h^t(\boldsymbol{\theta})$, in (23).
    3) Calculate the lower bound $\underline{\mathcal{D}}^t$ and the upper bound $\overline{\mathcal{D}}^t$ in (22), and the gap $\Delta^t$ in (25).
    4) Compute the projection of $\boldsymbol{\theta}^t$ onto the level set $\mathcal{L}^t$ by solving the optimization problem in (26).
    5) Update $t = t + 1$.
until $\Delta^t \le \delta$.
```

two consecutive steps close to each other avoiding oscillation on the solution.

The following pseudocode presents how to solve the GMKL by the level method.

Remarks: Two points about Algorithm 1 need to be emphasized.

1) Initialization of $\boldsymbol{\theta}$: We set $\boldsymbol{\theta}^0$ uniformly at each element, i.e., $\boldsymbol{\theta}^0 = c\mathbf{1}_Q$, where $c > 0$. From Theorem 1, we must have $\upsilon \cdot Q \cdot c + (1 - \upsilon) \cdot Q \cdot c^2 = 1$. This requires seeking the positive root of the quadratic equation as that in Algorithm 1.
2) In terms of computation, the main part of our GMKL is that we have introduced a quadratic constraint in (9). This may require a bit more computation when compared with the L_1-MKL approach. In Algorithm 1, there are two steps involving the quadratic constraint. They are the step 2 in Algorithm 1, which constructs the cutting plane in (23) by solving a linear program with a quadratic constraint, and the step 4 in Algorithm 1, which projects $\boldsymbol{\theta}$ to the level set by solving a quadratic programming with a quadratic constraint in (26). We believe some warm start methods, e.g., solving the corresponding problems with previously obtained optimal value [48], can be adopted to speed up the seeking of the next optimal value.

B. Convergence Analysis

Algorithm 1 is terminated when the gap between the two bounds is small. To analyze the convergence of the level method on our GMKL model, we first have the following theorem to state that, in each iteration, the gap is nonincreasing and the difference between the optimal objective value and the attained objective value is bounded by the gap.

Theorem 4: We have the following properties on the gap, Δ^i, $i = 1, \ldots, t$:

1) $\Delta^i \ge 0$;
2) $\Delta^1 \ge \Delta^2 \ge \ldots \ge \Delta^t$;
3) $|\mathcal{D}(\boldsymbol{\theta}^i, \boldsymbol{\alpha}^i) - \mathcal{D}(\boldsymbol{\theta}^\star, \boldsymbol{\alpha}^\star)| \le \Delta^t$.

We then have the following theorem, which provides the convergence rate of Algorithm 1.

TABLE II

SUMMARY OF THE SYNTHETIC AND UCI DATASETS

Type	Dataset	Training (N)	Test	Dim (d)	Kernel (Q)
Synthetic	Toy 1	150	150	20	273
	Toy 2	150	150	20	273
UCI	Breast	341	342	10	143
	Heart	135	135	13	182
	Ionosphere	175	176	33	442
	Liver	172	173	6	91
	Pima	384	384	8	117
	Sonar	104	104	60	793
	Wdbc	284	285	30	403
	Wpbc	99	99	33	442

Theorem 5: For any $\delta > 0$, Algorithm 1 converges to the desired precision after T steps

$$T \ge \frac{2c(\tau)V^2}{\delta^2} \tag{27}$$

where $c(\tau) = 1/((1 - \tau)^2 \tau(2 - \tau))$, V is a term calculated by $(1/2)NC^2\sqrt{Q}\max_{1 \le q \le Q} \Lambda_{\max}(\mathbf{K}_q)$, and $\Lambda_{\max}(\mathbf{K}_q)$ defines the maximum eigenvalue of matrix $\mathbf{K}_q$.

We put the proof of the Theorems 4 and 5 in Appendixes II and III, respectively. It is noted that the convergence rate of the level method is $O(\delta^{-2})$. According to [45], empirically, a better convergence rate $O(N \log(1/\delta))$ is observed.

VI. EXPERIMENTS

We conduct a series of experiments on evaluating the proposed GMKL in contrast with the L_1-MKL, the L_2-MKL, and the UW-MKL with three objectives. The first objective is to show how our GMKL model can select important kernels in group manners. This is illustrated through two toy examples. The second objective is to show the efficiency of our GMKL model solved by the level method. This is verified by eight datasets from the UCI repository [49]. The third objective is to show that the GMKL can improve the performance on predicting the proteins subcellular localization by different kinds of kernels [28]. A summary of the three types of data, including 13 datasets, is listed in Tables II and III, respectively. Detailed descriptions of the data are in Sections VI-B to VI-D, respectively.

A. Experimental Setup

In the experiment, for all compared four MKL models, the regularization parameter C is tuned by cross validation on one run of the training data. The tradeoff parameter υ for the GMKL is set to 0.5 for simplicity.

The L_1-MKL is solved by the SimpleMKL toolbox [25]. The L_2-MKL is solved by our GMKL.[1] with the parameter $\upsilon = 0$. The optimization on constructing the cutting plane of (23) and seeking the projection of (26) in the level method

[1] Our GMKL toolbox can be downloaded at http://appsrv.cse.cuhk.edu.hk/~hqyang/doku.php?id=gmkl.

TABLE III

SUMMARY OF THE PROTEINS SUBCELLULAR LOCALIZATION DATASETS

Dataset	Classes	Training (N)	Test	Kernel (Q)
Plant	4	470	470	69
Psort+	4	270	271	69
Psort−	5	722	722	69

are solved by a standard toolbox, Mosek.[2] The algorithm parameter τ of the level method is set to 0.9 initially and increased to 0.99 when the ratio $\Delta^t/\mathcal{V}^t$ is less than 0.01, since a larger τ accelerates the projection when the solution is close to the optimal one. All MKL models use the same SVM solver with default settings. Since the UW-MKL is solved by the standard SVM solver once, we do not report its time cost in the experiments. To test the efficiency of our GMKL, we adopt the multiple stopping criteria similar to that in [25], the number of iterations exceeds 500, or the difference of $\boldsymbol{\theta}$ in consecutive step is lower than 0.001. For the L_1-MKL, additional stopping criterion, i.e., the duality gap being lower than 0.01, is set as that in [25].

B. Toy Examples

In designing the synthetic datasets, we have the following expectations: 1) data containing nonlinearity on the features, and 2) data being embedded with redundant features while some features playing the same roles. We then generate two 20-D toy examples by additive models motivated by an example in [50].

1) In example 1, the data are generated by

$$Y_i = \operatorname{sign}\left(\sum_{j=1}^{3} f_1(x_{ij}) + \epsilon_i\right) \tag{28}$$

where sign $(\cdot)$ is determined by the sign of the value in the bracket, $\mathbf{x}$ is uniformly distributed in $[0, 1]^{N\times 20}$, $f_1(a) = -2\sin(2a) + 1 - \cos(2)$, and the noise $\epsilon_i \sim \mathcal{N}(0, 1)$ is a Gaussian noise. Hence, the data contain 17 irrelevant features.

2) In example 2, the data are generated by

$$Y_i = \operatorname{sign}\left(\sum_{j=1}^{3} f_1(x_{ij}) + \sum_{j=4}^{6} f_2(x_{ij}) + \sum_{j=7}^{9} f_3(x_{ij}) + \sum_{j=10}^{12} f_4(x_{ij}) + \epsilon_i\right) \tag{29}$$

where there are four kinds of mapping f_1, f_2, f_3, and f_4. f_1 is the same as Example 1, $f_2(a) = a^2 - (1/3)$, $f_3(a) = a-(1/2)$, and $f_4(a) = e^{-a}+e^{-1}-1$. For $\mathbf{x}$ and ϵ_i, they are the same as Example 1. The output Y_i is determined by the corresponding features, from 1 to 12, of $\mathbf{x}_i$ which are mapped by f_1, f_2, f_3, and f_4, respectively. The data therefore contain eight irrelevant features.

TABLE IV

AVERAGE PERFORMANCE MEASURED BY OUR GMKL, THE L_1-MKL, THE L_2-MKL, THE SPICYMKL, AND THE UW-MKL ALGORITHMS ON TOY EXAMPLES

Dataset	Method	Accuracy	Kernel	Times (s)
Toy 1	GMKL	**70.4** ± 3.3	36.8 ± 5.0	2.9 ± 0.2
	L_1-MKL	69.2 ± 4.5	22.1 ± 5.2	4.4 ± 1.2
	L_2-MKL	68.2 ± 3.0	273	2.9 ± 0.4
	UW-MKL	66.3 ± 5.3	273	–
	SpicyMKL	**70.4** ± 4.0	106.7 ± 4.5	1.5 ± 0.2
Toy 2	GMKL	**72.9** ± 3.2	43.4 ± 7.1	2.8 ± 0.1
	L_1-MKL	72.3 ± 3.1	30.2 ± 8.1	4.9 ± 1.3
	L_2-MKL	71.9 ± 3.6	273	2.9 ± 0.1
	UW-MKL	71.6 ± 4.0	273	–
	SpicyMKL	72.7 ± 3.6	119.8 ± 4.7	1.8 ± 0.4

Another main difference between Example 2 and Example 1 is that the output in Example 2 is dominated by more different function mappings, including polynomial (linear) mapping and exponential function mapping.

It is noted that by the above generation scheme, the data have the following properties.

1) The outputs (labels) of the data are dominated by only some features. The corresponding feature is mapped by a linear function f_3, or nonlinear functions f_1, f_2, and f_4.
2) Each mapping f_i, $i = 1, 2, 3, 4$, acts on three features equally, which implicitly incorporates grouping effect on those features.
3) The mean of the output is zero since each mapping is with zero mean on the corresponding feature.

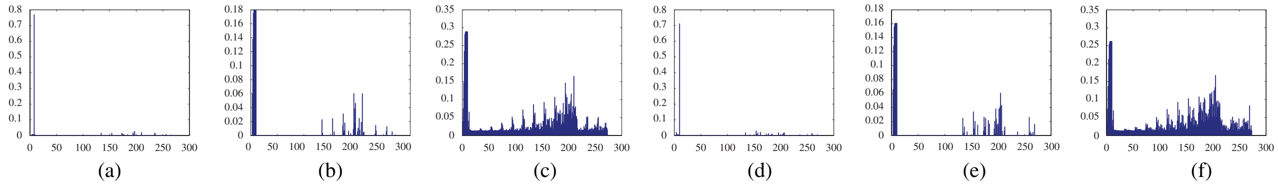
In the experiment, we randomly sample 300 instances, where 150 data are used for training and other 150 data are used for test. Following the settings of [25], we construct the base kernel matrices as follows.

1) Gaussian kernels with 10 different widths ($\{2^{-3}, 2^{-2}, \ldots, 2^{6}\}$) on all features and on each single feature.
2) Polynomial kernels of degree 1 to 3 on all features and on each single feature.

Each base kernel matrix is further normalized to unit trace as [25]. Therefore, we build 273 kernels for the toy examples.

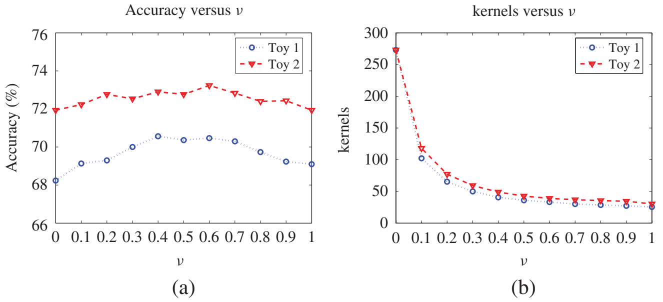
Table IV reports the average accuracy, the number of selected kernels, and executed time, after repeating the algorithms 20 times. Our GMKL obtains significant improvement on the accuracy against the L_1-MKL and the L_2-MKL with 95% confidence level on the paired t-test. The results show that our GMKL can utilize the grouping structure information embedded in the data sufficiently. Table IV also shows that both the L_2-MKL and the UW-MKL achieve worse accuracies than the sparse MKL models. This verifies that the non-sparse MKL models are prone to the noise. In terms of the number of selected kernels, our GMKL selects more kernels, about 1.5 times of that selected by the L_1-MKL, while the L_2-MKL selects all kernels (see Fig. 1 for more details). The computation cost of our GMKL and the L_2-MKL is nearly

[2] Available at: http://www.mosek.com.



(a) (b) (c) (d) (e) (f)

Fig. 1. Coefficients of kernel weights learned by the L_1-MKL, the GMKL, and the L_2-MKL on the Toy 1 example and the Toy 2 example, respectively. The L_1-MKL selects few kernels and discards some useful information. The L_2-MKL selects all kernels and is prone to the noise. Meanwhile, the GMKL selects suitable kernels with the grouping effect, see detailed description in the text. (a) L_1-MKL on Toy 1. (b) GMKL on Toy 1. (c) L_2-MKL on Toy 1. (d) L_1-MKL on Toy 2. (e) GMKL on Toy 2. (f) L_2-MKL on Toy 2.



(a) (b)

Fig. 2. Accuracy and the number of selected kernels by the GMKL with respect to υ on the toy datasets. The best accuracy for the Toy 1 dataset is achieved when $\upsilon = 0.4$ and it is achieved for the Toy 2 dataset when $\upsilon = 0.6$. The number of selected kernels decreases as υ increases (see text for more descriptions). (a) Accuracy versus υ. (b) Number of selected kernels versus υ.

the same, and they cost less time than that of the L_1-MKL. This is because the level method consumes less outer iterations than the SimpleMKL [25] used. In addition, we also report the results of the SpicyMKL [35]. (Both regularization parameters are selected in $[10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{2}, \text{and } 10^{3}]$). The results show that the SpicyMKL also achieves similar accuracy to our GMKL, but it usually selects more kernels than our GMKL.

To show whether the grouping effect is achieved, i.e., similar kernels attain close kernel coefficient values, we plot the average coefficients obtained by the L_1-MKL, the GMKL, and the L_2-MKL in Fig. 1. The kernel coefficients are shown in the following order: the Gaussian kernels on all features; the polynomial kernels on all features; the Gaussian kernels on each single feature; the polynomial kernels on each single feature. In Example 1, the output response is dominated by three features which are mapped by a linear combination of sine and cosine functions. Hence, Gaussian kernels on these three features can form similar kernels. Fig. 1 clearly shows that our GMKL attains close coefficient values on some Gaussian kernels on all features and on the designed three features, i.e., forming the grouping effect on the kernels. Similarly, the kernel coefficients on the toy dataset 2 also reveal the same property. Hence, Fig. 1 shows the grouping effect and the sparsity of our GMKL. The results in the figures refer to the results in the fourth column of the Table IV.

Fig. 1 further shows the average coefficients obtained by the L_1-MKL, the GMKL, and the L_2-MKL. The figure again shows the grouping effect and the sparsity of our GMKL. The results in the figures refer to the results in the fourth column of the Table IV.

We further test the effect of υ on the accuracy and the number of selected kernels for the toy datasets. We vary υ from 0 to 1 with an incremental step of 0.1 and show the results in Fig. 2. Actually, Fig. 2 includes the results reported in Table IV, the L_1-MKL ($\upsilon = 1$), the L_2-MKL ($\upsilon = 0$), and our GMKL with $\upsilon = 0.5$. It is shown that the optimal υ is around 0.5 for both toy datasets. Fig. 2(b) indicates that, as υ increases, the number of selected kernels decreases. This shows that the optimal υ is data-dependent, i.e., a better υ corresponds to the suitable number of kernels selected for that training data. Hence, usually we can tune the parameter υ by cross validation on the training data.

C. UCI Datasets

In order to verify the performance of our GMKL on datasets which do not show manifest group structure on the base kernels, we employ eight UCI datasets in our test from the UCI repository [49] in our test. These datasets have been frequently used in evaluating the MKL models [12], [25], and [31].

We repeat all the algorithms 20 times on each dataset. In each run, 50% of the examples are randomly selected as the training data and the remaining data are used for testing. The training data are normalized to have zero mean and unit variance, and the test data are then normalized using the mean and variance of the training data. The construction and the postprocessing of the base kernel matrices are conducted in the same way as the synthetic data in Section VI-B.

Table V reports the average results, including accuracy, the number of selected kernels, and the running time, on the UCI datasets. Our GMKL achieves the highest accuracy for five datasets: i.e., "Breast," "Heart," "Pima," "Wdbc," and "Wpbc." Especially, for the datasets of "Pima," our GMKL obtains significantly better results. The L_2-MKL gets the highest accuracy for the rest three datasets: "Ionosphere," "Liver," and "Sonar," and attains significantly better results for "Liver" and "Sonar." The UW-MKL gets the same highest accuracy as the GMKL for "Breast" and "Heart." It is important to note that better results can be obtained by tuning υ through cross validation on the training data. For example, the cross-validation procedure on the "Ionosphere" dataset suggests that a smaller υ with the value near zero can recover the result of the L_2-MKL.

In terms of the number of selected kernels, on average, our GMKL selects a few more kernels than the L_1-MKL,

TABLE V

AVERAGE PERFORMANCE MEASURED BY THE GMKL, THE L_1-MKL, THE L_2-MKL, AND THE UW-MKL ALGORITHMS ON UCI DATASETS. BETTER RESULTS ARE IN BOLD. SIGNIFICANTLY BETTER RESULTS WITH 95% CONFIDENCE LEVEL OVER OTHER METHODS ARE INDICATED BY †

Dataset	Method	Accuracy	Kernel	Times(s)
Breast	GMKL	**97.2** ± 0.5	61.1 ± 6.5	2.8 ± 0.5
	L_1-MKL	97.0 ± 0.7	18.6 ± 3.8	23.0 ± 3.9
	L_2-MKL	96.9 ± 0.4	143	5.1 ± 0.3
	UW-MKL	**97.2** ± 0.5	143	–
Heart	GMKL	**83.9** ± 1.9	38.5 ± 5.4	1.4 ± 0.1
	L_1-MKL	83.4 ± 2.6	29.7 ± 4.6	3.5 ± 0.7
	L_2-MKL	82.8 ± 2.5	182	1.7 ± 0.1
	UW-MKL	**83.9** ± 1.9	182	–
Ionosphere	GMKL	91.8 ± 1.7	66.5 ± 7.2	5.1 ± 0.3
	L_1-MKL	91.5 ± 2.1	38.4 ± 5.0	19.2 ± 3.3
	L_2-MKL	**92.0** ± 1.8	442	4.0 ± 0.4
	UW-MKL	89.9 ± 1.8	442	–
Liver	GMKL	67.6 ± 1.8	19.5 ± 1.7	1.0 ± 0.0
	L_1-MKL	64.3 ± 2.8	9.2 ± 3.0	1.7 ± 0.4
	L_2-MKL	†**69.7** ± 2.2	91	1.4 ± 0.0
	UW-MKL	67.2 ± 4.6	91	–
Pima	GMKL	†**76.9** ± 1.6	27.1 ± 2.4	3.8 ± 0.2
	L_1-MKL	76.5 ± 1.9	18.7 ± 2.7	24.8 ± 3.4
	L_2-MKL	76.0 ± 1.8	117	6.2 ± 1.0
	UW-MKL	76.2 ± 1.7	117	–
Sonar	GMKL	80.4 ± 4.1	81.1 ± 6.5	12.4 ± 0.6
	L_1-MKL	80.4 ± 4.2	60.3 ± 7.4	16.7 ± 2.0
	L_2-MKL	†**83.8** ± 3.7	793	3.9 ± 0.3
	UW-MKL	81.5 ± 4.3	793	–
Wdbc	GMKL	**96.0** ± 1.1	79.7 ± 7.6	6.6 ± 0.8
	L_1-MKL	95.3 ± 1.4	34.9 ± 8.9	37.8 ± 5.8
	L_2-MKL	95.9 ± 0.7	403	7.8 ± 1.6
	UW-MKL	93.9 ± 1.0	403	–
Wpbc	GMKL	**76.7** ± 3.3	275.4 ± 96.9	1.3 ± 1.0
	L_1-MKL	76.6 ± 2.8	40.4 ± 10.2	4.8 ± 1.0
	L_2-MKL	76.3 ± 3.7	442	1.6 ± 0.2
	UW-MKL	76.6 ± 2.9	442	–

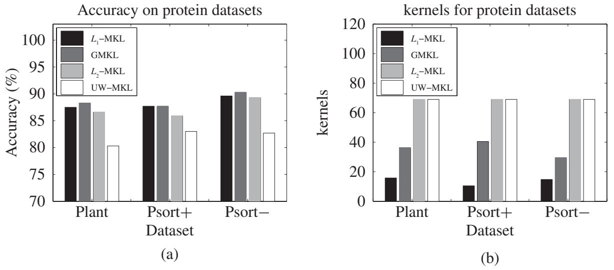


Fig. 3. Accuracy and the number of kernels selected by the L_1-MKL, the GMKL, and the L_2-MKL on the protein subcellular localization datasets, where the L_2-MKL, and the UW-MKL select all the 69 kernels. Our GMKL achieves the best results on all datasets and selects about 3-4 times the number of kernels compared to that selected by the L_1-MKL. It should be noted that here the accuracy of the plant dataset is measured by the Matthew correlation coefficient (MCC) [28], while for the Psort+ and the Psort− datasets, it is measured by the F1 score. (a) Accuracy. (b) Number of selected kernels.

owing to the grouping effect on some features. This is expected since, among all the datasets, our GMKL achieves no worse results than the L_1-MKL. Especially, our GMKL improves the accuracy from 64.3% to 67.6% for the Liver dataset, and from 95.3% to 96.0% for the Wdbc dataset.

For the running time, our GMKL is efficient. The time needed by our GMKL and the L_2-MKL is much less than that used in the L_1-MKL for the datasets of Breast, Ionosphere, Pima, and Wdbc. Especially, for the datasets of Breast, Pima, and Wdbc, the number of data points is larger than other datasets, and the SimpleMKL costs more time. This is because that the simple MKL has to solve more quantum photonics problems when updating the descent direction. When the number of training samples is large, more time is required in the SVM solver.

D. Protein Subcellular Localization Datasets

Three datasets are used to predict the proteins subcellular localization,[3] where the plant dataset of TargetP is a four-class problem, and the other two datasets of bacterial protein locations are the Psort+ dataset consisting of four classes and the Psort− dataset consisting of five classes. The summary of the datasets is in Table III. MKL methods have succeeded in these datasets with those well-defined graph kernels [18], [28]. We hypothesize that the graph kernels may still provide the grouping effect and help to improve the prediction performance.

For the proteins subcellular localization datasets, we follow the setup of [28] and construct 69 kernels: 2 kernels on phylogenetic trees, 3 kernels from BLAST E-values, and 64 sequence motif kernels. Each kernel for the proteins subcellular localization datasets is normalized such that the implied variance equals one as [28] by $\mathbf{K}(\mathbf{x}, \mathbf{z}) = \mathbf{K}(\mathbf{x}, \mathbf{z})/(1/N \sum_{i=1}^{N} \mathbf{K}(\mathbf{x}_i, \mathbf{x}_i) - 1/N^2 \sum_{i,j=1}^{N} \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j))$.

Different from [28], we randomly split the protein subcellular localization datasets into two parts equally, where half the data are used for training and the rest of the data are used for test. We use the 1-versus-rest scheme on the multiclass classification problems. As in [28], the MCC is used to evaluate the plant dataset, and the F1 score is used to evaluate the Psort+ and the Psort− datasets.

Fig. 3(a) reports the average results on 10 runs. Our GMKL achieves the best results on all three datasets. The MCC obtained by our GMKL for the plant dataset is 88.3% compared to 87.5% obtained by the L_1-MKL, 86.6% obtained by the L_2-MKL, and 80.3% obtained by the UW-MKL. For the two bacterial protein locations datasets, our GMKL and the L_1-MKL get the same 87.7% F1 score compared to the L_2-MKL of 85.9% and the UW-MKL of 83.0% for the Psort+ dataset and obtains 90.3% F1 score compared to the L_1-MKL of 89.6%, the L_2-MKL of 89.3%, and the UW-MKL of 82.7%. Hence, the results verify our hypothesis.

To further verify whether our GMKL model performs statistically better than the other three MKL methods, we report the p-values of the paired t-test of our GMKL on the L_1-MKL, the L_2-MKL, the UW-MKL in Table VI. The results show that our GMKL improves the classification accuracy significantly compared to the UW-MKL for all three protein datasets. Our GMKL performs significantly better than the L_2-MKL for the Psort+ dataset and the Psort− dataset. Compared to the L_1-MKL, our GMKL performs significantly better than the Psort− dataset.

[3] Available at: http://www.fml.tuebingen.mpg.de/raetsch/suppl/protsubloc/.

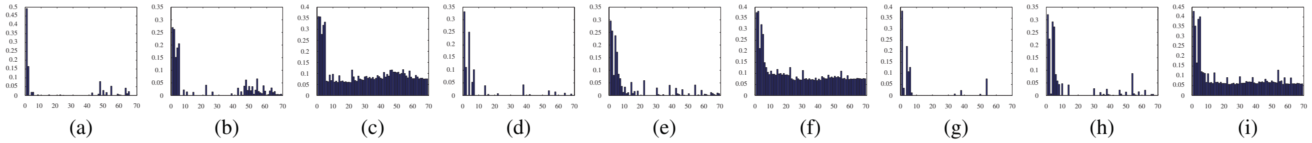


Fig. 4. Coefficients of kernel weights learned by the L_1-MKL, the GMKL, and the L_2-MKL on the protein subcellular localization datasets. The horizontal axis indexes the 69 kernels. The two phylogenetic profile kernels and the three BLAST E-value kernels are on the left. The L_1-MKL selects few kernels and the L_2-MKL selects all kernels. Meanwhile, the GMKL selects suitable number of kernels. Most of the grouping kernels are from the sequence motif kernels. (a) L_1-MKL on Plant. (b) GMKL on Plant. (c) L_2-MKL on Plant. (d) L_1-MKL on Psort+. (e) GMKL on Psort+. (f) L_2-MKL on Psort+. (g) L_1-MKL on Psort–. (h) GMKL on Psort–. (i) L_2-MKL on Psort–.

TABLE VI

p-VALUES OF THE PAIRED t-TEST OF OUR GMKL VERSUS THE L_1-MKL, THE L_2-MKL, AND THE UW-MKL ON THE PROTEINS SUBCELLULAR LOCALIZATION DATASETS

Dataset	GMKL versus L_1-MKL	GMKL versus L_2-MKL	GMKL versus UW-MKL
Plant	0.319	0.054	**0.000**
Psort+	0.545	**0.047**	**0.002**
Psort−	**0.049**	**0.040**	**0.000**

Fig. 3(b) shows the number of selected kernels by the L_1-MKL, the GMKL, and the L_2-MKL. Our GMKL again selects more kernels than the L_1-MKL. Fig. 4 further shows the obtained kernel weights by the L_1-MKL, our GMKL, and the L_2-MKL for the protein subcellular localization datasets. Our GMKL again can obtain sparse solutions with the grouping effect. It is noted that most of the groups are embedded in the sequence motif kernels and captured by our GMKL.

In summary, the experimental results in the above section indicate the good performance in terms of accuracy, sparsity, and efficiency. The advantage of our GMKL is more explicit on data with latent group structure.

VII. CONCLUSION

In this paper, we presented a GMKL model by introducing a linear combination of the L_1-norm and the squared L_2-norm regularization on the kernel weights to seek the optimal kernel combination. Our GMKL generalizes the previously proposed L_1-MKL and the L_2-MKL methods. The theoretical analysis on the GMKL guarantees to having sparse solutions and also encourages the grouping effect. Moreover, the optimization of GMKL is a convex optimization problem, where the global optimality can be assured. We further adopted the level method to efficiently solve the optimization problem, followed by the convergence analysis and optimal condition on the algorithm.

Experimental results on both synthetic and real-world datasets indicate that the proposed GMKL can take advantage of the group structure of data, and thus produce sparse solutions accordingly. In addition, it keeps the balance between accuracy and sparsity of MKL: it improves the accuracy of the L_1-MKL, and at the same time produces more sparse solutions than the L_2-MKL while achieving competitive accuracy. Moreover, the reported running time on the datasets indicates the efficiency of the level method on solving our GMKL.

There are several future works associated with our GMKL. First, it would be interesting to apply our GMKL model in other applications, e.g., regression, multiclass classification problems, and so on. Second, it is promising to employ advanced optimization techniques to speed up our GMKL, e.g., employing warm start on the previously obtained solution on solving the optimization problem with a quadratic constraint, or solving the optimization problem by second-order methods or coordinate-wise optimizers. Third, it is attractive to extend our GMKL to include the UW-MKL as a special case.

APPENDIX I
PROOF OF THEOREM 2

Proof:

1) When $\upsilon \neq 1$, we denote the objective in (17) as $\mathcal{L}(\boldsymbol{\theta}) = \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}^\star) + \lambda\left(\upsilon\|\boldsymbol{\theta}\| + (1-\upsilon)\|\boldsymbol{\theta}\|_2^2\right)$. Since the objective function is continuous on $\boldsymbol{\theta}$, its minimizer should satisfy
$$\frac{\partial \mathcal{L}}{\partial \theta_q} = -\frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y})^\top \mathbf{K}_q(\boldsymbol{\alpha} \circ \mathbf{y}) + \lambda(\upsilon + 2(1-\upsilon)\theta_q) = 0.$$
As $\lambda > 0$, combining with $\boldsymbol{\theta} \geq \mathbf{0}$, we get $\theta_q^\star$ as (18). When $\mathbf{K}_i = \mathbf{K}_j$, we then have $\theta_i^\star = \theta_j^\star$.

2) When $\upsilon = 1$, the regularizer, $\upsilon\|\boldsymbol{\theta}\| + (1-\upsilon)\|\boldsymbol{\theta}\|_2^2$ reduces to lasso regularizer. It can be easily verified that both minimizers, i.e., $\boldsymbol{\theta}^\star$ and $\tilde{\boldsymbol{\theta}}$, achieve the same objective value. ■

APPENDIX II
PROOF OF THEOREM 4

Proof: To prove Theorem 4, we first need the following proposition.

Proposition 1: For any $\boldsymbol{\theta} \in \Theta$, we have:

1) $h^{t+1}(\boldsymbol{\theta}) \geq h^t(\boldsymbol{\theta})$;
2) $h^t(\boldsymbol{\theta}) \leq \max_{\boldsymbol{\alpha} \in \mathcal{A}} \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha})$.

The above two propositions can be easily checked by their definitions. They support the definition of the lower bound and the upper bound in (22).

Next, we have the following lemma indicating the relation between bounds:

Lemma 1: Suppose we have a sequence of bounds, $\{\underline{\mathcal{D}}^t\}_{t=1}^T$ and $\{\overline{\mathcal{D}}^t\}_{t=1}^T$, defined in (22). We can then obtain the following properties for their relation.

1) $\underline{\mathcal{D}}^t \leq \mathcal{D}(\boldsymbol{\theta}^\star, \boldsymbol{\alpha}^\star) \leq \overline{\mathcal{D}}^t$.
2) $\overline{\mathcal{D}}^1 \geq \overline{\mathcal{D}}^2 \geq \ldots \geq \overline{\mathcal{D}}^t$.
3) $\underline{\mathcal{D}}^1 \leq \underline{\mathcal{D}}^2 \leq \ldots \leq \underline{\mathcal{D}}^t$.

We now give a short proof of 1) in Lemma 1. Parts 2) and 3) of Lemma 1 can be easily verified based on the definitions.

First, Proposition 1 indicates that for any $\boldsymbol{\theta} \in \Theta$, $h^t(\boldsymbol{\theta}) \le \max_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha})$. Hence

$$\underline{\mathcal{D}}^t = \min_{\boldsymbol{\theta}\in\Theta} h^t(\boldsymbol{\theta}) \le \min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \mathcal{D}(\boldsymbol{\theta}^\star, \boldsymbol{\alpha}^\star).$$

Second, since $\mathcal{D}(\boldsymbol{\theta}^t, \boldsymbol{\alpha}^t) = \max_{\boldsymbol{\alpha}\in\mathbf{A}} \mathcal{D}(\boldsymbol{\theta}^t, \boldsymbol{\alpha})$, then we have

$$\begin{aligned}\overline{\mathcal{D}}^t &= \min_{1\le k\le t} \mathcal{D}(\boldsymbol{\theta}^k, \boldsymbol{\alpha}^k) = \min_{\boldsymbol{\theta}\in\{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_t\}} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha})\\ &\ge \min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{D}(\boldsymbol{\theta}, \boldsymbol{\alpha}) = \mathcal{D}(\boldsymbol{\theta}^\star, \boldsymbol{\alpha}^\star).\end{aligned}$$

The above two results conclude 1) of Lemma 1.

Hence, by applying 1) of Lemma 1, we can obtain 1) and 3) of Theorem 4. Combining 2) and 3) of Lemma 1, we have 2) of Theorem 4. ■

APPENDIX III
PROOF OF THEOREM 5

Proof: Before starting the proof, we first introduce the theorem.

Theorem 6 ([45, Th. 8.2.1, Ch. 8]): Let $\mathcal{D}$ be a convex and Lipschitz continuous function defined on the domain Θ of diameter $D(\Theta)$ with the Lipschitz constant being $L(\mathcal{D}) < \infty$. Applying the level method to this convex problem, the gap Δ^T converges to 0, or for any positive δ, one has

$$T \ge c(\tau)\left(\frac{L(\mathcal{D})D(\Theta)}{\delta}\right)^2 \tag{30}$$

where $c(\tau) = 1/((1-\tau)^2\tau(2-\tau))$.

We then can derive the result based on the above theorem.

First, let us define $\theta_{\max}$ be the maximum element value of $\boldsymbol{\theta}$. We then have $\theta_{\max} \le 1$. It can be derived by

$$\begin{aligned}1 &= \upsilon\|\boldsymbol{\theta}\|_1 + (1-\upsilon)\|\boldsymbol{\theta}\|_2^2, \quad \text{from Theorem 1}\\ &\ge \upsilon\theta_{\max} + (1-\upsilon)\theta_{\max}^2, \quad \text{by } \boldsymbol{\theta} \ge \mathbf{0}.\end{aligned} \tag{31}$$

The above inequality derives $\theta_{\max} \le 1$, so as $\boldsymbol{\theta} \le \mathbf{1}$.

Next, by applying $\boldsymbol{\theta} \le \mathbf{1}$, we have

$$\boldsymbol{\theta}^\top\boldsymbol{\theta} \le \mathbf{1}^\top\boldsymbol{\theta} = \|\boldsymbol{\theta}\|_1, \quad \forall\boldsymbol{\theta}\in\Theta. \tag{32}$$

Hence, $\forall\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Theta$, we have

$$\begin{aligned}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2 &\le \boldsymbol{\theta}^\top\boldsymbol{\theta} + \boldsymbol{\theta}'^\top\boldsymbol{\theta}'\\ &= \upsilon(\boldsymbol{\theta}^\top\boldsymbol{\theta} + \boldsymbol{\theta}'^\top\boldsymbol{\theta}') + (1-\upsilon)(\boldsymbol{\theta}^\top\boldsymbol{\theta} + \boldsymbol{\theta}'^\top\boldsymbol{\theta}')\\ &\le \upsilon(\|\boldsymbol{\theta}\|_1 + \|\boldsymbol{\theta}'\|_1) + (1-\upsilon)(\|\boldsymbol{\theta}\|_2^2 + \|\boldsymbol{\theta}'\|_2^2)\\ &= 1 + 1 = 2\,.\end{aligned}$$

We then obtain the diameter $D(\Theta)$ as

$$D(\Theta) = \max_{\boldsymbol{\theta},\boldsymbol{\theta}'\in\Theta} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2 = \sqrt{2}. \tag{33}$$

Further, the Lipschitz constant for the GMKL is

$$\begin{aligned}L_{\boldsymbol{\theta}}(\mathcal{D}) &= \max_{\boldsymbol{\theta}\in\Theta,\boldsymbol{\alpha}\in\mathcal{A}} \|\nabla\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{\theta}, \boldsymbol{\alpha})\|_2 = \max_{\boldsymbol{\alpha}\in\mathcal{A}} \|[\mathbf{V}_1, \ldots, \mathbf{V}_Q]^\top\|_2\\ &\le \frac{1}{2}NC^2\sqrt{Q}\max_{1\le q\le Q} \Lambda_{\max}(\mathbf{K}_q)\end{aligned} \tag{34}$$

where $\mathbf{V}_q = (1/2)(\boldsymbol{\alpha}\circ\mathbf{y})^\top\mathbf{K}_q(\boldsymbol{\alpha}\circ\mathbf{y})$, and $\Lambda_{\max}(\mathbf{K}_q)$ defines the maximum eigenvalue of the matrix $\mathbf{K}_q$.

Substituting (33) and (34) into (30) of Theorem 6, we can obtain the result and conclude the proof. ■

REFERENCES

[1] K. Huang, H. Yang, I. King, and M. R. Lyu, "Maxi-min margin machine: Learning large margin classifiers locally and globally," *IEEE Trans. Neural Netw.*, vol. 19, no. 2, pp. 260–272, Feb. 2008.

[2] K.-Z. Huang, H.-Q. Yang, I. King, and M. Lyu, "Machine learning: Modeling data locally and globally," in *Advanced Topics in Science and Tecnology in China*, 1st ed. New York: Springer-Verlag, Apr. 2008.

[3] B. Schölkopf and A. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.

[4] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[5] G. R. G. Lanckriet, T. D. Bie, N. Cristianini, M. I. Jordan, and W. S. Noble, "A statistical framework for genomic data fusion," *Bioinformatics*, vol. 20, no. 16, pp. 2626–2635, 2004.

[6] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero norm with linear models and kernel methods," *J. Mach. Learn. Res.*, vol. 3, pp. 1439–1461, Mar. 2003.

[7] Y. Chen, M. R. Gupta, and B. Recht, "Learning kernels from indefinite similarities," in *Proc. 26th Int. Conf. Mach. Learn.*, Montreal, QC Canada, 2009, pp. 1–8.

[8] M. Gönen and E. Alpaydin, "Localized multiple kernel learning," in *Proc. Int. Conf. Mach. Learn.*, Helsinki, Finland, 2008, pp. 352–359.

[9] M. Hu, Y. Chen, and J. T.-Y. Kwok, "Building sparse multiple-kernel SVM classifiers," *IEEE Trans. Neural Netw.*, vol. 20, no. 5, pp. 827–839, May 2009.

[10] J. Liu, J. Chen, S. Chen, and J. Ye, "Learning the optimal neighborhood kernel for classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 2009, pp. 1144–1149.

[11] I. W.-H. Tsang and J. T.-Y. Kwok, "Efficient hyperkernel learning using second-order cone programming," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 48–58, Jan. 2006.

[12] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, Jan. 2004.

[13] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. 21st Int. Conf. Mach. Learn.*, Banff, AB, Canada, 2004, pp. 41–48.

[14] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola, "On kernel-target alignment," in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2001, pp. 367–373.

[15] J. Ye, J. Chen, and S. Ji, "Discriminant kernel and regularization parameter learning via semidefinite programming," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 1095–1102.

[16] C. Cortes, M. Mohri, and A. Rostamizadeh, "L_2 regularization for learning kernels," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 1–8.

[17] M. Kloft, U. Brefeld, P. Laskov, and S. Sonnenburg, "Non-sparse multiple kernel learning," in *Proc. NIPS Workshop Kernel Learn.: Autom. Sel. Optimal Kernels*, Whistler, BC, Canada, 2008, pp. 1–4.

[18] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien, "Efficient and accurate l_p-norm multiple kernel learning," in *Proc. Adv. Neural Inf. Process. Syst. 22*, Vancouver, BC, Canada, 2010, pp. 997–1005.

[19] S. Sonnenburg, G. Rätsch, and C. Schäfer, "Learning interpretable SVMs for biological sequence classification," in *Research in Computational Molecular Biology* (LNBI), vol. 3500. Berlin, Germany: Springer-Verlag, 2005, pp. 389–407.

[20] Z. Harchaoui and F. Bach, "Image classification with segmentation graph kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Minneapolis, MN, Jun. 2007, pp. 1–8.

[21] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Royal Stat. Soc., Ser. B*, vol. 68, no. 1, pp. 49–67, 2006.

[22] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *J. Royal Stat. Soc., Ser. B*, vol. 67, no. 1, pp. 91–108, 2005.

[23] F. R. Bach, "Consistency of the group lasso and multiple kernel learning," *J. Mach. Learn. Res.*, vol. 9, pp. 1179–1225, Jun. 2008.

[24] M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy, "Composite kernel learning," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, 2008, pp. 1040–1047.

[25] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, Nov. 2008.

[26] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. New York: Springer-Verlag, 1999.

[27] C. A. Micchelli and M. Pontil, "Learning the kernel function via regularization," *J. Mach. Learn. Res.*, vol. 6, pp. 1099–1125, Jul. 2005.

[28] A. Zien and C. S. Ong, "Multiclass multiple kernel learning," in *Proc. 24th Int. Conf. Mach. Learn.*, Corvallis, OR, 2007, pp. 1191–1198.

[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[30] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *J. Mach. Learn. Res.*, vol. 7, pp. 1531–1565, Jul. 2006.

[31] Z. Xu, R. Jin, I. King, and M. R. Lyu, "An extended level method for efficient multiple kernel learning," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. 2009, pp. 1825–1832.

[32] Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu, "Simple and efficient multiple kernel learning by group lasso," in *Proc. Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, pp. 1–8.

[33] M. Kowalski, M. Szafranski, and L. Ralaivola, "Multiple indefinite kernel learning with mixed norm regularization," in *Proc. 26th Int. Conf. Mach. Learn.*, Montreal, QC, Canada, 2009, pp. 545–552.

[34] S. N. Jagarlapudi, D. Govindaraj, R. S. C. Bhattacharyya, A. Ben-Tal, and K. R. Ramakrishnan, "On the algorithmics and applications of a mixed-norm based kernel learning formulation," in *Proc. Adv. Neural Inf. Process. Syst. 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Vancouver, BC, Canada, 2010, pp. 844–852.

[35] R. Tomioka and T. Suzuki. (2010). Sparsity-accuracy trade-off in MKL [Online]. Available: http://arxiv.org/abs/1001.2615v1

[36] M. Kloft, U. Rückert, and P. Bartlett. (2010). A unifying view of multiple kernel learning. in *Proc. Eur. Conf. Mach. Learn.* [Online]. Available: http://arxiv.org/abs/1005.0437

[37] C. Longworth and M. J. F. Gales, "Combining derivative and parametric kernels for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 4, pp. 748–757, May 2009.

[38] J. Shawe-Taylor, "Kernel learning for novelty detection," in *Proc. NIPS Workshop Kernel Learn.: Autom. Sel. Optimal Kernels*, Dec. 2008, pp. 1–45.

[39] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Royal Stat. Soc., Ser. B*, vol. 67, no. 2, pp. 301–320, 2005.

[40] I. Steinwart and A. Christmann, *Support Vector Machines*. New York: Springer-Verlag, 2008.

[41] L. Bottou and C.-J. Lin, "Support vector machine solvers," in *Large Scale Kernel Machines*, L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, Eds. Cambridge, MA: MIT Press, 2007, pp. 301–320.

[42] P.-H. Chen, R.-E. Fan, and C.-J. Lin, "A study on SMO-type decomposition methods for support vector machines," *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 893–908, Jul. 2006.

[43] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[44] M. Varma and B. R. Babu, "More generality in efficient multiple kernel learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 382. Montreal, QC, Canada, 2009, pp. 1–8.

[45] A. Nemirovski, "Efficient methods in convex programming," Faculty Ind. Eng. Manage., Technion - Israel Inst. Technol., Haifa, Israel, Lecture Notes, 1994.

[46] A. J. Smola, S. V. N. Vishwanathan, and Q. V. Le, "Bundle methods for machine learning," in *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press, 2007, pp. 1–8.

[47] J. Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, 3rd ed. Princeton, NJ: Princeton Univ. Press, 1953.

[48] K. Koh, S.-J. Kim, and S. Boyd, "An interior-point method for large-scale l_1-regularized logistic regression," *J. Mach. Learn. Res.*, vol. 8, pp. 1519–1555, Jul. 2007.

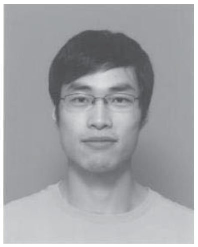
[49] C. Blake and C. Merz. (1998). *UCI Repository of Machine Learning Databases* [Online]. Available: http://archive.ics.uci.edu/ml/

[50] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz, *Nonparametric and Semiparametric Models*. New York: Springer-Verlag, 2004.

Haiqin Yang (S'09) received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2010.

His current research interests include machine learning, data mining, and financial engineering.

Dr. Yang won the Global Scholarship Program for Research Excellence to study convex optimization at Stanford University in 2009 and won the Pacific Century CyberWorks Foundation Scholarship in 2010.



Zenglin Xu (M'09) received the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, in 2009.

He is currently a Post-Doctoral Researcher in the Department of Computer Science, Purdue University, West Lafayette, IN. Earlier, he was a Researcher in the Max Planck Institute for Informatics, Saarbruecken, Germany, and Saarland University, Saarbruecken. From 2007 to 2008, he was a Visiting Researcher in Michigan State University, East Lansing, working on the problem of semisupervised learning and kernel learning. He has published papers in several top journals and conference proceedings, such as the Conference on Neural Information Processing Systems, the International Conference on Machine Learning, the International Joint Conference on Artificial Intelligence, the Association for the Advancement of Artificial Intelligence (AAAI), the Uncertainty in Artificial Intelligence, the Conference on Information and Knowledge Management, and the International Conference on Data Mining. His current research interests include machine learning and its applications to information retrieval, web search, and social computing.

Dr. Xu has served as Program Committee Member in a number of conferences, such as AAAI, the International Joint Conference on Neural Networks, WI, and the International Conference on Neural Information Processing.

Jieping Ye received the Ph.D. degree in computer science from the University of Minnesota, Twin Cities, in 2005.

He is an Associate Professor of computer science and engineering at Arizona State University (ASU), Tempe. His current research interests include machine learning, data mining, and biomedical informatics.

Dr. Ye won the Outstanding Student Paper Award at the International Conference on Machine Learning in 2004, the SCI Young Investigator of the Year Award at ASU in 2007, the SCI Researcher of the Year Award at ASU in 2009, the National Science Foundation CAREER Award in 2010, and the Knowledge Discovery and Data Mining Best Research Paper Award honorable mention in 2010.

Irwin King (SM'08) received the B.Sc. degree in engineering and applied science from the California Institute of Technology, Pasadena, and the M.Sc. and Ph.D. degrees in computer science from the University of Southern California, Los Angeles.

He is a Professor in the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong. He has over 30 research and applied grants. He has contributed over 20 book chapters and edited several volumes. He has published/presented over 200 technical publications in journals/conferences in his areas of expertise. His current research interests include social computing, machine learning, web intelligence, and multimedia processing.

Dr. King is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS. He is a member of several editorial boards. He is a member of the Association for Computing Machinery, the International Neural Network Society (INNS), and the Asian Pacific Neural Network Assembly (APNNA). Currently, he is serving as a member of the Board of Governors of INNS and a Vice-President and a Governing Board Member of APNNA.

Michael R. Lyu (F'04) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, the M.S. degree in computer engineering from the University of California, Santa Barbara, and the Ph.D. degree in computer engineering from the University of California, Los Angeles.

He is a Professor in the Computer Science and Engineering Department, Chinese University of Hong Kong, Hong Kong. He has worked at the Jet Propulsion Laboratory, Pasadena, CA, Bellcore, Piscataway, NJ, and the Bell Laboratory, Murray Hill, NJ, and taught at the University of Iowa, Iowa City. He has participated in more than 30 industrial projects. He has published close to 400 papers in the following areas. His current research interests include software engineering, distributed systems, multimedia technologies, machine learning, social computing, and mobile networks.

Prof. Lyu initiated the International Symposium on Software Reliability Engineering (ISSRE), and was a Program Chair for ISSRE in 1996, the Program Co-Chair for the Tenth International World Web Conference, the Symposium on Reliable Distributed Systems in 2005, the International Conference on e-Business Engineering in 2007, and the International Conference on Services Computing in 2010. He was the General Chair for ISSRE in 2001, the Pacific Rim International Symposium on Dependable Computing in 2005, and the International Conference on Dependable Systems and Networks in 2011. He also received the Best Paper Awards in ISSRE in 1998 and 2003, and the SigSoft Distinguished Paper Award in International Conference on Software Engineering in 2010. He is a Fellow of the American Association for the Advancement of Science. He has been named by the IEEE Reliability Society as the Reliability Engineer of the Year in 2011, for his contributions to software reliability engineering and software fault tolerance.