Soft Margin Multiple Kernel Learning

Xinxing Xu, Ivor W. Tsang, and Dong Xu, *Member, IEEE*

***Abstract*— Multiple kernel learning (MKL) has been proposed for kernel methods by learning the optimal kernel from a set of predefined base kernels. However, the traditional L_1MKL method often achieves worse results than the simplest method using the average of base kernels (i.e., average kernel) in some practical applications. In order to improve the effectiveness of MKL, this paper presents a novel soft margin perspective for MKL. Specifically, we introduce an additional slack variable called kernel slack variable to each quadratic constraint of MKL, which corresponds to one support vector machine model using a single base kernel. We first show that L_1MKL can be deemed as hard margin MKL, and then we propose a novel soft margin framework for MKL. Three commonly used loss functions, including the hinge loss, the square hinge loss, and the square loss, can be readily incorporated into this framework, leading to the new soft margin MKL objective functions. Many existing MKL methods can be shown as special cases under our soft margin framework. For example, the hinge loss soft margin MKL leads to a new box constraint for kernel combination coefficients. Using different hyper-parameter values for this formulation, we can inherently bridge the method using average kernel, L_1MKL, and the hinge loss soft margin MKL. The square hinge loss soft margin MKL unifies the family of elastic net constraint/regularizer based approaches; and the square loss soft margin MKL incorporates L_2MKL naturally. Moreover, we also develop efficient algorithms for solving both the hinge loss and square hinge loss soft margin MKL. Comprehensive experimental studies for various MKL algorithms on several benchmark data sets and two real world applications, including video action recognition and event recognition demonstrate that our proposed algorithms can efficiently achieve an effective yet sparse solution for MKL.**

***Index Terms*— Multiple kernel learning, support vector machines.**

I. Introduction

KERNEL methods, such as support vector machine (SVM) [1], [2], and kernel principal component analysis have shown to be powerful tools for numerous applications. However, their generalization performances are often decided by the choice of the kernel [3], [4], which represents the similarity between two data points. For kernel methods, a poor kernel can lead to impaired prediction performance, thus many works [5]–[10] have been proposed for learning the optimal kernel for kernel methods.

Manuscript received January 9, 2012; accepted April 6, 2012. Date of publication February 11, 2013; date of current version March 8, 2013. This work was supported by Multi-plAtform Game Innovation Centre (MAGIC) in Nanyang Technological University. MAGIC is funded by the Interactive Digital Media Programme Office (IDMPO) hosted by the Media Development Authority of Singapore. IDMPO was established in 2006 under the mandate of the National Research Foundation to deepen Singapore's research capabilities in interactive digital media (IDM), fuel innovation and shape the future of media.

The authors are with the School of Computer Engineering, Nanyang Technological University, 639798, Singapore (e-mail: xuxi0006@ntu.edu.sg; IvorTsang@ntu.edu.sg; dongxu@ntu.edu.sg).

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/TNNLS.2012.2237183

One of the pioneering works for kernel learning was proposed to simultaneously train the SVM classifier and learn the kernel matrix [9]. However, learning the general kernel matrix is a non-trivial task. The learning problem is generally formulated as a semi-definite programming problem, which suffers from the high computational cost even for one hundred training samples. Thus, this approach can be applicable for small scale data sets only. To reduce the computational cost, Lanckriet *et al*. [9] further assumed that the kernel is in the form of a linear combination of a set of predefined base kernels. Then the SVM classifier and the kernel combination coefficients are learned simultaneously, which is known as Multiple Kernel Learning (MKL). Since the proposed objective function has a simplex constraint for the kernel combination coefficients, it is also known as L_1MKL.

There are two major research directions for MKL methods, in which the first one focuses on the development of efficient learning algorithms, while the second one focuses on the improvement of the generalization performance. For the first direction, Bach *et al*. [11] employed a sequential minimization optimization (SMO) method for solving medium-scale MKL problems. Sonnenburg *et al*. [12] applied a semi-infinite linear programming (SILP) strategy by reusing the state-of-the-art SVM implementations for solving the subproblems inside the MKL optimization more efficiently, which makes MKL applicable to large scale data sets. The similar SILP strategy was also used by [13] for multiclass MKL problem. Following [12], the sub-gradient based method [14] and the level-method [15] have been proposed to further improve the convergence for solving MKL problems.

Although the optimization efficiency for L_1MKL has been improved in recent years, Cortes *et al*. [16] and Kloft *et al*. [17] showed that the L_1MKL formulation from [9] cannot achieve better prediction performance when compared with the simplest method using the average of base kernels (i.e., average kernel) for some real world applications. To improve the effectiveness, lots of new MKL formulations [16]–[32] have recently been proposed.

The simplex constraint for the traditional L_1MKL formulation usually yields a sparse solution. The recent works in [16] and [17] conjectured that such a sparsity constraint may omit some useful base kernels for the prediction. Thereafter, they introduced a L_2-norm constraint to replace the L_1-norm constraint in L_1MKL, leading to a non-sparse solution for the kernel combination coefficients. The L_2-norm constraint was further extended to the L_p-norm ($p>1$) constraint in [17]. Other MKL variants (e.g., [21], [33], and [34]) were proposed by removing the L_1-norm constraint, while directly adding one regularization term based on the L_1-norm, L_2-norm, or L_p-norm of the kernel combination coefficients to the objective function, which are indeed equivalent to the

formulation as in [17]. To further improve the efficiency of L_pMKL, Xu *et al*. [35] and Kloft *et al*. [17] proposed an analytical way to update the kernel combination coefficients. The SMO strategy was also employed in [34] to accelerate the optimization for the L_pMKL problem. In [22], a L_2-norm regularizer of the kernel combination coefficients is directly added to the objective function while keeping the simplex constraint fixed. Alternatively, Yang *et al*. [23] used the elastic net regularizer on the kernel combination coefficients as a constraint for MKL. Note that the elastic net regularizer in the block-norm form first appeared in [11] as a numerical tool for optimizing the L_1MKL and was further discussed in [25] with a variant form. Moreover, the extensions of elastic net regularizer for MKL in primal form with more general block-norm regularization were also discussed in [24] and [27]. However, it is still unclear why these regularizers can enhance the prediction performances for MKL.

To answer this question, in this paper, we first show that the traditional L_1MKL can be deemed as hard margin MKL, which only selects the base kernels with the minimum objective and throws away other useful base kernels. Then, we propose a novel soft margin perspective for MKL problems by starting from the dual of the traditional MKL method. The proposed soft margin framework for MKL is in analogy to the well-known soft margin SVM [36], which makes SVM more robust in real applications by introducing a slack variable for each of the training data. Similarly, with the introduction of a slack variable for each of the base kernels, we propose three novel soft margin MKL formulations, namely, the hinge loss soft margin MKL, the square hinge loss soft margin MKL, and the square loss soft margin MKL by using different loss functions.

The square loss soft margin MKL formulation incorporates L_2MKL naturally. The square hinge loss soft margin MKL connects a few MKL methods using the elastic net like regularizers or constraints. The hinge loss soft margin MKL leads to a totally new formulation, which bridges L_1MKL and the simplest approach based on the average kernel by using the different hyper-parameter values. These three cases reveal the connections between many independently proposed algorithms in the literature under our framework of soft margin MKL for the kernel learning, thus explain why the regularization, such as the L_2-norm or the elastic net like regularizer/constraint helps to improve the performance over L_1MKL in a new perspective.

In summary, the core contributions of this paper are listed in the following.

1) A novel soft margin framework for MKL is proposed. Particularly, a kernel slack variable is first introduced for each of the base kernels when learning the kernel. Three new MKL formulations, namely the hinge loss soft margin MKL, the square hinge loss soft margin MKL, and the square loss soft margin MKL are also developed under this framework.
2) A new block-wise coordinate descent algorithm based on the analytical updating rule for learning the kernel combination coefficients is developed to efficiently solve the new hinge loss soft margin MKL problem. With our proposed framework, a simplex projection method is also introduced to solve the square hinge loss soft margin MKL, leading to a more efficient optimization procedure when compared with the existing optimization algorithms for elastic net MKL.
3) Comprehensive experimental results on the benchmark data sets and two video applications, including real video event recognition and action recognition demonstrate the effectiveness and efficiency of our proposed soft margin MKL learning framework. Compared with L_2MKL (L_pMKL), the new hinge loss soft margin MKL and the square hinge loss soft margin MKL have much sparser solution for kernel combination coefficients; nevertheless, these two MKL models can achieve better generalization performance. This defends the effectiveness using sparse kernel combination coefficients in MKL.

This paper is organized as follows. In Section II, we first review ν-SVM and MKL. In Section III, a unified framework for soft margin MKL is proposed, and three novel soft margin MKL formulations are developed based on different loss functions for the kernel slack variables. New formulations for MKL are developed under our proposed soft margin MKL framework, and some existing formulations for MKL are revisited as the special cases under this framework. Then, a new block-wise coordinate descent algorithm for solving the hinge loss soft margin MKL and a simplex projection-based algorithm for solving the square hinge loss soft margin MKL are introduced in Section IV. Experimental results on the standard benchmark data sets and the YouTube and Event6 data sets from computer vision applications are shown in Section V. Finally, the conclusive remarks and the future work are presented in the last section.

II. Preliminaries and Related Works

Throughout the rest of this paper, we use the superscript $'$ to denote the transpose of a vector, and $\mathbf{0}, \mathbf{1} \in \mathbb{R}^l$ denote the zero vector and the vector of all ones, respectively. We also define $\boldsymbol{\alpha} \odot \mathbf{y}$ as the element-wise product between two vectors $\boldsymbol{\alpha}$ and $\mathbf{y}$. Moreover, $||\boldsymbol{\mu}||_p$ represents the L_p-norm of a vector $\boldsymbol{\mu}$, and the inequality $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_l]' \geq \mathbf{0}$ means that $\mu_i \geq 0$ for $i = 1, \ldots l$.

A. ν-SVM

Given a set of labeled training data $S = \{(\mathbf{x}_i, \mathrm{y}_i) | i = 1, \ldots, l\}$ sampled independently from $\mathcal{X} \times \mathcal{Y}$ with $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} = \{-1, +1\}$, a kernel matrix $\mathbf{K} \in \mathbb{R}^{l \times l}$ is usually constructed by using a mapping function $\phi(\mathbf{x})$ to map the data $\mathbf{x}$ from $\mathcal{X}$ to a reproducing kernel Hilbert space $\mathcal{H}$ such that $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)'\phi(\mathbf{x}_j)$. Then, ν-SVM [37]–[39] learns the decision function

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i \mathrm{y}_i k(\mathbf{x}, \mathbf{x}_i) + b \tag{1}$$

where α_i's are the coefficients associated with training samples, b is the bias of the decision function f. Let us define $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_l]'$ and $\mathbf{y} = [y_1, \ldots, y_l]'$. We minimize the model complexity $\|f\|_{\mathcal{H}}^2 = (\boldsymbol{\alpha} \odot \mathbf{y})'\mathbf{K}(\boldsymbol{\alpha} \odot \mathbf{y})$ and the training

errors (represented by slack variables ξ_i's) for the decision function f simultaneously, then we arrive at the corresponding optimization problem[1]

$$\min_{\boldsymbol{\alpha},b,\rho,\xi_i} \quad \frac{1}{2}(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}(\boldsymbol{\alpha}\odot\mathbf{y}) + C\sum_{i=1}^{l}\xi_i - \rho$$
$$\text{s. t.} \quad \mathrm{y}_i f(\mathbf{x}_i) \geq \rho - \xi_i,\ \xi_i \geq 0 \qquad \forall i = 1,\ldots,l \quad (2)$$

where $\rho/\|f\|_{\mathcal{H}}$ is the margin separation between two opposite classes and $C > 0$ is the regularization parameter. Note one can show that $C = 1/l\nu$, where ν is the lower bound of fraction of outliers [37], [38], [40]. By using the duality property, it is easy to show that the dual of the objective in (2) is

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} \mathrm{SVM}\{\mathbf{K},\boldsymbol{\alpha}\} \quad (3)$$

where $\mathrm{SVM}\{\mathbf{K},\boldsymbol{\alpha}\} = -1/2(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}(\boldsymbol{\alpha}\odot\mathbf{y})$ is the dual of the objective in SVM, and

$$\mathcal{A} = \{\boldsymbol{\alpha}|\boldsymbol{\alpha}'\mathbf{1} = 1, \boldsymbol{\alpha}'\mathbf{y} = 0, \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}\} \quad (4)$$

is the domain for $\boldsymbol{\alpha}$. From the Karush–Kuhn–Tucker (KKT) conditions of (2), one can show that the optimal solution $\boldsymbol{\alpha}^\star$ in the dual (3) is the same as that in the primal (2). Hence, for the given training set S and $\mathbf{K}$, the maximization of $\mathrm{SVM}\{\mathbf{K},\boldsymbol{\alpha}\}$ with respect to $\boldsymbol{\alpha} \in \mathcal{A}$ indeed gives the solution of the SVM classifier in (1).

B. L_1MKL

Now, we review MKL [11], [14]. With a set of given M base kernels $\mathcal{K} = \{\mathbf{K}_1,\ldots,\mathbf{K}_M\}$, L_1MKL tries to learn the optimal kernel combination coefficients and the decision function f simultaneously. When the ν-SVM model is used, the primal problem of L_1MKL with block-norm regularization is written as

$$\min_{f_m,b,\rho,\xi_i} \quad \frac{1}{2}\Big(\sum_{m=1}^{M}\|f_m\|_{\mathcal{H}_m}\Big)^2 + C\sum_{i=1}^{l}\xi_i - \rho$$
$$\text{s.t.} \quad \mathrm{y}_i\Big(\sum_{m=1}^{M} f_m(\mathbf{x}_i) + b\Big) \geq \rho - \xi_i, \qquad \xi_i \geq 0. \quad (5)$$

The first term in (5) is the group lasso regularizer [41], [42] to choose groups of nonlinear features with small model complexity $\|f_m\|_{\mathcal{H}_m}$, in which each group of nonlinear features is induced by using one base kernel. By using the Lagrangian method, the dual of L_1MKL is

$$\max_{\boldsymbol{\alpha}\in\mathcal{A},\tau} \ \tau : \mathrm{SVM}\{\mathbf{K}_m,\boldsymbol{\alpha}\} \geq \tau \qquad \forall m = 1,\ldots,M \quad (6)$$

where $\mathrm{SVM}\{\mathbf{K}_m,\boldsymbol{\alpha}\} = -1/2(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})$. Alternatively, the dual (6) of L_1MKL can also be written as

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}}\min_{\boldsymbol{\mu}\in\mathcal{M}} \sum_{m=1}^{M}\mu_m\mathrm{SVM}\{\mathbf{K}_m,\boldsymbol{\alpha}\} \quad (7)$$

where $\boldsymbol{\mu} = [\mu_1,\ldots,\mu_M]'$, μ_m is the coefficient to measure the importance of the mth base kernel, and $\mathcal{M} = \{\boldsymbol{\mu}|\mathbf{0} \leq \boldsymbol{\mu}, \sum_{m=1}^{M}\mu_m = 1\}$ is the domain for $\boldsymbol{\mu}$. Then the final classifier is given by

$$f(\mathbf{x}) = \sum_{i=1}^{l}\alpha_i\mathrm{y}_i\Big(\sum_{m=1}^{M}\mu_m k_m(\mathbf{x},\mathbf{x}_i)\Big) + b.$$

C. Hard Margin Perspective for L_1MKL

From the constraints in (6), each SVM dual objective is no less than τ. The "error" is not allowed for each SVM dual objective that is below τ, and only the base kernels with the objective equal to τ are retained. In other words, the objective of L_1MKL is essentially the same as $\max_{\boldsymbol{\alpha}\in\mathcal{A}}\min_{m=1,\ldots,M}\mathrm{SVM}\{\mathbf{K}_m,\boldsymbol{\alpha}\}$, which learns the SVM classifier by first choosing the model with the minimal objective. Ideally, only one base kernel will be chosen. Hence, L_1MKL usually gets a very sparse solution for the kernel combination coefficients, and some useful base kernels may not be used. Since the constraints in (6) "push" the SVM dual objectives as large as possible, the variable τ can be considered as the hard margin in L_1MKL, which reveals the hard margin property of L_1MKL in the margin point of view, and paves the way for soft margin MKL formulations in the sequel.

Remark that the non-sparse L_2MKL [16] was proposed by substituting the simplex constraint with the L_2-norm ball constraint $\boldsymbol{\mu} \in \{\boldsymbol{\mu}|\mathbf{0} \leq \boldsymbol{\mu}, \sum_{m=1}^{M}\mu_m^2 \leq 1\}$. L_pMKL [17] directly extends the MKL formulation in (7) by simply substituting the simplex constraint with the L_p-norm constraint $\boldsymbol{\mu} \in \{\boldsymbol{\mu}|\mathbf{0} \leq \boldsymbol{\mu}, \sum_{m=1}^{M}\mu_m^p \leq 1\}$ for $p > 1$. However, the kernel combination coefficients of these two models are always nonzeros, resulting in impaired prediction performance especially when many noisy or irrelevant base kernels are included. Therefore, how to remove noisy or irrelevant base kernels, and how to keep and emphasize the useful base kernels are the key issues for MKL methods.

III. Soft Margin Framework for MKL

MKL learns the classifier and the optimal kernel simultaneously with a set of predefined base kernels. These base kernels can be obtained by using any well-known kernel functions (e.g., Gaussian kernel function, polynomial kernel function, spline kernel function, etc.,) with different kernel parameters or specially designed by domain experts for the learning task. Moreover, in computer vision tasks, such as image classification or video event recognition, different types of features are extracted from lots of feature extraction methods (e.g., SITF [43], STIP [44], and HOG [45], etc.,). Even with the same feature extraction method, there are many parameters. Usually, each type of features can be used to form a base kernel [46] for representing images/videos. However, only some base kernels are informative for classification, and others may be irrelevant or even harmful. Recent studies show that the combination of several features can achieve better prediction performance for computer vision applications. However, L_1MKL usually chooses only one or few base kernels due to its hard margin property. On the other hand, we always obtain the dense solution for kernel combination coefficients by using L_2-MKL, L_p-MKL, and the simplest method based

[1]Although this formulation looks different from the original one proposed in [37], they are essentially equivalent according to [39] and [40].

on average kernel. Some noisy or irrelevant base kernels are inevitably included for prediction.

As pointed out in Section II-C, L_1MKL is indeed a hard margin MKL, which only selects the base kernels with the minimum objective. This could easily suffer from the over-fitting problem especially when some base kernels are formed by using noisy features. Recall that hard margin SVM assumes that the data of two opposite classes can always be separated with the hard margin, and the error is not allowed for training the model. However, to make SVM applicable for real applications, the slack variables were introduced to hard margin SVM in [36]. The introduction of the slack variables allows some training errors for the training data, thus alleviating over-fitting encountered in hard margin SVM.

Inspired by the success of slack variables for SVM [36], [47], in this section, we introduce the concept of the kernel slack variable for each of the base kernels, and develop a soft margin MKL framework, which is the counterpart to soft margin SVM. Herein, we have the following definition.

Definition 1: Given M base kernels $\mathcal{K} = \{\mathbf{K}_1, \ldots, \mathbf{K}_M\}$ for the training data $S = \{(\mathbf{x}_i, \mathrm{y}_i)|i = 1, \ldots, l\}$ sampled independently from $\mathcal{X} \times \mathcal{Y}$, we define kernel slack variable to be the difference of the target margin τ and the SVM dual objective SVM$\{\mathbf{K}_m, \boldsymbol{\alpha}\}$ for the given kernel $\mathbf{K}_m \in \mathcal{K}$ as

$$\zeta_m = \tau - \mathrm{SVM}\{\mathbf{K}_m, \boldsymbol{\alpha}\} \qquad \forall m = 1, \ldots, M. \tag{8}$$

Then, the loss introduced from the kernel slack variable is defined as

$$\mathrm{z}_m = \ell(\zeta_m) \qquad \forall m = 1, \ldots, M \tag{9}$$

where $\ell(\cdot)$ is any general loss function.

In the following, we mainly consider three loss functions, namely, the hinge loss (i.e., $\ell(\zeta_m) = \max(0, \zeta_m)$), the square hinge loss (i.e., $\ell(\zeta_m) = (\max(0, \zeta_m))^2$), and the square loss $\ell(\zeta_m) = \zeta_m^2$. Based on these loss functions on the kernel slack variables, we will present our proposed soft margin MKL formulations, respectively. Note that our soft margin MKL framework can cater for not only the abovementioned loss functions but also many other loss functions. These three loss functions are studied due to their simplicity and successful utilization in the standard soft margin SVM formulations.

A. Hinge Loss Soft Margin MKL

Based on the definition of the kernel slack variable for each base kernel, we are now ready to propose our soft margin MKL formulations. When the hinge loss is used for the kernel slack variables, we have the following objective function for the hinge loss soft margin MKL:

$$\begin{aligned} \min_{\tau, \boldsymbol{\alpha} \in \mathcal{A}, \zeta_m} \; & -\tau + \theta \sum_{m=1}^{M} \zeta_m \\ \text{s.t.} \; & \mathrm{SVM}\{\mathbf{K}_m, \boldsymbol{\alpha}\} \geq \tau - \zeta_m, \zeta_m \geq 0, \; m = 1, \ldots, M. \end{aligned} \tag{10}$$

The objective of the above hinge loss soft margin MKL is to maximize the margin τ while considering the "errors" from the given M base kernels. The parameter θ balances the contribution of the loss term represented by slack variables ζ_m's and the margin τ. To further discover the properties of the newly proposed hinge loss soft margin MKL formulation, we have the following proposition.

Proposition 2: The solution of the following optimization problem is also the solution of hinge loss soft margin MKL:

$$\min_{\boldsymbol{\mu} \in \mathcal{M}_1} \max_{\boldsymbol{\alpha} \in \mathcal{A}} \; \mathbf{J}(\boldsymbol{\mu}, \boldsymbol{\alpha}) \tag{11}$$

where the objective function is $\mathbf{J}(\boldsymbol{\mu}, \boldsymbol{\alpha}) = -1/2 \sum_{m=1}^{M} \mu_m (\boldsymbol{\alpha} \odot \mathbf{y})' \mathbf{K}_m (\boldsymbol{\alpha} \odot \mathbf{y})$ and $\mathcal{M}_1 = \{\boldsymbol{\mu} | \sum_{m=1}^{M} \mu_m = 1, \mathbf{0} \leq \boldsymbol{\mu} \leq \theta \mathbf{1}\}$.

The proof of this proposition is shown in the Appendix. Note that the objective function $\mathbf{J}(\boldsymbol{\mu}, \boldsymbol{\alpha})$ is the same as the one in the hard margin MKL formulation, and the difference is in the constraint for the coefficients $\boldsymbol{\mu}$. In hard margin MKL, the constraint for $\boldsymbol{\mu}$ is the simplex constraint $\boldsymbol{\mu} \in \mathcal{M} = \{\boldsymbol{\mu} | \sum_{m=1}^{M} \mu_m = 1, \mathbf{0} \leq \boldsymbol{\mu}\}$. In contrast, we have $\boldsymbol{\mu} \in \mathcal{M}_1$ in our new hinge loss soft margin MKL. This new constraint enforces the values of the $\boldsymbol{\mu}$ no more than the regularization parameter θ, which can prevent extreme large values of kernel combination coefficients frequently encountered in hard margin MKL. We similarly observe the counterpart property of this formulation from the relationship between the hard margin SVM [48] and the hinge loss soft margin SVM [36]. For the hard margin SVM, the boundary constraint for $\boldsymbol{\alpha}$ is given by $\mathbf{0} \leq \boldsymbol{\alpha}$, while the constraint is $\mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}$ for the hinge loss soft margin SVM. Note C in the soft margin SVM and θ in our soft margin MKL are the regularization parameters that balance the training error and the complexity of the model.

We also have the following interesting observations for this new objective function:

1) θ should be in the range $\{\theta | \theta \geq 1/M\}$, otherwise there is no solution to the proposed problem. This can be easily verified from the constraints in (11);
2) when $\theta = 1/M$, according to constraint $\mathcal{M}_1$ in (11), we can obtain the uniform solution for $\boldsymbol{\mu}$ (i.e., $\boldsymbol{\mu} = 1/M\mathbf{1}$);
3) when $\theta \geq 1$, the constraint $\mathcal{M}_1$ in (11) becomes the same as $\mathcal{M}$ in the hard margin MKL (i.e., L_1MKL [14]).

We clearly observe that the structural risk function is well controlled by introducing the penalty parameter θ, and the solution of the MKL problem can be changed by varying this parameter, which gives a novel perspective to the MKL problems. This objective function also bridges L_1MKL and the simple approach using average kernel by choosing different regularization parameter θ.

B. Square Hinge Loss Soft Margin MKL

When we define the loss function for the kernel slack variables as the square hinge loss, then we can arrive at the following objective function for the square hinge loss soft margin MKL:

$$\begin{aligned} \min_{\tau, \boldsymbol{\alpha} \in \mathcal{A}, \zeta_m} \; & -\tau + \frac{\theta}{2} \sum_{m=1}^{M} \zeta_m^2 \\ \text{s.t.} \; & \mathrm{SVM}\{\mathbf{K}_m, \boldsymbol{\alpha}\} \geq \tau - \zeta_m, \quad m = 1, \ldots, M. \end{aligned} \tag{12}$$

Similar to the hinge loss soft margin MKL, τ is the margin of the final classifier, and each SVM dual objective for the

base kernels is lower bounded by the difference between the margin τ and the kernel slack variable ζ_m. We also have the following proposition.

Proposition 3: The solution of the following optimization problem gives the solution of square hinge loss soft margin MKL:

$$\min_{\boldsymbol{\mu}\in\mathcal{M}_2}\max_{\boldsymbol{\alpha}\in\mathcal{A}}\ \mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha})+\frac{1}{2\theta}\sum_{m=1}^{M}\mu_m^2 \tag{13}$$

where the function $\mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha})$ is $\mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha}) = -1/2\sum_{m=1}^{M}\mu_m(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})$ and $\mathcal{M}_2=\{\boldsymbol{\mu}|\sum_{m=1}^{M}\mu_m=1,\mathbf{0}\le\boldsymbol{\mu}\}$.

The proof of this proposition is similar to that of Proposition 2, thus it is omitted. Compared with L_1MKL, this formulation shares the same simplex constraint for $\boldsymbol{\mu}$, but it has one more L_2-norm regularization term $1/2\theta\sum_{m=1}^{M}\mu_m^2$ for the coefficients in the objective function. The regularization parameter θ balances the regularization for $\boldsymbol{\mu}$ and the margin of the classifier $\mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha})$.

The relationship between hard margin MKL and the square hinge loss soft margin MKL is also similar to that between hard margin SVM and the square hinge loss soft margin SVM [36], where the constraint for $\boldsymbol{\alpha}$ remains the same while one more regularization term $\sum_{i=1}^{l}\alpha_i^2/2C$ is added in the objective function of the hard margin SVM formulation.

Note the simplex constraint is removed from L_2MKL to L_1MKL. In contrast, this formulation still has the simplex constraint. The previous work [22] has used such type of regularization by directly adding the L_2-norm regularization term for the kernel combination coefficients in the objective function of L_1MKL. To further discover the connections of our square hinge loss soft margin MKL with previous works, we have the following proposition.

Proposition 4: The primal form of the square hinge loss soft margin MKL is given as

$$\begin{aligned}\min_{\boldsymbol{\mu},f_m,b,\rho,\xi_i}\quad & \frac{1}{2}\sum_{m=1}^{M}\frac{\|f_m\|_{\mathcal{H}_m}^2}{\mu_m}+C\sum_{i=1}^{l}\xi_i-\rho+\frac{1}{2\theta}\sum_{m=1}^{M}\mu_m^2\\ \text{s.t.}\quad & \mathrm{y}_i\left(\sum_{m=1}^{M}f_m(\mathbf{x}_i)+b\right)\ge\rho-\xi_i,\xi_i\ge 0\\ & \sum_{m=1}^{M}\mu_m=1,\quad \mathbf{0}\le\boldsymbol{\mu}.\end{aligned} \tag{14}$$

Proof: With fixed $\boldsymbol{\mu}$, we can write the Lagrangian as $\mathcal{L} = 1/2\sum_{m=1}^{M}\|f_m\|_{\mathcal{H}_m}^2/\mu_m + C\sum_{i=1}^{l}\xi_i - \rho + 1/2\theta\sum_{m=1}^{M}\mu_m^2 - \sum_{i=1}^{l}\alpha_i\left(\mathrm{y}_i(\sum_{m=1}^{M}f_m(\mathbf{x}_i)+b)-\rho+\xi_i\right)-\sum_{i=1}^{l}\xi_i\beta_i$, where $\alpha_i\ge 0$, $\beta_i\ge 0$ are the Lagrange multipliers of the corresponding constraints. By setting the derivatives of the primal variables f_m, b, ρ, ξ_i to be zeros, we can get the corresponding KKT conditions. By replacing the primal variables in the Lagrangian with the KKT conditions, we can arrive at the min max optimization problem as shown in (13). Together with Proposition 3, we prove the proposition. ∎

In the primal form, the objective function can be denoted as $f=\arg\min_f\Omega(f)+R_{\text{emp}}(f)$, where $\Omega(f)$ is the regularization term for the functional f, $R_{\text{emp}}(f)$ is the empirical risk term from the given training samples. Specifically, for (14), we have $\Omega(f)=1/2\sum_{m=1}^{M}\|f_m\|_{\mathcal{H}_m}^2/\mu_m+1/2\theta\sum_{m=1}^{M}\mu_m^2$ with $\boldsymbol{\mu}\in\mathcal{M}_2$ and $R_{\text{emp}}(f)$ is the standard hinge loss from the training samples. In this formulation, we can see that the L_1-norm of the kernel combination coefficients is enforced in the constraint, and the L_2-norm of the kernel combination coefficients is penalized in $\Omega(f)$. Therefore, it is essentially the elastic net regularization [49] for MKL. In [23], the regularization is given as $\Omega(f)=1/2\sum_{m=1}^{M}\|f_m\|_{\mathcal{H}_m}^2/\mu_m$ under the elastic net constraint $\upsilon\sum_{m=1}^{M}\mu_m+(1-\upsilon)\sum_{m=1}^{M}\mu_m^2\le 1,\boldsymbol{\mu}\ge\mathbf{0}$. This can be regarded as a variant of (14) after considering the general conversion between Tikhonov regularization and Ivanov regularization as shown in [17, Th. 1].

Several existing works [11], [18], [24], [25], [27] have also been proposed for MKL with the (generalized) elastic net regularization in the primal form with the block-norm regularization, without explicitly containing the kernel combination coefficients $\boldsymbol{\mu}$. For instance, the work in [11] utilized the regularization term $\Omega(f)=\lambda/2\sum_{m=1}^{M}\|f_m\|_{\mathcal{H}_m}^2+1/2(\sum_{m=1}^{M}\|f_m\|_{\mathcal{H}_m})^2$ to facilitate the optimization of the L_1MKL problem. Interestingly, it can be shown that the primal form of square hinge loss soft margin MKL in (14) is equivalent to the formulations in [11], [18] and [25] by seeking the entire regularization path [50].

Note that, the square norm $\sum_{m=1}^{M}\zeta_m^2$ for the kernel slack variables ζ_m in (12) can be readily extended to a more general norm $\sum_{m=1}^{M}\zeta_m^{p/(p-1)}$ with $1<p<\infty$ in a similar fashion as from L_2MKL to L_pMKL (see Section III-C for more discussions). We can then obtain a more general $p/(p-1)$-hinge loss soft margin MKL as $\min_{\tau,\boldsymbol{\alpha}\in\mathcal{A},\zeta_m}-\tau+\theta/2\sum_{m=1}^{M}\zeta_m^{p/(p-1)}$ s.t. $\text{SVM}\{\mathbf{K}_m,\boldsymbol{\alpha}\}\ge\tau-\zeta_m,\zeta_m\ge 0, m=1,\ldots,M$. The extensions of elastic net MKL in the primal form with more general block-norm regularization are also discussed in [24] and [27], which can also be deemed as the soft margin MKL.

C. Square Loss Soft Margin MKL

By setting the margin $\tau=0$ in (12), the loss function for the kernel slack variables becomes the square loss, and we can get the following square loss soft margin MKL:

$$\begin{aligned}\min_{\boldsymbol{\alpha}\in\mathcal{A},\zeta_m}\quad & \frac{\theta}{2}\sum_{m=1}^{M}\zeta_m^2\\ \text{s.t.}\quad & -\text{SVM}\{\mathbf{K}_m,\boldsymbol{\alpha}\}\le\zeta_m,\qquad m=1,\ldots,M.\end{aligned} \tag{15}$$

The L_2MKL comes out naturally from (15) under our soft margin MKL framework according to the proposition.

Proposition 5: The solution of the following problem gives the solution of square loss soft margin MKL:

$$\min_{\boldsymbol{\mu}\in\mathcal{M}_3}\max_{\boldsymbol{\alpha}\in\mathcal{A}}\ \mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha}) \tag{16}$$

where the function $\mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha})$ is $\mathbf{J}(\boldsymbol{\mu},\boldsymbol{\alpha}) = -1/2\sum_{m=1}^{M}\mu_m(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})$ and $\mathcal{M}_3=\{\boldsymbol{\mu}|\sum_{m=1}^{M}\mu_m^2\le 1,\mathbf{0}\le\boldsymbol{\mu}\}$.

Proof: It can also be proven by introducing the Lagrangian multipliers, i.e., the dual variables μ_m for each of the inequality constraint, we can arrive at the following dual form: $\max_{\boldsymbol{\mu}\ge 0}\min_{\boldsymbol{\alpha}\in\mathcal{A}}-1/2\theta\sum_{m=1}^{M}\mu_m^2+1/2\sum_{m=1}^{M}\mu_m(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})$. By using Theorem 1 from [17], it

is easy to show that for one specific parameter θ the above optimization problem is equivalent to: $\min_{\boldsymbol{\mu}\geq 0,\boldsymbol{\mu}'\boldsymbol{\mu}\leq 1} \max_{\boldsymbol{\alpha}\in\mathcal{A}} -1/2\sum_{m=1}^{M}\mu_m(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})$, which is essentially L_2MKL [16], [17]. Thus, we conclude that our proposed soft margin MKL framework also incorporates L_2MKL as one special case. ■

By considering a more generalized norm on ζ_m beyond the L_2-norm, the formulation can be further extended to $\min_{\boldsymbol{\alpha}\in\mathcal{A},\zeta_m}\theta/2\sum_{m=1}^{M}\zeta_m^{p/(p-1)}$ s.t. $-\text{SVM}\{\mathbf{K}_m,\boldsymbol{\alpha}\}\leq\zeta_m$, $m=1,\ldots,M$, which can be similarly reformulated as L_pMKL ($p>1$) [17]. In general, L_pMKL [17] can be regarded as a special case of our soft margin MKL as well.

IV. Optimization for Soft Margin MKL

In this section, we propose new optimization algorithms for our proposed soft margin MKLs.

All the optimization problems can be changed to the min max optimization problem, so we adopt the alternating optimization approach, which was widely used in previous works [12], [14], to alternatively learn the kernel combination coefficients and the model parameter by leveraging the standard SVM implementations. Note that the recent works [17], [35] proposed a new analytical updating rule for L_pMKL by considering the special structure in the primal form of L_pMKL. This type of solution can avoid the time consuming procedure for searching the new updating point for the kernel combination coefficients. Although the convergence when using $p=1$ is not proven, the stable convergence to the optimal solution was experimentally observed in [17] and [35]. Besides, the similar analytical updating rule was also adopted in [51]. In this paper, we also propose a new analytical solution for updating the kernel combination coefficients based on the structure of our new objective function for the hinge loss soft margin MKL. For the square hinge loss soft margin MKL, a simplex projection method is proposed.

A. Block-Wise Coordinate Descent Algorithm for Solving the Primal Hinge Loss Soft Margin MKL

We have the following proposition.

Proposition 6: The following problem is the primal form for hinge loss soft margin MKL:

$$\min_{\boldsymbol{\mu}\in\mathcal{M}_1,f_m,b,\rho,\xi_i} \frac{1}{2}\sum_{m=1}^{M}\frac{\|f_m\|_{\mathcal{H}_m}^2}{\mu_m}+C\sum_{i=1}^{l}\xi_i-\rho$$
$$\text{s.t.}\quad \mathrm{y}_i\left(\sum_{m=1}^{M}f_m(\mathbf{x}_i)+b\right)\geq\rho-\xi_i,\quad \xi_i\geq 0. \quad (17)$$

Proof: The Lagrangian can be written as

$$\begin{aligned}\mathcal{L}=&\frac{1}{2}\sum_{m=1}^{M}\frac{\|f_m\|_{\mathcal{H}_m}^2}{\mu_m}+C\sum_{i=1}^{l}\xi_i-\rho\\&-\sum_{i=1}^{l}\alpha_i\left(\mathrm{y}_i(\sum_{m=1}^{M}f_m(\mathbf{x}_i)+b)-\rho+\xi_i\right)-\sum_{i=1}^{l}\xi_i\beta_i\\&-\sum_{m=1}^{M}\mu_m\eta_m-\sum_{m=1}^{M}\zeta_m(\theta-\mu_m)+\tau(\sum_{m=1}^{M}\mu_m-1) \quad (18)\end{aligned}$$

where $\alpha_i\geq 0$, $\beta_i\geq 0$, $\eta_m\geq 0$, $\zeta_m\geq 0$, and τ are the Lagrange multipliers for the corresponding constraints.

By setting the derivatives of the Lagrangian in (18) with respect to the primal variables f_m, b, ρ, ξ_i, and μ_m to be zeros, and substituting the primal variables back into the Lagrangian according to the corresponding KKT conditions, we have:

$$\begin{aligned}\max_{\tau,\boldsymbol{\alpha}\in\mathcal{A},\zeta_m}\quad&-\tau-\theta\sum_{m=1}^{M}\zeta_m\\\text{s.t.}\quad&-\frac{1}{2}(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})\geq-\tau-\zeta_m\\&\zeta_m\geq 0,\quad m=1,\ldots,M. \quad (19)\end{aligned}$$

By multiplying -1 to the objective function in (19), it is converted into a minimization problem. Substituting τ with $-\tau$, we arrive at the same formulation as the hinge loss soft margin MKL in (10). Thus, we complete the proof. ■

In the primal form as in (6), we have the box constraint for $\boldsymbol{\mu}$, i.e., $\boldsymbol{\mu}\in\mathcal{M}_1=\{\boldsymbol{\mu}|\sum_{m=1}^{M}\mu_m=1,\mathbf{0}\leq\boldsymbol{\mu}\leq\theta\mathbf{1}\}$. The work in [52] proposed a family of structured sparsity to improve the lasso for linear regression problem. Specifically, a box constraint is directly enforced on the unknown regression variables to enforce the structured sparsity. By simplifying our model to the linear case without the group structure [41], [42], we could include [52] as a special case.

The primal problem in (17) is convex in the objective function [14] and linear in the constraints, thus it is convex. It can be solved by using the block-wise coordinate descent algorithm [17], [35].

1) Fix $\boldsymbol{\mu}$, Update f_m, b, ρ, ξ_i: With a fixed $\boldsymbol{\mu}$, the optimization problem in (17) becomes a standard maximum margin SVM problem, which can be equivalently reformulated as a standard Quadratic Programming (QP) problem with respect to $\boldsymbol{\alpha}$ as shown in (20), and many efficient QP solvers can be readily used to obtain the optimal $\boldsymbol{\alpha}$

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}}\quad-\frac{1}{2}\sum_{m=1}^{M}\mu_m(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y}). \quad (20)$$

After obtaining the optimal $\boldsymbol{\alpha}$, the primal variables f_m, b, ρ, ξ_i can be recovered accordingly.

2) Fix f_m, b, ρ, ξ_i, Update $\boldsymbol{\mu}$: With fixed f_m, b, ρ, ξ_i, the optimization problem in (17) reduces to the following convex programming problem:

$$\min_{\boldsymbol{\mu}\in\mathcal{M}_1}\quad\sum_{m=1}^{M}\frac{a_m}{\mu_m} \quad (21)$$

with $a_m=1/2\|f_m\|_{\mathcal{H}_m}^2=\frac{1}{2}\mu_m^2(\boldsymbol{\alpha}\odot\mathbf{y})'\mathbf{K}_m(\boldsymbol{\alpha}\odot\mathbf{y})$.

The remaining problem is how to efficiently solve the subproblem (21). Let us suppose all the base kernels are positive definite, and then we have $a_m>0$ for $m=1,\ldots,M$. Without the loss of generality, we also assume that a_m has been sorted such that $a_1\geq a_2\geq\cdots\geq a_M$. Inspired by the Lagrangian multipliers method [53] used for simplex projection, we introduce the Lagrangian multipliers λ, η_ms,

and ζ_ms for the constraints in (21). Then we can get the following Lagrangian:

$$\mathcal{L} = \sum_{m=1}^{M} \frac{a_m}{\mu_m} - \sum_{m=1}^{M} \mu_m \eta_m - \sum_{m=1}^{M} \zeta_m(\theta - \mu_m) + \lambda\left(\sum_{m=1}^{M} \mu_m - 1\right). \tag{22}$$

Setting the derivative of $\mathcal{L}$ with respect to μ_m to be zeros, we have the following KKT condition:

$$-\frac{a_m}{\mu_m^2} - \eta_m + \zeta_m + \lambda = 0 \tag{23}$$

with the complementary KKT conditions $\mu_m \eta_m = 0$, $\zeta_m(\theta - \mu_m) = 0$, and $\lambda(\sum_{m=1}^{M} \mu_m - 1) = 0$.

Thus, for $0 < \mu_m < \theta$, we have

$$-\frac{a_m}{\mu_m^2} + \lambda = 0, \text{ or } \mu_m = \sqrt{\frac{a_m}{\lambda}}. \tag{24}$$

If $a_m > 0$, we have $\mu_m > 0$. Thus, for the case that all the a_m's are larger than 0, the constraint $0 \leq \mu_m$ can be replaced by $0 < \mu_m$. If we know ω, the number of elements in $\boldsymbol{\mu}$ whose value strictly equals to θ, the solution of the above problem can be directly obtained as:

$$\mu_m = \begin{cases} \theta, & m \leq \omega \\ \frac{(1-\omega\theta)\sqrt{a_m}}{\sum_{p=\omega+1}^{M} \sqrt{a_p}}, & m > \omega. \end{cases} \tag{25}$$

We have the following two lemmas to obtain the solution for the problem in (21).

Lemma 7: Let $\boldsymbol{\mu}^*$ be the optimal solution to problem (21), and suppose $a_p > a_q$ for any two given indices p, q $\in \{1, \ldots, M\}$. If $\mu_q^* = \theta$, then we have $\mu_p^* = \theta$.

Proof: Suppose that $\boldsymbol{\mu}^*$ is the optimal solution to the problem in (21), and $\mu_q^* = \theta$. If using proof by contradiction, we have $\mu_p^* < \theta$. Let $\tilde{\boldsymbol{\mu}}$ be another vector whose elements have the same value with $\boldsymbol{\mu}^*$ except that $\tilde{\mu}_p = \mu_q^*$ and $\tilde{\mu}_q = \mu_p^*$. Then, we observe that $\tilde{\boldsymbol{\mu}}$ satisfies all the constraints in (21). Thus, $\sum_{m=1}^{M} a_m/\mu_m^* - \sum_{m=1}^{M} a_m/\tilde{\mu}_m = a_p/\mu_p^* + a_q/\mu_q^* - a_p/\tilde{\mu}_p - a_q/\tilde{\mu}_q = (a_p - a_q)(1/\mu_p^* - 1/\theta) > 0$. So we have $\sum_{m=1}^{M} a_m/\mu_m^* > \sum_{m=1}^{M} a_m/\tilde{\mu}_m$, which contradicts with the assumption that $\boldsymbol{\mu}^*$ is the optimal solution to (21). So the original assumption is incorrect and thus we complete the proof. ■

Lemma 8: Let $\boldsymbol{\mu}^*$ be the optimal solution to the problem in (21), and suppose that $a_1 \geq a_2 \geq \cdots \geq a_M$. Then ω, the number of elements whose value strictly equals to θ in $\boldsymbol{\mu}^*$, is

$$\min\left\{p \in \{0, 1, \ldots, M-1\} \middle| \frac{\sqrt{a_{p+1}}(1 - p\theta)}{\sum_{m=p+1}^{M} \sqrt{a_m}} < \theta\right\}.$$

The proof is similar with that of Lemma 7 by using the proof by contradiction and thus it is omitted here.

3) Whole Optimization Procedure: Based on the above derivations, we can easily develop the whole optimization procedure for the hinge loss soft margin MKL, and the detailed block-wise coordinate descent algorithm is shown in Algorithm 1.

Algorithm 1 Procedure of the Block-Wise Coordinate Descent Algorithm for Hinge Loss Soft Margin MKL

1: Initialize $\boldsymbol{\mu}^1$.
2: $t = 1$
3: **while** stop criteria is not reached **do**
4: Obtain $\boldsymbol{\alpha}^t$ by solving the subproblem in (20) using the standard QP solver with $\boldsymbol{\mu}^t$
5: Calculate a_m and update $\boldsymbol{\mu}^{t+1}$ by solving the subproblem in (21)
6: $t = t + 1$
7: **end while**

B. Simplex Projection Method for Solving the Square Hinge Loss Soft Margin MKL

For solving the square hinge loss soft margin MKL, we directly solve the problem in (13). With a fixed $\boldsymbol{\mu}$, the optimization problem with respect to $\boldsymbol{\alpha}$ is a standard QP problem, which can be optimized by using the QP solver.

With a fixed $\boldsymbol{\alpha}$, the projected gradient descent-based algorithm is used to update the kernel combination coefficients. Following [6], the gradient $\mathbf{p}^t$ of the optimization problem in (13) with respect to $\boldsymbol{\mu}$ can be calculated as:

$$\mathrm{p}_m = -h_m + \frac{1}{\theta}\mu_m, \qquad m = 1, \ldots, M \tag{26}$$

where $h_m = 1/2(\boldsymbol{\alpha} \odot \mathbf{y})' \mathbf{K}_m (\boldsymbol{\alpha} \odot \mathbf{y})$.

Then, the coefficient $\boldsymbol{\mu}$ is updated by using the coefficients $\boldsymbol{\mu}^t$ at the current iteration, namely

$$\boldsymbol{\mu}_{\text{sub}}^{\star} = \Pi_{\mathcal{M}_2}(\boldsymbol{\mu}^t - \eta_t \mathbf{p}^t) \tag{27}$$

where $\Pi_{\mathcal{M}_2}(\cdot)$ is the simplex projection operation and η_t is the updating step size determined by the standard line search strategy. The simplex projection operation is a standard QP problem, which can also be solved by using the general QP solver. However, due to the special simplex constraint for $\boldsymbol{\mu}$, the efficient simplex projection method in [54] is used in this paper. The detailed optimization procedure is shown in Algorithm 2.

V. Experiments

In this section, we first evaluate different MKL algorithms on the benchmark data sets. Then we show the experimental studies on two real computer vision applications (i.e., video action recognition and video event recognition).

A. Comparison Algorithms

We evaluate the following algorithms.

1) *AveKernel:* We use average combination of the base kernels. Specifically, the kernel combination coefficients is given by $\boldsymbol{\mu} = 1/M\mathbf{1}$, then the maximum margin classifier is learnt by SVM.
2) *SimpleMKL [14]:* The classifier and the kernel combination coefficients are optimized by solving the L_1MKL problem as in (5).
3) *L_2MKL [16], [17]:* The classifier and the kernel combination coefficients are optimized under the constraint $||\boldsymbol{\mu}||_2 \leq 1$.

Algorithm 2 Procedure of the Iterative Approach for Square Hinge Loss Soft Margin MKL

1: Initialize $\boldsymbol{\mu}^1$.
2: $t = 1$.
3: **while** stop criteria is not reached **do**
4: Obtain $\boldsymbol{\alpha}^t$ by solving the subproblem in (20) using the standard QP solver with $\boldsymbol{\mu}^t$
5: Calculate $\boldsymbol{\mu}^{\star}_{\text{sub}}$ that can reduce the objective function value for the problem in (13)
6: Update $\boldsymbol{\mu}^{t+1} = \boldsymbol{\mu}^{\star}_{\text{sub}}$
7: $t = t + 1$
8: **end while**

4) *L_pMKL [17]:* The classifier and the kernel combination coefficients are optimized under the constraint $||\boldsymbol{\mu}||_{\text{p}} \leq 1$ with $p \geq 1$.
5) *SGMKL [23]:* The sparse generalized multiple kernel learning as in [23], where the constraint for the kernel combination coefficients is the elastic net constraint, i.e., $\upsilon||\boldsymbol{\mu}||_1 + (1-\upsilon)||\boldsymbol{\mu}||_2 \leq 1$ with $0 \leq \upsilon \leq 1$.
6) *SM1MKL:* Our proposed hinge loss soft margin MKL, in which the classifier and the kernel combination coefficients are optimized by solving the hinge loss soft margin MKL problem.
7) *SM2MKL:* The square hinge loss soft margin MKL, in which the classifier and the kernel combination coefficients are optimized by solving the square hinge loss soft margin MKL problem.

To be consistent with previous works [14], [17], [23], the experiments for different MKL algorithms are all based on the C-SVC formulation as used in [14], and the SVM QP problem is solved by using the LibSVM C-SVC QP solver.[2] For the SimpleMKL codes downloaded from the web,[3] we additionally change the SVM solver in their implementation with the LibSVM QP solver. For L_pMKL, the implementation is available in Shogun toolbox [55]; however, we implement the algorithm by using the analytical updating rule for the kernel combination coefficients exactly as in [17] and [35] for better utilization of the LibSVM QP solver for fair comparison. For SGMKL [23], we download their MATLAB implementation,[4] and replace the SVM QP solver with the LibSVM QP solver, and also use Mosek[5] to solve the subproblem for updating the kernel combination coefficients in their implementation.

The SVM regularization parameter C is set in the range of $\{0.01, 0.1, 1, 10, 100\}$ for all the algorithms on all the data sets in the following experiments. One more model parameter p is introduced for L_pMKL, υ is introduced for SGMKL, and θ is introduced for SM1MKL and SM2MKL. These parameters are set as follows:

1) for L_pMKL, $p \in \{1, 32/31, 16/15, 8/7, 4/3, 2, 3, \infty\}$;
2) for SGMKL, υ is in the range of $\{0, 0.1, 0.2, \ldots, 1\}$;
3) for SM1MKL, θ is set to be $1/\upsilon M$, where υ is a ratio parameter from $\{1/M, 0.1, 0.2, \ldots, 1\}$;
4) for SM2MKL, θ is in the range of $\{10^{-5}, \ldots, 10^4, 10^5\}$.

Then all the algorithms have multiple sets of parameters, and the optimal parameters are determined by using five-fold cross validation on the training set.

B. Experiments on Benchmark Data Sets

We first evaluate our proposed algorithms on some benchmark data sets. The experiments are conducted on seven publicly available data sets.[6]

1) Experimental Settings: For the construction of base kernels on these benchmark data sets, we follow the method in [14] by designing the base kernels in the following manner:

1) Gaussian kernels using ten different bandwidth parameters from $\{2^{-3}, 2^{-2}, \ldots, 2^6\}$ by using all the variables and each single variable;
2) polynomial kernels with the degree from $\{1, 2, 3\}$ by using all the variables and each single variable.

We randomly partition the data set into two parts, namely 70% for training and the rest 30% for testing. For each partition, all the dimensions of samples in the training set are normalized to have zero mean and unit variance, while the samples in the test set are normalized accordingly. The experiments are then repeated 10 times, and the mean accuracy and the standard deviation on each test set are reported for comparison.

2) Experimental Results: Table I shows the performance comparison of different algorithms, which demonstrates the effectiveness of our proposed MKL formulations when compared with the other MKL formulations. The average rank for each algorithm is calculated in the last row of Table I. The average rank of SM2MKL is 2.00, and the average rank of SM1MKL is 2.57. So, SM1MKL and SM2MKL achieve similar performances. SimpleMKL follows SM1MKL and achieves the third position. In terms of the rank, SGMKL and L_2MKL are a bit worse than SimpleMKL, and AveKernel is the worst. The results show that AveKernel and L_pMKL cannot outperform L_1MKL, probably because of redundant base kernels constructed in this setting. In terms of the loss functions defined on the kernel slack variables, the square loss is usually more sensitive to outliers than (square) hinge loss, thus the generalization ability of L_2MKL (L_pMKL) may be limited when compared with the hinge loss soft margin MKL (SM1MKL) and square hinge loss soft margin MKL (SM2MKL).

The average numbers of selected base kernels for different MKL formulations are shown in Fig. 1. We observe that SimpleMKL (L_1MKL) selects the smallest number of base kernels on most of the data sets, and L_2MKL selects almost all the base kernels, leading to dense solutions. The AveKernel selects all the base kernels. SGMKL, SM1MKL, and SM2MKL generally obtain sparser solutions when compared

[2] Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
[3] http://asi.insa-rouen.fr/enseignants/~arakotom/code/mklindex.html.
[4] Available at: http://appsrv.cse.cuhk.edu.hk/~hqyang/doku.php?id=gmkl.
[5] Available at: http://www.mosek.com/.
[6] Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets and http://www.fml.tuebingen.mpg.de/Members/raetsch/benchmark. The data sets include Heart, Diabetes, Australian, Ionosphere, Ringnorm, Banana, and FlareSolar.

TABLE I

PERFORMANCE EVALUATION [MEAN CLASSIFICATION ACCURACY (%) ± STANDARD DEVIATION] FOR DIFFERENT ALGORITHMS ON THE BENCHMARK DATA SETS. THE NUMBER IN THE PARENTHESIS SHOWS THE RANK OF EACH ALGORITHM IN TERMS OF THE MEAN CLASSIFICATION ACCURACY

	AveKernel	SimpleMKL	L_2MKL	L_pMKL	SGMKL	SM1MKL	SM2MKL
Heart	81.60 ± 2.76 (7)	81.98 ± 3.20 (5)	82.47 ± 3.23 (1)	81.85 ± 3.08 (6)	82.10 ± 3.15 (3)	81.60 ± 4.21 (4)	82.47 ± 3.28 (1)
Diabetes	75.22 ± 4.02 (7)	75.30 ± 3.35 (6)	75.91 ± 2.83 (4)	75.61 ± 2.71 (5)	76.00 ± 2.92 (3)	76.35 ± 2.79 (1)	76.26 ± 2.94 (2)
Australian	85.94 ± 2.24 (2)	85.12 ± 1.82 (4)	85.07 ± 1.84 (6)	85.27 ± 1.77 (3)	84.78 ± 1.80 (7)	86.23 ± 1.94 (1)	85.12 ± 1.82 (4)
Ionosphere	90.29 ± 4.01 (7)	91.81 ± 1.92 (3)	91.71 ± 2.70 (4)	91.33 ± 2.64 (5)	91.90 ± 2.39 (2)	91.33 ± 2.82 (5)	92.10 ± 2.38 (1)
Ringnorm	95.42 ± 2.01 (7)	98.25 ± 1.07 (1)	96.67 ± 1.47 (6)	97.67 ± 1.10 (4)	97.67 ± 1.35 (4)	98.00 ± 1.12 (2)	97.75 ± 1.36 (3)
Banana	73.00 ± 5.79 (7)	90.08 ± 2.95 (1)	88.58 ± 2.72 (6)	89.50 ± 2.43 (4)	89.50 ± 2.43 (4)	89.75 ± 2.39 (3)	90.08 ± 2.68 (1)
FlareSolar	67.04 ± 3.45 (7)	67.59 ± 3.99 (6)	68.64 ± 2.96 (1)	68.59 ± 2.93 (2)	67.94 ± 3.32 (5)	68.59 ± 2.93 (2)	68.59 ± 2.93 (2)
Average Rank	6.28	3.71	4.00	4.14	4.00	2.57	2.00

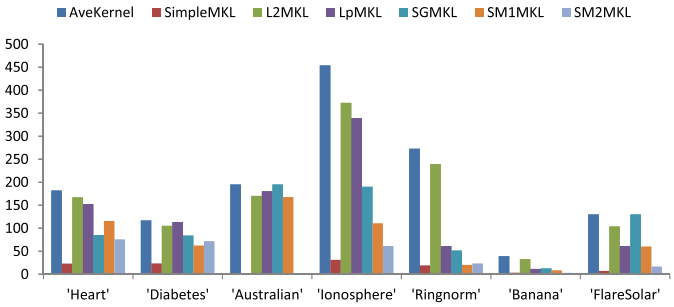


Fig. 1. The average number of selected base kernels for each of the methods on the benchmark data sets.

to L_pMKL, which demonstrates that whether the solution is sparse or non-sparse should not be the main factor for the effectiveness of MKL methods.

Table II shows the mean CPU time costs for training each of the model on the training set. The average rank for each algorithm is also listed in the last row of the table. We can observe that generally AveKernel using the single average kernel is the fastest since SVM model is trained only once for prediction. For MKL algorithms, SGMKL and SimpleMKL are comparable to each other but they are less efficient when compared with other methods. The L_pMKL is more efficient due to the analytical solution for the kernel combination coefficients [17], [35]. For SM1MKL and SM2MKL, the training is very efficient thanks to the analytical updating rule for SM1MKL and the efficient simplex projection procedure for SM2MKL. Moreover, SM2MKL is much faster than SGMKL due to the utilization of the simplex projection method in our optimization process.

C. Measuring the Impact of Noisy Base Kernels for Different MKL Algorithms

From the soft margin point of view, we also analyze the characteristics for the MKL methods by using the regularization on the kernel slack variables. Specifically, some MKL formulations are more sensitive to noisy base kernels. To verify it, we compare AveKernel with other MKL methods using different loss functions on the kernel slack variables, including L_1MKL (hard margin), L_2MKL (square loss), SM1MKL (hinge loss), and SM2MKL (square hinge loss). We use the first round of experiments for "Diabetes" from the benchmark data set to show the results of different algorithms when using noisy base kernels. The feature vector is augmented with $r * d$ dimensions of randomly generated features, where d is the dimension of the original feature vector and r is the percentage of the augmented noisy features in the range of $\{0, 0.2, 0.4, \ldots, 1.2\}$.

Fig. 2 shows the accuracy of different MKL methods when using different levels of the noisy features for "Diabetes." We can clearly observe that AveKernel can achieve good results when the base kernels are clean. But when there are more noisy base kernels, the performance of AveKernel becomes much worse than the other algorithms. Moreover, in this experiment, the hinge loss for the kernel slack variables is the most robust loss function when there are strong noisy base kernels.

D. Experiments on YouTube for Action Recognition

In computer vision applications, many features can be extracted for the image or video data sets, and the best results are usually obtained by fusing multiple types of features. However, some features may only be suitable for some specific applications and may even be harmful for other applications. Thus, how to fuse or combine different features is an important problem for computer vision applications. In the following, we will show the effectiveness of MKL algorithms for action recognition [56].

1) Experimental Setting: We evaluate different MKL algorithms on the YouTube data set, which contains 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. The data set contains a total number of 1168 video sequences. We follow the pre-defined partitions as in [56], where the whole data set is partitioned to 25 folds. In order to compare the generalization ability of the different MKL formulations, we further choose 20 folds for training and use the remaining five folds for testing. The 20 training folds are also used to determine the parameters for all the algorithms.

TABLE II

TRAINING TIME EVALUATION [MEAN CPU TIME (SECOND) ± STANDARD DEVIATION] FOR DIFFERENT ALGORITHMS ON THE BENCHMARK DATA SETS. THE NUMBER IN THE PARENTHESIS SHOWS THE RANK OF EACH ALGORITHM IN TERMS OF THE MEAN CPU TIME

	AveKernel	SimpleMKL	L_2MKL	L_pMKL	SGMKL	SM1MKL	SM2MKL
Heart	0.1938 ± 0.039 (1)	19.32 ± 11.71 (6)	9.023 ± 1.957 (5)	4.273 ± 2.533 (3)	129.9 ± 92.55 (7)	1.928 ± 3.517 (2)	8.245 ± 4.498 (4)
Diabetes	1.075 ± 0.5016 (1)	410.8 ± 89.47 (6)	154.5 ± 17.50 (4)	83.62 ± 63.18 (3)	1206 ± 694.8 (7)	39.41 ± 22.79 (2)	240.6 ± 88.93 (5)
Australian	1.134 ± 0.144 (1)	194.6 ± 25.60 (6)	159.6 ± 76.61 (5)	82.94 ± 63.39 (4)	897.3 ± 535.5 (7)	23.58 ± 18.77 (3)	22.96 ± 3.542 (2)
Ionosphere	0.5656 ± 0.2193 (1)	146.2 ± 48.85 (6)	40.92 ± 12.64 (5)	26.16 ± 24.25 (3)	618.8 ± 529.0 (7)	14.58 ± 11.60 (2)	31.90 ± 20.41 (4)
Ringnorm	0.606 ± 0.292 (1)	297.3 ± 10.4.7 (6)	69.18 ± 40.48 (2)	277.5 ± 288.4 (5)	1035 ± 547.8 (7)	165.6 ± 142.8 (3)	239.7 ± 36.16 (4)
Banana	0.1734 ± 0.1051 (1)	15.07 ± 5.127 (4)	15.71 ± 1.656 (5)	49.74 ± 37.16 (7)	16.85 ± 6.792 (6)	11.56 ± 5.480 (2)	15.23 ± 17.55 (3)
FlareSolar	0.7609 ± 0.2173 (1)	2243 ± 6244 (7)	152.5 ± 24.54 (4)	382.7 ± 414.4 (5)	649.5 ± 253.7 (6)	81.03 ± 129.8 (3)	58.69 ± 38.95 (2)
Average Rank	1.00	5.86	4.28	4.28	6.71	2.43	3.43

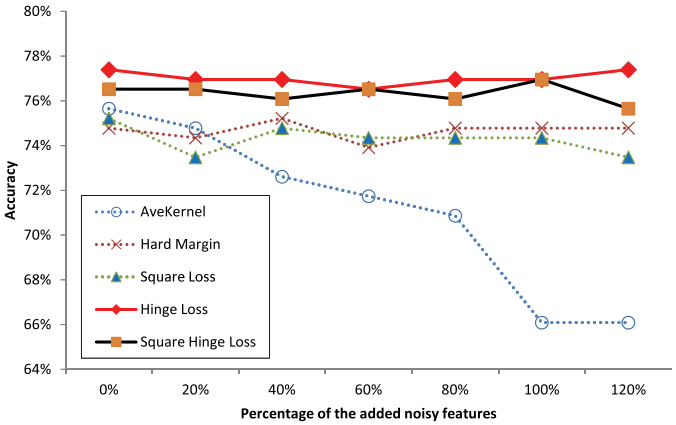


Fig. 2. Performance of MKL when using different loss functions on kernel slack variables with respect to the level of noisy features for "Diabetes."

Four types of features, namely, Trajectory, HOG, HOF, and MBH [57], are extracted from each of the video sequences. Then the base kernels are constructed from each of the four types of features by using the χ^2-kernel. The kernel mapping function is given as $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\gamma D(\mathbf{x}_i, \mathbf{x}_j)\big)$, where $D(\mathbf{x}_i, \mathbf{x}_j)$ is the χ^2 distance between any two videos for each type of features, and $\gamma = 1/A4^{n-1}$ with A being the mean value of the χ^2 distances between all the training samples. The kernel parameter n is from $\{-1, -0.5, \ldots, 1\}$, thus a total number of 20 base kernels are used in the experiment.

For the performance evaluation, we use the non-interpolated average precision (AP), which has been widely used as the performance metric for image and video retrieval applications. It corresponds to the multipoint average precision values of a precision-recall curve and incorporates the effect of recall. Mean AP (MAP) means the mean of APs over all the 11 semantic action concepts.

2) Experimental Results: We report the MAP, the mean number of selected kernels (MNK), and the mean training CPU time (MTT) in Table III on this data set. The results are based on the mean of the 11 evaluated concepts. We can observe that the MAP of SimpleMKL is 87.47% and it is worse than AveKernel (88.39%), which indicates that L_1MKL (SimpleMKL) may throw away some useful base kernels due to the hard margin property. We also observe that all the soft margin formulations L_2MKL, L_pMKL, SGMKL, SM1MKL, and SM2MKL achieve better results when compared with AveKernel and L_1MKL (SimpleMKL) and SM1MKL is the best in terms of MAP.

As shown in Table III, we also observe that AveKernel and L_2MKL select all the 20 base kernels, and L_1MKL selects the smallest number of base kernels, (i.e., 3.09 base kernels on average). SM1MKL, SM2MKL, and SGMKL select fewer base kernels than AveKernel and L_pMKL. Again, we conclude that whether the solution is sparse or non-sparse is not the key factor for the effectiveness of the MKL methods even though our new formulations can obtain sparser solutions compared with L_pMKL.

We also find that the training time of AveKernel is much faster than other MKL algorithms, and SGMKL and SimpleMKL are slower when compared with other MKL algorithms, such as L_2MKL, SM1MKL, and SM2MKL, which have similar training time. L_pMKL becomes slower in this experiment due to the smaller p value obtained from cross validation. SM2MKL is much faster than SGMKL due to the efficient simplex projection method proposed under our soft margin framework. In general, our new SM1MKL outperforms other MKL learning algorithms in terms of both efficiency and effectiveness on this data set.

E. Experiments on Event6 for Video Event Recognition

1) Experimental Setting: We evaluate different algorithms on another real world Event6 data set [58]. This data set contains 1101 videos, in which 924 videos are used as the training set and the remaining 177 are used as the test set. Six events (i.e., "wedding," "birthday," "picnic," "parade," "show," and "sports") are used for performance evaluation. Two types of local features (i.e., "STIP," "SIFT") are extracted from each of the video sequences, and then K-means is used to build the visual vocabularies for each of the local features. The spatial pyramid is also used to construct the final feature vector, in which two levels are used. Thus, four types of distances from two types of features and two pyramid levels are calculated as suggested in [58].

For any given distance $\mathbf{D}$, four types of kernels are used as the base kernels: Gaussian kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\gamma D^2(\mathbf{x}_i, \mathbf{x}_j)\big)$), Laplacian kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\sqrt{\gamma} D(\mathbf{x}_i, \mathbf{x}_j)\big)$), inverse square distance (ISD) kernel

TABLE III
PERFORMANCE EVALUATION FOR DIFFERENT ALGORITHMS ON THE YOUTUBE DATA SET IN TERMS OF THE MEAN AVERAGE PRECISION (MAP %), THE MNK, AND THE MTT OVER 11 CONCEPTS ON THE TEST SET

	AveKernel	SimpleMKL	L_2MKL	L_pMKL	SGMKL	SM1MKL	SM2MKL
MAP (%)	88.39	87.47	88.66	89.21	89.20	89.26	89.09
MNK	20	3.09	20	12.91	8.64	9.09	8.54
MTT (Second)	1.04	57.77	36.78	123.8	191.8	36.48	45.37

TABLE IV
PERFORMANCE EVALUATION FOR DIFFERENT ALGORITHMS ON THE VIDEO EVENT DATA SET IN TERMS OF THE MEAN AVERAGE PRECISION (MAP %), THE MNK, AND THE MTT OVER SIX EVENTS ON THE TEST SET

	AveKernel	SimpleMKL	L_2MKL	L_pMKL	SGMKL	SM1MKL	SM2MKL
MAP (%)	44.33	47.14	53.34	53.49	53.81	54.84	53.98
MNK	80.00	3.63	68.00	60.50	60.53	53.83	61.77
MTT (Second)	2.297	542.9	261.1	396.4	1639	410.1	367.1

(i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = 1/(\gamma D^2(\mathbf{x}_i, \mathbf{x}_j) + 1)$), and inverse distance (ID) kernel (i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = 1/(\sqrt{\gamma} D(\mathbf{x}_i, \mathbf{x}_j) + 1)$), where $D(\mathbf{x}_i, \mathbf{x}_j)$ denotes the distance between two samples $\mathbf{x}_i$ and $\mathbf{x}_j$. We set $\gamma = 4^{n-1}\gamma_0$, where $n \in \{-2, -1, \ldots, 2\}$ and $\gamma_0 = 1/A$ with A being the mean value of square distances between all the training samples, thus a total number of 80 base kernels are constructed from the four types of distances. Please refer to [58] for more details on the features and the kernels.

2) Experimental Results: We report the MAP, the mean number of the selected base kernels (MNK), and the mean training CPU time (MTT) over all the six events in Table IV. The MAP for AveKernel is only 44.33%, which is the worst on this data set. A possible explanation is the poor performance of the STIP features as shown in [58]. L_1MKL (SimpleMKL) can improve the MAP to 47.14%, and L_pMKL can further improve the performance to 53.49%. SM2MKL and SGMKL achieve comparable performances. However, our newly proposed SM1MKL achieves the best MAP 54.84%. We observe that AveKernel can be much worse when the base kernels are noisy. While SimpleMKL can discard the noisy base kernels, it may also discard some useful base kernels due to the hard margin property. Although L_pMKL improves the performance, the generalized square loss is usually more sensitive to the outliers than the (square) hinge loss, thus it cannot achieve the best result. The hinge loss for the kernel slack variables should be the most robust one on this data set, thus SM1MKL achieves the best results when compared with other algorithms.

In terms of the MNK, AveKernel selects all the base kernels, and L_2MKL still selects as more as possible base kernels, and L_pMKL selects fewer base kernels due to a smaller value p determined from cross-validation. SGMKL, SM1MKL, and SM2MKL can also select fewer base kernels when compared with AveKernel and L_2MKL. As for the training time, our new algorithms are still very efficient. SM1MKL is faster than SGMKL and SimpleMKL and it is comparable to L_pMKL. Again, we observe that the utilization of simplex projection method for SM2MKL significantly improves the efficiency, so SM2MKL is much faster than SGMKL, which again demonstrates it is beneficial to use our soft margin MKL framework to develop new efficient optimization method for improving the efficiency of square hinge loss soft margin MKL.

VI. CONCLUSION

In this paper, we have proposed a novel soft margin framework for MKL by introducing the kernel slack variables for kernel learning. Based on the formulation, we then propose the hinge loss soft margin MKL, the square hinge loss soft margin MKL, and the square loss soft margin MKL. We additionally discover their connections with previous MKL methods and compare different MKL formulations in terms of the robustness of loss functions defined on the kernel slack variables. Comprehensive experiments have been conducted on the benchmark data sets and the YouTube and Event6 data sets from computer vision applications. The experimental results demonstrate the effectiveness of our proposed framework.

In the future, we plan to analyze the theoretical bounds for the proposed soft margin MKLs and study their extensions to multi-class settings as well as investigate how to extend our MKL techniques for solving the more general ambiguity problem in [59] and [60].

APPENDIX

PROOF OF PROPOSITION 2

We can rewrite (10) in the following form:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathcal{A},\tau,\zeta_m} \quad & -\tau + \theta \sum_{m=1}^{M} \zeta_m \\ \text{s. t.} \quad & -\frac{1}{2}(\boldsymbol{\alpha} \odot \mathbf{y})' \mathbf{K}_m (\boldsymbol{\alpha} \odot \mathbf{y}) \geq \tau - \zeta_m \\ & \zeta_m \geq 0 \quad m = 1, \ldots, M \end{aligned} \tag{28}$$

where the domain for $\boldsymbol{\alpha}$ is $\mathcal{A} = \{\boldsymbol{\alpha} | \boldsymbol{\alpha}'\mathbf{1} = 1, \boldsymbol{\alpha}'\mathbf{y} = 0, \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}\}$.

The Lagrangian of (28) is

$$\mathcal{L} = -\tau + \theta \sum_{m=1}^{M} \zeta_m - \sum_{m=1}^{M} \mathrm{z}_m \zeta_m + \sum_{m=1}^{M} \mu_m \Big(\frac{1}{2} (\boldsymbol{\alpha} \odot \mathbf{y})' \mathbf{K}_m (\boldsymbol{\alpha} \odot \mathbf{y}) + \tau - \zeta_m \Big) \quad (29)$$

where $\mu_m \geq 0$ and $\mathrm{z}_m \geq 0$ are the non-negative Lagrangian multipliers for the inequalities in (28). Setting the derivatives of the Lagrangian with respect to the primal variables τ and ζ_m as zeros, we arrive at

$$\sum_{m=1}^{M} \mu_m = 1 \quad (30)$$

$$\theta - \mu_m - \mathrm{z}_m = 0, \quad m = 1, \ldots, M. \quad (31)$$

Substituting the (30) and (31) back into the Lagrangian, we finish the proof.

REFERENCES

[1] D. Geebelen, J. A. K. Suykens, and J. Vandewalle, "Reducing the number of support vectors of SVM classifiers using the smoothed separable case approximation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 4, pp. 682–688, Apr. 2012.

[2] F. Cai and V. Cherkassky, "Generalized SMO algorithm for SVM-based multitask learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 6, pp. 997–1003, Jun. 2012.

[3] K. R. Müller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

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

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

[6] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 131–159, 2002.

[7] C. S. Ong, A. J. Smola, and R. C. Williamson, "Learning the kernel with hyperkernels," *J. Mach. Learn. Res.*, vol. 6, pp. 1043–1071, Jul. 2005.

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

[9] G. R. G. Lanckriet, N. Cristianini, P. L. 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.

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

[11] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 1–9.

[12] 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.

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

[14] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2512, Nov. 2008.

[15] 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*, 2008, pp. 1825–1832.

[16] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 regularization for learning kernels," in *Proc. Conf. Uncertainty Artif. Intell.*, 2009, pp. 109–116.

[17] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "L_p-norm multiple kernel learning," *J. Mach. Learn. Res.*, vol. 12, pp. 953–997, Mar. 2011.

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

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

[20] F. Orabona, J. Luo, and B. Caputo, "Online-batch strongly convex multikernel learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 787–794.

[21] M. Varma and B. R. Babu, "More generality in efficient multiple kernel learning," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 1065–1072.

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

[23] H. Yang, Z. Xu, J. Ye, I. King, and M. R. Lyu, "Efficient sparse generalized multiple kernel learning," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 433–446, Mar. 2011.

[24] M. Kloft, U. Rückert, and P. L. Bartlett, "A unifying view of multiple kernel learning," in *Proc. ECML/PKDD (2)*, 2010, pp. 66–81.

[25] T. Suzuki and R. Tomioka, "Spicymkl: A fast algorithm for multiple kernel learning with thousands of kernels," *Mach. Learn.*, vol. 85, nos. 1–2, pp. 77–108, 2011.

[26] C. Cortes, M. Mohri, and A. Rostamizadeh, "Two-stage learning kernel algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 239–246.

[27] F. Orabona and J. Luo, "Ultrafast optimization algorithm for sparse multikernel learning," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 249–256.

[28] F. Orabona, J. Luo, and B. Caputo, "Multikernel learning with online-batch optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 227–253, Feb. 2012.

[29] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[30] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1667–1680, Sep. 2012.

[31] X. Wu, D. Xu, L. Duan, and J. Luo, "Action recognition using context and appearance distribution features," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, Jun. 2011, pp. 489–496.

[32] S. Yan, X. Xu, D. Xu, S. Lin, and X. Li, "Beyond spatial pyramids: A new feature extraction framework with dense spatial sampling for image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 473–487.

[33] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[34] S. V. N. Vishwanathan, Z. Sun, N. Theera-Ampornpunt, and M. Varma, "Multiple kernel learning and the SMO algorithm," in *Advances in Neural Information Processing Systems*, 2010, pp. 2361–2369.

[35] 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.*, 2010, pp. 1175–1182.

[36] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[37] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Comput.*, vol. 12, no. 5, pp. 1207–1245, 2000.

[38] C.-J. Lin, "A formal analysis of stopping criteria of decomposition methods for support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1045–1052, May 2002.

[39] C.-C. Chang and C.-J. Lin, "Training ν-support vector classifiers: Theory and algorithms," *Neural Comput.*, vol. 13, no. 9, pp. 2119–2147, 2001.

[40] C.-C. Chung and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

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

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

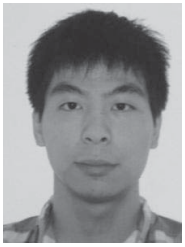
[43] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[44] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, no. 3, pp. 107–123, 2005.

[45] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, Jun. 2005, pp. 886–893.

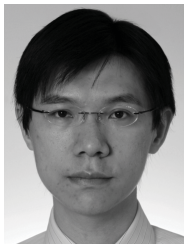
[46] P. V. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2009, pp. 221–228.

[47] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, MA: Cambridge Univ. Press, 2000.
[48] B. E. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. COLT*, 1992, pp. 1–8.
[49] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Royal Stat. Soc., B*, vol. 67, no. 2, pp. 301–320, 2005.
[50] F. R. Bach, R. Thibaux, and M. I. Jordan, "Computing regularization paths for learning multiple kernels," in *Proc. NIPS Conf.*, 2004, pp. 1–8.
[51] M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy, "Composite kernel learning," *Mach. Learn.*, vol. 79, no. 2, pp. 73–103, 2010.
[52] C. A. Micchelli, J. Morales, and M. Pontil, "A family of penalty functions for structured sparsity," in *Proc. NIPS Conf.*, 2010, pp. 1612–1623.
[53] S. Shalev-Shwartz and Y. Singer, "Efficient learning of label ranking by soft projections onto polyhedra," *J. Mach. Learn. Res.*, vol. 7, pp. 1567–1599, Jul. 2006.
[54] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 272–279.
[55] S. Sonnenburg, G. Rätsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. D. Bona, A. Binder, C. Gehl, and V. Franc, "The SHOGUN machine learning toolbox," *J. Mach. Learn. Res.*, vol. 11, pp. 1799–1802, Dec. 2010.
[56] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, 2009, pp. 1–8.
[57] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, Jun. 2011, pp. 3169–3176.
[58] L. Duan, D. Xu, I. W. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recogn.*, Jul. 2010, pp. 1959–1966.
[59] X. Xu, I. W.-H. Tsang, and D. Xu, "Handling ambiguity via input-output kernel learning," in *Proc. IEEE Int. Conf. Data Mining*, 2012, pp. 725–734.
[60] W. Li, L. Duan, I. W.-H. Tsang, and D. Xu, "Co-labeling: A new multiview learning approach for ambiguous problems," in *Proc. IEEE Int. Conf. Data Mining*, 2012, pp. 419–428.



Xinxing Xu received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2009. He is currently pursuing the Ph.D. degree with the School of Computer Engineering, Nanyang Technological University, Singapore.

His current research interests include kernel learning methods as well as their applications to computer vision.



Ivor W. Tsang received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2007.

He is currently an Assistant Professor with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. He is the Deputy Director of the Center for Computational Intelligence, NTU.

Dr. Tsang was the recipient of the Natural Science Award (Class II) in 2008, China, which recognized his contributions to kernel methods, the prestigious IEEE Transactions on Neural Networks Outstanding 2004 Paper Award in 2006, and a number of best paper awards and honors from reputable international conferences, including the Best Student Paper Award at CVPR 2010, the Best Paper Award at ICTAI 2011, and the Best Poster Award Honorable Mention at ACML 2012. He was also the recipient of the Microsoft Fellowship 2005, and ECCV 2012 Outstanding Reviewer Award.

Dong Xu (M'07) received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2001 and 2005, respectively.

He was with Microsoft Research Asia, Beijing, and the Chinese University of Hong Kong, Shatin, Hong Kong, for more than two years. He was a Post-Doctoral Research Scientist with Columbia University, New York, NY, for one year. He is currently an Associate Professor with Nanyang Technological University, Singapore. His current research interests include computer vision, statistical learning, and multimedia content analysis.

Dr. Xu was the co-author of a paper that won the Best Student Paper Award in the prestigious IEEE International Conference on Computer Vision and Pattern Recognition in 2010.