IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 1, 2020, accepted January 30, 2020, date of publication February 18, 2020, date of current version March 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974903

ℓ_q—Norm Sample-Adaptive Multiple Kernel Learning

QIANG WANG[1], XINWANG LIU[2], AND JIAQING XU[1]
[1]College of Computer, National University of Defense Technology, Changsha 410073, China
[2]Department of Computer Science, College of Computer, National University of Defense Technology, Changsha 410073, China

Corresponding author: Qiang Wang (qiangwang@nudt.edu.cn)

This work is supported by the National Key Research and Development Program of China under No. 2018YFB2202300.

ABSTRACT Existing multiple kernel learning (MKL) algorithms indiscriminately apply the same set of kernel combination weights to all samples by pre-specifying a group of base kernels. Sample-adaptive MKL learning (SAMKL) overcomes this limitation by adaptively switching on/off the base kernels with respect to each sample. However, it restricts to solving MKL problems with pre-specified kernels. And, the formulation of existing SAMKL falls to an ℓ_1-norm MKL which is not flexible. To allow for robust kernel mixtures that generalize well in practical applications, we extend SAMKL to the arbitrary norm and apply it to image classification. In this paper, we formulate a closed-form solution for optimizing the kernel weights based on the equivalence between group-lasso and MKL, and derive an efficient ℓ_q-norm ($q \geq 1$ and denoting the ℓ_q-norm of kernel weights) SAMKL algorithm. The cutting plane method is used to solve this margin maximization problem. Besides, we propose a framework for solving MKL problems in image classification. Experimental results on multiple data sets show the promising performance of the proposed solution compared with other competitive methods.

INDEX TERMS Multiple kernel learning, cutting plane, deep learning.

I. INTRODUCTION

Kernel methods [1], have been an attractive topic in machine learning [2]–[6]. They introduce nonlinearity to the decision function by mapping the original features to a higher dimensional space. Due to their descent computational complexity, high usability and solid mathematical foundation, they have been widely used for classification [7], clustering [8] and regression [9] tasks in numerous applications, such as pattern recognition [10] and object detection [11].

In many practical applications, data has multiple representations or data sources, which usually contain complementary and compatible information. For example, in the classification task of Oxford Flower17 [12], flowers can be represented by different features, such as color, shape, and texture. It is difficult for us to design an appropriate kernel function for this task. We have multiple kernel candidates because multiple features are derived from images or because different kernel functions (e.g., polynomial, RBF) are used to measure the similarities between samples for given feature representation [13]. It is of vital importance to find the optimal combination of these kernels for this task. This is exactly what multiple kernel learning (MKL) needs to solve.

Recently, MKL has attracted much attention. It not only provides an efficient way to learn an optimal kernel but also builds an elegant framework to integrate complementary information with distinct base kernels extracted from multiple heterogeneous data sources or features. Research on MKL has been flourishing and can be roughly categorized into two aspects. One is to improve the computational efficiency of MKL. Using Semi-Definite Programming or alternating approaches, these methods try to make MKL capable of handling large-scale learning tasks [4]–[6], [14]–[18]. The other one is to improve the classification performance of MKL by exploring the possible combination ways of base kernels [19]–[25].

While MKL has been studied extensively, it is restricted to learning a global combination for the whole input space. Due to the characteristic of data distribution, the set of kernels that are important for discrimination may vary from sample to sample. In the sense that all input samples share the same kernel weights, ignoring the fact that it may be beneficial

The associate editor coordinating the review of this manuscript and approving it for publication was Nikhil Padhi.

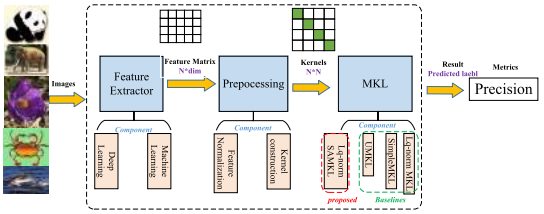


FIGURE 1. Framework of MKL.

to assign different samples with different kernel weights. For instance, the data distribution of different images for a given feature representation (e.g., color) may have a big difference [13]. Therefore, the kernels derived from different data distribution may have different effects on different images. Thus, introducing local learning into MKL for a localized sample-specific combination would achieve better performances and various local MKL methods have been proposed [24], [26]–[31].

Nevertheless, the base kernels may be irregularly corrupted across different samples. For instance, when the input features of samples are contaminated by noise, the kernel combination weights predicted via a parametric model will not be accurate anymore. To handle this problem, Liu *et al.* [3] proposed a sample-adaptive MKL algorithm (SAMKL), in which base kernels are allowed to be adaptively switched on/off with respect to each sample. Latent binary variables are introduced to each base kernel to decide whether a particular kernel should operate on a particular sample or not. The kernel combination weights and the latent variables are jointly optimized alternately via margin maximization principle.

However, previous MKL methods usually use pre-specified kernels to improve classification performance. In this work, we propose a framework to construct kernels from features extracted by ourselves and derive an efficient ℓ_q-norm SAMKL problem. The cutting plane method [32] is used to optimize the objective function. Extensive experimental results on multiple data sets exhibit the promising performance of the proposed technique compared with other competitive methods.

Fig. 1 illustrates the framework of our work. Given an image data set, we extract features using traditional machine learning methods (HOG [33], SIFT [34], LBP [35], and etc.), and deep learning methods [36]. Then, we construct Gaussian kernels on the normalized feature matrices. After that, we perform MKL algorithms on these computed kernels to get the predicted labels for classification performance evaluation. Precision is used as the metric to evaluate the classification performance in our experiments.

The contributions of this study can be summarized as follows:

(a) An efficient ℓ_q-norm SAMKL is proposed which is much more flexible compared with SAMKL.

(b) The cutting plane method is used to solve this margin maximization problem. By exhibiting a trick on constraints of the objective function, we can achieve comparable computational complexity of SAMKL.

(c) Comprehensive experimental results on multiple data sets demonstrate the effectiveness and efficiency of the proposed ℓ_q-norm SAMKL.

The rest of this paper is organized as follows. Section 2 provides a brief overview of related work. Section 3 presents the proposed optimization methods for SAMKL. Section 4 shows the extensive experimental results. Finally, Section 5 concludes this paper.

II. BACKGROUND

Given a labeled dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} = \{\mathbf{x} : \mathbf{x}_i \in \mathbb{R}^d, \forall i\}$ denotes the collection of n training samples that are in a d-dimensional space and $\mathcal{Y} = \{\mathbf{y} : y_i \in \{1, \cdots, c\}, \forall i\}$ denotes the label of the corresponding sample in $\mathcal{X}$ with c denoting the number of sample classes. Considering a feature mapping $\phi : \mathbb{R}^d \to \mathcal{H}$, in the new Hilbert space a hyperplane can be written as $f(\mathbf{x}) = \boldsymbol{\omega}^\top \phi(\mathbf{x}) + b$. In the existing MKL framework, it involves a linear combination of m pre-defined base kernels $\{\mathbf{K}_p(.,.)\}_{p=1}^m$ with each element of the p-th kernel calculated by $k_{ij} = \phi_p(\mathbf{x}_i)^\top \phi_p(\mathbf{x}_j)$, where $\phi(\cdot) = [\phi_1^\top(\cdot), \phi_2^\top(\cdot), \cdots, \phi_m^\top(\cdot)]^\top$. Then, the hyperplane (i.e., discriminant function, linear classifier) can be written as [37]

$$\begin{aligned} & f(\mathbf{x}) = \sum_{p=1}^{m} \gamma_p \boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}) + b \\ & w.r.t.\ \boldsymbol{\omega}_p \in \mathbb{R}^T, \quad b \in \mathbb{R} \\ & s.t. \sum_{p=1}^{m} \gamma_p = 1, \quad \gamma_p \geq 0,\ \forall p \end{aligned} \tag{1}$$

where $\boldsymbol{\omega}_p$ is the weights of the original feature space corresponding to the p-th feature mapping, γ_p is the weight of the classifier induced by the p-th feature mapping, and b is the bias term of the classifier. MKL learns the classifier by maximizing the margin between classes via solving the following quadratic optimization problem.

$$\begin{aligned} & \min_{\{\boldsymbol{\omega}_p\}_{p=1}^m, b, \boldsymbol{\xi}} \frac{1}{2} \left(\sum_{p=1}^{m} ||\boldsymbol{\omega}_p||_{\mathcal{H}_p} \right)^2 + C \sum_{i=1}^{n} \xi_i \\ & w.r.t.\ \boldsymbol{\omega}_p \in \mathbb{R}^T, \quad b \in \mathbb{R} \\ & s.t.\ y_i \left(\sum_{p=1}^{m} \boldsymbol{\omega}_p^\top \phi_p(\mathbf{x}_i) + b \right) \geq 1 - \xi_i, \quad \xi_i \geq 0,\ \forall i, \end{aligned} \tag{2}$$

where $\mathcal{H}_p$ represents the feature space corresponding to the p-th base kernel, T is the dimensionality of the feature space given by $\phi_p(\cdot)$, ξ is the slack variables, and C is a regularization parameter.

According to [38], the problem in Eq. 2 is proven to be equivalent to the one in the following equation.

$$\begin{aligned} & \min_{\{\boldsymbol{\omega}_p\}_{p=1}^m, b, \boldsymbol{\xi}, \gamma \in \delta} \frac{1}{2} \sum_{p=1}^{m} \frac{||\boldsymbol{\omega}_p||_{\mathcal{H}_p}^2}{\gamma_p} + C \sum_{i=1}^{n} \xi_i \\ & w.r.t.\ \boldsymbol{\omega}_p \in \mathbb{R}^T, b \in \mathbb{R} \end{aligned}$$

$$s.t.\ y_i\left(\sum_{p=1}^{m}\boldsymbol{\omega}_p^\top\phi_p(\mathbf{x}_i)+b\right)\geq 1-\xi_i,\quad \xi_i\geq 0,\ \forall i,$$
$$\sum_{p=1}^{m}\gamma_p=1,\quad \gamma_p\geq 0,\ \forall p \tag{3}$$

where γ_p is the combined weight of the p-th base kernel and controls the smoothness of the kernel function. The primal optimization problem of Eq. 3 is convex and differentiable [6] and it is equivalent to solve the min-max optimization problem of the following dual problem [14].

$$\min_{\boldsymbol{\gamma}}\max_{\boldsymbol{\alpha}}-\frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})^\top\left(\sum_{p=1}^{m}\gamma_p\mathbf{K}_p\right)(\boldsymbol{\alpha}\circ\mathbf{y})+\boldsymbol{\alpha}^\top\mathbf{1}$$
$$s.t.\ \boldsymbol{\alpha}^\top\mathbf{y}=0,\quad 0\leq\alpha_i\leq C,\ \forall i,$$
$$\sum_{p=1}^{m}\gamma_p=1,\quad \gamma_p\geq 0,\ \forall p \tag{4}$$

where $\alpha=[\alpha_1,\alpha_2,\cdots,\alpha_n]^\top$ are the Lagrange multipliers, $\mathbf{1}$ is a vector of all ones and $(\boldsymbol{\alpha}\circ\mathbf{y})$ denotes the component-wise multiplication between $\boldsymbol{\alpha}$ and $\mathbf{y}$. It should be noted when $\boldsymbol{\gamma}\in\Delta$ lies in a simplex, i.e., $\Delta=\{\boldsymbol{\gamma}:\sum_{p=1}^{m}\gamma_p=1,\gamma_p\geq 0,\forall p\}$, it is a ℓ_1-norm of kernel weights. Correspondingly, when $\Delta=\{\boldsymbol{\gamma}:||\gamma||_p\leq 1,\gamma_p\geq 0,\forall p\}$, it is an ℓ_p-norm of kernel weights and the resulting model is called ℓ_p-MKL [4]. After obtaining the optimal $\boldsymbol{\alpha}$, b and $\boldsymbol{\gamma}$, we get $\boldsymbol{\omega}_p=\sum_{i=1}^{n}\alpha_i y_i\phi_p(x)$. The discriminant function can be formulated as

$$f(\mathbf{x})=\sum_{i=1}^{n}\alpha_i y_i\left(\sum_{p=1}^{m}\gamma_p\mathbf{K}_p(\mathbf{x}_i,\mathbf{x})\right)+b. \tag{5}$$

The problem in Eq. 4 is usually solved by performing an alternating optimization strategy which consists of solving a canonical SVM optimization problem with given $\boldsymbol{\gamma}$ and updating $\boldsymbol{\gamma}$ using the gradient calculated via Eq. 6 with $\boldsymbol{\alpha}$ found in the first step [39]. This MKL framework is called simpleMKL [6].

$$\frac{\partial J(\boldsymbol{\gamma})}{\partial\gamma_p}=-\frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})^\top\frac{\partial\gamma_p\mathbf{K}_p}{\partial\gamma_p}(\boldsymbol{\alpha}\circ\mathbf{y})=-\frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})^\top\mathbf{K}_p(\boldsymbol{\alpha}\circ\mathbf{y}) \tag{6}$$

Most of existing MKL algorithms are restricted to learning a global combination of kernel weights for the pre-specified kernel matrices. That is, all input samples share the same kernel weights, ignoring the fact that samples may have the underlying local structure, which in turn degrades the MKL performance. Therefore, it is reasonable to assign different samples with different kernel weights by suppressing kernels that are irrelevant for learning tasks and selecting kernels that are beneficial for MKL tasks. Based on this idea, many localized MKL algorithms have been proposed. Xu *et al.* [4] discussed the connection between multiple kernel learning and the group-LASSO regularizer and proposed an efficient ℓ_p-norm MKL algorithm. The algorithm generalized the formulation of MKL to ℓ_q-norm MKL by replacing $\sum_{p=1}^{m}\gamma_p\leq 1$ with $\sum_{p=1}^{m}\gamma_p^q\leq 1$ where $q>0$. This proposed algorithm can be applied to the entire family of ℓ_q models, besides which the kernel weights can be calculated by a closed-form formulation without employing other commercial optimization software.

However, the base kernels may be irregularly corrupted across samples. To improve this situation, Liu *et al.* [3] proposed a sample-adaptive MKL (SAMKL) algorithm to localized MKL, where base kernels can be adaptively switched on/off at the example level. The optimization problem of the proposed SAMKL is as follows,

$$\min_{\{\boldsymbol{\omega}_p,\mathbf{h}_i\},\boldsymbol{\xi},b,\boldsymbol{\gamma}\in\Delta}\left(\frac{1}{2}\sum_{p=1}^{m}\frac{||\boldsymbol{\omega}_p||_{\mathcal{H}_p}^2}{\gamma_p}+C\sum_{i=1}^{n}\xi_i+C'\sum_{i=1}^{n}||\mathbf{h}_i||_1\right)$$
$$s.t.\ \frac{y_i}{\tau_i}\left(\sum_{p=1}^{m}h_{ip}\boldsymbol{\omega}_p^\top\phi_p(\mathbf{x}_i)+b\right)\geq 1-\xi_i,\quad \xi_i\geq 0,\forall i$$
$$||\mathbf{h}_i-\mathbf{h}_0||_1\leq m_0,\mathbf{h}_i\in\{0,1\}^m,\quad \forall i$$
$$\tau_i=\frac{\sum_{p=1}^{m}h_{ip}||\boldsymbol{\omega}_p||_{\mathcal{H}_p}}{\sum_{p=1}^{m}||\boldsymbol{\omega}_p||_{\mathcal{H}_p}},\quad \forall i$$
$$\sum_{p=1}^{m}\gamma_p=1,\quad \gamma_p\geq 0,\ \forall p \tag{7}$$

where latent binary variables $\mathbf{h}_i=[h_{i1},h_{i2},\cdots,h_{im}]^\top\in\{1,0\}^m$ with respect to $\mathbf{x}_i$ are introduced to decide whether a particular kernel should operate on a particular point or not. Specifically, $h_{ip}=1$ means that the p-th feature mapping $\phi_p(\cdot)$ is beneficial for the classification of the i-th sample $\mathbf{x}_i$, while $h_{ip}=0$ indicates the opposite. The optimization problem of Eq. 7 can be solved by considering a two-stage alternating optimization which consists of solving an MKL problem for different subspaces simultaneously with fixed values of the latent variables and secondly obtaining new values of the latent variables by running an integer program solver. Note that each step of the iteration here solves costly operations (an MKL solver and an integer problem solver) in comparison with the SVM solvers in the other approaches [26].

As can be seen in Eq. 7, the combination of kernel weights falls into the ℓ_1-MKL model. Following our previous analysis, we improve this situation by formulating a closed-form solution for optimizing the kernel weights and derive an efficient ℓ_q-norm SAMKL algorithm. Besides, the cutting plane method is used to solve this margin maximization problem, and the computational complexity of our algorithm is equivalent to that of Eq. 7.

III. SAMPLE-ADAPTIVE MULTIPLE KERNEL LEARNING

This section introduces the proposed ℓ_q-norm SAMKL problem. First, the problem formulation of ℓ_q-norm SAMKL is given. Second, cutting plane based methods are used to

optimize the objective function of our proposed problem. A discussion of our work is then provided.

A. PROBLEM FORMULATION

For MKL problems with latent variables, we want to learn a prediction rule of the form

$$f(\mathbf{x}) = \max_{(y,h)\in\mathcal{Y}\times\mathcal{H}} \left[\boldsymbol{\omega}^\top \Psi(\mathbf{x}; y, h)\right] \tag{8}$$

where $\Psi(\mathbf{x}; y, h)$ is a joint feature mapping on data $\mathcal{X}$, labels $\mathcal{Y}$ and latent variables $\mathcal{H}$. The objective of latent MKL can be formulated as

$$\min_{\boldsymbol{\omega},\boldsymbol{\gamma}} \frac{1}{2}\sum_{p=1}^{m} \frac{\|\boldsymbol{\omega}_p\|^2}{\gamma_p} \quad + \quad C\sum_{i=1}^{n} \max\{0, f_i(\boldsymbol{\omega}) - g_i(\boldsymbol{\omega})\} \tag{9}$$

where

$$\begin{aligned} f_i(\boldsymbol{\omega}) &= \max_{(u,\mathbf{h})\in\mathcal{Y}\times\mathcal{H}} \left[\boldsymbol{\omega}_y^\top \Psi(\mathbf{x}_i; u, \mathbf{h}) + \Delta(y_i, u)\right] \\ &= \max_{(u,\mathbf{h})\in\mathcal{Y}\times\mathcal{H}} \left[\sum_{p=1}^{m} h_p \boldsymbol{\omega}_p^\top \Phi_p(\mathbf{x}_i; u) + \Delta(y_i, u)\right] \end{aligned} \tag{10}$$

and

$$g_i(\boldsymbol{\omega}) = \max_{\mathbf{h}\in\mathcal{H}} \sum_{p=1}^{m} h_p \boldsymbol{\omega}_p^\top \Phi_p(\mathbf{x}_i, y_i) \tag{11}$$

where $\boldsymbol{\omega} = \left[\boldsymbol{\omega}_1^\top, \cdots, \boldsymbol{\omega}_m^\top\right]^\top$, $\mathbf{h} = [h_1, \cdots, h_m]^\top$, $\mathcal{H} = \{\mathbf{h} : \mathbf{h} \in \{0,1\}^m \wedge \|\mathbf{h} - \mathbf{h}_0\| = m_0\}$. $\mathbf{h}_0$ is a binary vector with all bits set to 1, indicating all feature mappings are beneficial for classification of all samples. m_0 is a pre-specified parameter controlling the deviation of each $\mathbf{h}_i$ from $\mathbf{h}_0$.

We generalize the MKL formulation for arbitrary ℓ_q-norms by regularizing over the kernel coefficients or equivalently. The optimization problem of Eq. (9) can be rewritten in the following functional form

$$\begin{aligned} &\min_{\boldsymbol{\omega},\boldsymbol{\gamma}} \frac{1}{2}\sum_{p=1}^{m} \frac{\|\boldsymbol{\omega}_p\|^2}{\gamma_p} + C\sum_{i=1}^{n} \xi_i \\ &s.t. \sum_{p=1}^{m} \gamma_p^q = 1, \gamma_p \geq 0 \\ &\quad f_i(\boldsymbol{\omega}) - g_i(\boldsymbol{\omega}) \leq \xi_i, \quad \xi_i \geq 0, \ \forall i \end{aligned} \tag{12}$$

We use a classical Lagrangian approach [4], [39]–[41] to get $\boldsymbol{\gamma}$. The Lagrangian of the primal is:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\omega}, \boldsymbol{\gamma}) = &\frac{1}{2}\sum_{p=1}^{m} \frac{\|\boldsymbol{\omega}_p\|^2}{\gamma_p} + C\sum_{i=1}^{n} \xi_i - s\left(\sum_{p=1}^{m} \gamma_p^q - 1\right) \\ &- \sum_{p=1}^{m} t_p\gamma_p - l\left(f_i(\boldsymbol{\omega}) - g_i(\boldsymbol{\omega}) - \xi_i\right) \\ &- \sum_{i=1}^{n} w_i\xi_i \end{aligned} \tag{13}$$

Setting the partial derivatives w.r.t. $\boldsymbol{\gamma}$, we obtain the following condition on the optimality of $\boldsymbol{\gamma}$,

$$\frac{\partial\mathcal{L}}{\partial\gamma_p} = -\frac{1}{2}\frac{\|\boldsymbol{\omega}_p\|^2}{{\gamma_p}^2} - sq\gamma_p^{q-1} - t_p = 0 \tag{14}$$

At optimality, we have these conditions which satisfy the KKT condition:

$$\begin{aligned} &(a) \quad \|\boldsymbol{\omega}_p\|^2 = -sq\gamma_p^{q+1} - t_p\gamma_p^2 \\ &(b) \quad \sum_{p=1}^{m} \gamma_p^q = 1 \\ &(c) \quad t_p\gamma_p = 0, \quad \forall p \end{aligned} \tag{15}$$

According to (c), we can state for all p that either $\gamma_p = 0$ and thus $\|\boldsymbol{\omega}_p\| = 0$ or $t_p = 0$ and thus $\gamma_p = \|\boldsymbol{\omega}_p\|^{\frac{2}{q+1}} / (-2sq)^{\frac{1}{q+1}}$. Then at optimility, we have $t_p = 0$ following the KKT condition. With $\sum_{p=1}^{m} \gamma_p^q = \sum_{p=1}^{m} (\|\boldsymbol{\omega}_p\|^{\frac{2}{q+1}} / (-2sq)^{\frac{1}{q+1}})^q = 1$, we have $(-2sq)^{\frac{1}{q+1}} = (\sum_{p=1}^{m} \|\boldsymbol{\omega}_p\|^{\frac{2q}{q+1}})^{\frac{1}{q}}$. Combining these conditions with (a), γ_p can be updated by

$$\gamma_p = \frac{\|\boldsymbol{\omega}_p\|^{\frac{2}{q+1}}}{(\sum_{p=1}^{m} \|\boldsymbol{\omega}_p\|^{\frac{2q}{q+1}})^{\frac{1}{q}}} \tag{16}$$

We optimize the upper bound of the problem in Eq.(9),

$$\begin{aligned} \min_{\boldsymbol{\omega}} L(\boldsymbol{\omega}, \{\mathbf{h}_i\}) = &\frac{1}{2}\sum_{p=1}^{m} \frac{\|\boldsymbol{\omega}_p\|^2}{\gamma_p} \\ &+ C\sum_{i=1}^{n} \max\{0, f_i(\boldsymbol{\omega}) - g_i(\boldsymbol{\omega}; \mathbf{h}_i)\} \end{aligned} \tag{17}$$

where $g_i(\boldsymbol{\omega}; \mathbf{h}_i) = \sum_{p=1}^{m} h_{ip}\boldsymbol{\omega}_p^\top \Phi_p(\mathbf{x}_i; y_i)$ and $\mathbf{H} = [\mathbf{h}_1; \cdots; \mathbf{h}_n]^\top$

According to the representer theorem [42], we have

$$\boldsymbol{\omega}_p = \gamma_p \sum_{i=1}^{n}\sum_{u\in\mathcal{Y}} \hat{\alpha}_{iu}\Phi_p(\mathbf{x}_i; u) \tag{18}$$

and

$$\begin{aligned} \|\boldsymbol{\omega}_p\|^2 &= \gamma_p^2 \sum_{i,j=1}^{n}\sum_{u,v\in\mathcal{Y}} \hat{\alpha}_{iu}\hat{\alpha}_{jv}k_p((\mathbf{x}_i; u), (\mathbf{x}_j; v)) \\ &= \gamma_p^2 \boldsymbol{\alpha}^\top \tilde{\mathbf{K}}_p \boldsymbol{\alpha} \end{aligned} \tag{19}$$

$$\begin{aligned} f_i(\boldsymbol{\omega}) &= \max_{(u,\mathbf{h})\in\mathcal{Y}\times\mathcal{H}} \left[\sum_{p=1}^{m} h_p\omega_p^\top \Phi_p(\mathbf{x}_i; u) + \Delta(y_i, u)\right] \\ &= \max_{(u,\mathbf{h})\in\mathcal{Y}\times\mathcal{H}} \left[\boldsymbol{\alpha}^\top \sum_{p=1}^{m} \gamma_p h_p \mathbf{a}_p + \Delta(y_i, u)\right] \end{aligned} \tag{20}$$

and

$$g_i(\boldsymbol{\omega}) = \max_{\mathbf{h}\in\mathcal{H}} \sum_{p=1}^{m} h_p\boldsymbol{\omega}_p^\top \Phi_p(\mathbf{x}_i; y_i) = \max_{\mathbf{h}\in\mathcal{H}} \boldsymbol{\alpha}^\top \sum_{p=1}^{m} \gamma_p h_p \mathbf{b}_p \tag{21}$$

where $\tilde{\mathbf{K}}_p$ is calculated via $kron(K_{\mathcal{Y}}, \mathbf{K}_p)$ and $K_{\mathcal{Y}}$ is a similarity matrix defied on label set $\mathcal{Y}$. Parameter $\boldsymbol{\alpha}$ is defined as $[\hat{\alpha}_{11}\cdots\hat{\alpha}_{n1},\cdots,\hat{\alpha}_{1c}\cdots\hat{\alpha}_{nc}]^\top \in \mathbb{R}^{n\times c}$ and $\mathbf{a}_p = [\mathbf{K}_p(:,\mathbf{x}_i)^\top K_{\mathcal{Y}}(1,u),\cdots,\mathbf{K}_p(:,\mathbf{x}_i)K_{\mathcal{Y}}(c,u)]^\top$ and $\mathbf{b}_p = [\mathbf{K}_p(:,\mathbf{x}_i)K_{\mathcal{Y}}(1,y_i),\cdots,\mathbf{K}_p(:,\mathbf{x}_i)^\top K_{\mathcal{Y}}(c,y_i)]^\top$.

Combining Eq.(19) and Eq.(16), we obtain

$$\gamma_p = \frac{(\gamma_p^2\boldsymbol{\alpha}^\top\tilde{\mathbf{K}}_p\boldsymbol{\alpha})^{\frac{1}{q+1}}}{(\sum_{p=1}^m(\gamma_p^2\boldsymbol{\alpha}^\top\tilde{\mathbf{K}}_p\boldsymbol{\alpha})^{\frac{q}{q+1}})^{\frac{1}{q}}} \tag{22}$$

Combining Eq.(17) and Eq.(19), we obtain

$$\min_{\boldsymbol{\alpha},\mathbf{H},\gamma\in\theta} L(\boldsymbol{\alpha};\{\mathbf{h}_i\}) = \frac{1}{2}\boldsymbol{\alpha}^\top\left(\sum_{p=1}^m\gamma_p\tilde{\mathbf{K}}_p\right)\boldsymbol{\alpha} + C\sum_{i=1}^n\max\{0, f_i(\boldsymbol{\alpha}) - g_i(\boldsymbol{\alpha};\mathbf{h}_i)\} \tag{23}$$

where

$$f_i(\boldsymbol{\alpha}) = \max_{(u,\mathbf{h})\in\mathcal{Y}\times\mathcal{H}}\left[\boldsymbol{\alpha}^\top\sum_{p=1}^m\gamma_p h_{ip}\mathbf{a}_p + \Delta(y_i,u)\right] \tag{24}$$

$$g_i(\boldsymbol{\alpha};\mathbf{h}_i) = \max_{\mathbf{h}\in\mathcal{H}}\boldsymbol{\alpha}^\top\sum_{p=1}^m\gamma_p h_{ip}\mathbf{b}_p \tag{25}$$

B. OPTIMIZATION

Inspired by the works in [32], we try to solve the ℓ_q-norm SAMKL by the cutting plane method. In this section, we use a "n-slack" formulation to solve the optimization problem. Two different ways of using a hinge loss to covex upper bound the loss is proposed in [43], namely "margin-rescaling" and "slack-rescaling". Margin-rescaling methond is used in this section. Combing Eq.(23), Eq.(24) and Eq.(25), we obtain the following optimization problem

$$\begin{aligned}
&\min_{\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{\xi}\geq 0} L(\boldsymbol{\alpha};\{\mathbf{h}_i\}) = \frac{1}{2}\boldsymbol{\alpha}^\top\left(\sum_{p=1}^m\gamma_p\tilde{\mathbf{K}}_p\right)\boldsymbol{\alpha} + C\sum_{i=1}^n\xi_i\\
&s.t.\ \forall(\bar{y}_1,\mathbf{h}_1)\in\mathcal{Y}\times\mathcal{H}: \boldsymbol{\alpha}^\top\sum_{p=1}^m\gamma_p h_{1p}^\star\mathbf{b}_p - \boldsymbol{\alpha}^\top\sum_{p=1}^m\gamma_p h_{1p}\mathbf{a}_p\\
&\quad\geq \Delta(y_1,\bar{y}_1) - \xi_1\\
&\quad\vdots\\
&s.t.\ \forall(\bar{y}_n,\mathbf{h}_n)\in\mathcal{Y}\times\mathcal{H}: \boldsymbol{\alpha}^\top\sum_{p=1}^m\gamma_p h_{np}^\star\mathbf{b}_p - \boldsymbol{\alpha}^\top\sum_{p=1}^m\gamma_p h_{np}\mathbf{a}_p\\
&\quad\geq \Delta(y_n,\bar{y}_n) - \xi_n
\end{aligned} \tag{26}$$

where ξ_i is shared among constraints from the same sample. $\Delta(y_i,\bar{y}_i)$ is a function that quantifies the loss associated with predicting $\bar{y}_i$ when y_i is the ground-truth. The ground-truth labels are not excluded from the constraints because they correspond to non-negativity constraints on the slack variables ξ_i. And $\sum\xi_i$ is an upper bound on the empirical risk on the training sample $\mathbf{S} = \{(\mathbf{x}_1,y_1),\cdots,(\mathbf{x}_n,y_n)\}$ [32], [44]. We give the cutting-plane algorithm with margin-rescaling via the n-slack formulation in Alg. 1.

Algorithm 1 Cutting Plane for SAMKL With Margin-Rescaling via the n-Slack Formulation

Input: $\mathbf{C}, \varepsilon, \mathbf{S}$.
Output: $(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\xi})$.

$\mathcal{W} \leftarrow \emptyset,\ \ \xi_i \leftarrow \emptyset\ \ , \forall i$
repeat
  for $i = 1,\cdots,n$ **do**
    $(\hat{y}_i^\star, \hat{\mathbf{h}}_i^\star) \leftarrow \arg\max_{\hat{y}_i\in\mathcal{Y},\mathbf{h}_i\in\mathcal{H}}\left\{\Delta(y_i,\hat{y}_i) + \boldsymbol{\alpha}^\top\sum_{p=1}^m\gamma_p h_{ip}\mathbf{a}_p\right\}$
    $\mathbf{h}_i^\star \leftarrow \arg\max_{\mathbf{h}_i\in\mathcal{H}}\boldsymbol{\alpha}^\top\sum_{p=1}^m\gamma_p h_{ip}\mathbf{b}_p$
    if $\Delta(y_i,\hat{y}_i^\star) - \boldsymbol{\alpha}^\top\sum_{p=1}^m\left(\gamma_p h_{ip}^\star\boldsymbol{b}_p - \gamma_p\hat{h}_{ip}^\star\mathbf{a}_p\right) > \xi_i + \varepsilon$ **then**
      $\mathcal{W} \leftarrow \mathcal{W}\cup\{\hat{y}_i^\star, \mathbf{h}_i^\star\}$
      $(\boldsymbol{\alpha},\xi_i) \leftarrow \arg\min_{\boldsymbol{\alpha},\boldsymbol{\xi}\geq 0}\frac{1}{2}\boldsymbol{\alpha}^\top\left(\sum_{p=1}^m\gamma_p\tilde{\mathbf{K}}_p\right)\boldsymbol{\alpha} + \mathbf{C}\sum_{i=1}^m\xi_i$
      $s.t.\ \ \forall(\bar{y}_1,\mathbf{h}_1)\in\mathcal{W}: \boldsymbol{\alpha}^\top\sum_{p=1}^m\gamma_p(h_{1p}^\star\mathbf{b}_p - h_{1p}\mathbf{a}_p) \geq \Delta(y_1,\bar{y}_1) - \xi_1$
      $\vdots$
      $s.t.\ \ \forall(\bar{y}_n,\mathbf{h}_n)\in\mathcal{W}: \boldsymbol{\alpha}^\top\sum_{p=1}^m\gamma_p(h_{np}^\star\mathbf{b}_p - h_{np}\mathbf{a}_p) \geq \Delta(y_n,\bar{y}_n) - \xi_n$
    end if
  end for
  for $p = 1,\cdots,m$ **do**
    $\gamma_p \leftarrow \tau_p^{\frac{1}{q+1}} \Big/ \left(\sum_{p=1}^m\tau_p^{\frac{q}{q+1}}\right)^{\frac{1}{q}}$ where $\tau_p = \gamma_p^2\boldsymbol{\alpha}^\top\tilde{\mathbf{K}}_p\boldsymbol{\alpha}$
  end for
Until no $\mathcal{W}_i$ has changed during iteration
return$(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\xi})$

Since the optimization problem in Eq.(26) has $O(n|\mathcal{Y}|\times|\mathcal{H}|)$ constraints, it can not be solved efficiently. For real-value outputs, $|\mathcal{Y}|$ is typically extremely large. Besides, $|\mathcal{H}|$ grows exponentially with the increase of the number of base kernels. For sample i in $\mathbf{S}$, it has at most 2^m $\mathbf{h}_i$, which constructs $\mathcal{H}$. We reformulate the optimization problem by replacing the n cutting-plane models of the hinge loss, one for each training example, with a single cutting plane model for the sum of the hinge-losses. The "1-slack" formulation of Eq.(26) is

$$\begin{aligned}
&\min_{\boldsymbol{\alpha},\boldsymbol{\gamma},\xi\geq 0} L(\boldsymbol{\alpha};\{\mathbf{h}_i\}) = \frac{1}{2}\boldsymbol{\alpha}^\top\left(\sum_{p=1}^m\gamma_p\tilde{\mathbf{K}}_p\right)\boldsymbol{\alpha} + C\xi\\
&s.t.\ \forall((\bar{y}_1,\mathbf{h}_1),\cdots,(\bar{y}_n,\mathbf{h}_n))\in\mathcal{Y}^n\times\mathcal{H}^n:\\
&\quad\frac{1}{n}\boldsymbol{\alpha}^\top\sum_{i=1}^n\sum_{p=1}^m\left(\gamma_p h_{ip}^\star\mathbf{b}_p - \gamma_p h_{ip}\mathbf{a}_p\right) \geq \frac{1}{n}\sum_{i=1}^n\Delta(y_i,\bar{y}_i) - \xi
\end{aligned} \tag{27}$$

In the formulation aforementioned, for each possible combination of labels and hidden models $((\bar{y}_1, \mathbf{h}_1), \cdots, (\bar{y}_n, \mathbf{h}_n))$, it has only one slack variable ξ that is shared across all constraints. This optimization problem in Eq. 27 is a non-linear integer programming, which can be solved via quadratic programming. We give the cutting-plane algorithm with margin-rescaling via the 1-slack formulation in Alg. 2.

Algorithm 2 Cutting Plane for SAMKL With Margin-Rescaling via the 1-Slack Formulation

Input: $\mathbf{C}, \varepsilon, \mathbf{S}$.
Output: $(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \xi)$.

1: $\mathcal{W} \leftarrow \emptyset$
2: **repeat**
3: $(\boldsymbol{\alpha}, \xi) \leftarrow \underset{\boldsymbol{\alpha}, \xi \geq 0}{\arg\min} \frac{1}{2}\boldsymbol{\alpha}^\top \left(\sum_{p=1}^{m} \gamma_p \tilde{\mathbf{K}}_p \right) \boldsymbol{\alpha} + \mathbf{C}\xi$
s.t. $\forall ((\bar{y}_1, \mathbf{h}_1), \ldots, (\bar{y}_n, \mathbf{h}_n)) \in \mathcal{W}$:
$\frac{1}{n}\boldsymbol{\alpha}^\top \sum_{i=1}^{n} \sum_{p=1}^{m} \left(\gamma_p h_{ip}^{\star} \mathbf{b}_p - \gamma_p h_{ip} \mathbf{a}_p \right)$
$\geq \frac{1}{n} \sum_{i=1}^{n} \Delta(y_i, \bar{y}_i) - \xi$
4: **for** $i = 1, \ldots, n$ **do**
5: $(\hat{y}_i^{\star}, \hat{\mathbf{h}}_i^{\star}) \leftarrow \underset{\hat{y}_i \in \mathcal{Y}, \mathbf{h}_i \in \mathcal{H}}{\arg\max} \left\{ \Delta(y_i, \hat{y}_i) + \boldsymbol{\alpha}^\top \sum_{p=1}^{m} \gamma_p h_{ip} \mathbf{a}_p \right\}$
6: $\mathbf{h}_i^{\star} \leftarrow \underset{\mathbf{h}_i \in \mathcal{H}}{\arg\max} \boldsymbol{\alpha}^\top \sum_{p=1}^{m} \gamma_p h^{\star} \mathbf{b}_p$
7: **end for**
8: **for** $p = 1, \ldots, m$ **do**
9: $\gamma_p \leftarrow \tau_p^{\frac{1}{q+1}} \Big/ \left(\sum_{p=1}^{m} \tau_p^{\frac{q}{q+1}} \right)^{\frac{1}{q}}$ where $\tau_p = \gamma_p^2 \boldsymbol{\alpha}^\top \tilde{\mathbf{K}}_p \boldsymbol{\alpha}$
10: **end for**
11: $\mathcal{W} \leftarrow \mathcal{W} \cup \{((\hat{y}_1^{\star}, \mathbf{h}_1^{\star}), \cdots, (\hat{y}_n^{\star}, \mathbf{h}_n^{\star}))\}$
12: **until** $\frac{1}{n} \sum_{i=1}^{n} \Delta(y_i, \hat{y}_i^{\star})$
$-\frac{1}{n}\boldsymbol{\alpha}^\top \sum_{i=1}^{n} \sum_{p=1}^{m} \left(\gamma_p h_{ip}^{\star} \mathbf{b}_p - \gamma_p \hat{h}_{ip}^{\star} \mathbf{a}_p \right) \leq \xi + \varepsilon$
13: **return**$(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \xi)$

Alg. 2 iteratively constructs a working set $\mathcal{W}$ of constraints. In each iteration, the algorithm computes the solution over $\mathcal{W}$ (Line 3), finds the most violated constraint (Lines 4-7) and adds it to the working set. The algorithm stops when no constraint can be found that is violated by more than the desired precision ε (Line 12).

C. DISCUSSION

As mentioned before, the $n-$slack and the $1-$slack formulations of our problem are equivalent. Therefore, the objective functions of Alg. 1 and Alg. 2 are equal. That means the theoretical results for those algorithms are consistent. The proof can be found in Theorem 1 mentioned in reference [32]. Unlike in the $n-$slack algorithm Alg. 1 where the number of constraints increases exponentially in the solving process, only a single constraint is added in each iteration of $1-$slack algorithm Alg. 2. So the $1-$ slack algorithm is more efficient than the $n-$slack algorithm. For this reason, we implement Alg. 2 to validate the high efficiency and effectiveness of our work in Section 4.

From Fig. 2(a) we can see that the computation time increases linearly with the number of iterations. It can be seen from Fig. 2(c) that the objective function value of the cutting plane based optimization Alg. 2 (denoted as CP-SAMKL) is monotonic, while the alternate coordinate descent-based optimization of Eq. 23 (denoted as ACD-SAMKL) is not. Therefore, the classification performance of CP-SAMKL is much more stable compared to ACD-SAMKL with the number of iteration increases. Besides, from Fig. 2(d), we can see that the CP-SAMKL is easier to converge than ACD-SAMKL. For these reasons, the cutting plane based optimization of SAMKL is used in Section 4.

IV. EXPERIMENTAL RESULTS

All of the experiments were carried out on a computer with a 3.6GHz Intel Xeon E5-1620 CPU and 48GB of memory with Matlab R2014a (64bit).

A. DATASETS

A wide range of image datasets used in our experiment is summarized in Table 1. The number of datasets classes ranges from 2 to 37, the sample number reaches up to 2,600, the views of each dataset scales from 7 to 14. Besides, two datasets used for protein subcellular localization are given in Table 2, including psortPos and plant datasets. These protein datasets have been widely used by MKL algorithms [37] and can be downloaded from website.[1]

Caltech256:[2] It is a collection of 256 object categories containing a total of 30,607 images. These categories are grouped by animate and inanimate and other finer distinctions [45]. And the animate objects - 69 categories in all - tend to be more cluttered than the inanimate objects, and harder to identify. The air animals of the animate objects are used in our experiment except for iris and hawksbill-101, which are not air animals. That is to say, a subset of Caltech256 with a total of 1,032 samples in 9 classes are used in our experiments. These categories are depicted in Fig. 3(b).

Birds200:[3] It is an image dataset with photos of 200 bird species (mostly North American). A total of 882 samples in 15 birds categories are selected in our experiments. These 15 categories are easy to be confused by the human eye and we get this subset by clustering. These categories are depicted in Fig. 3(a)

STL-10:[4] It is an image recognition dataset for developing unsupervised feature learning, deep learning, self-taught learning algorithms. A total of 2,600 samples from this dataset are selected, with each of the two classes (dog and cat) has 1,300 samples.

[1] http://mkl.ucsd.edu/dataset/
[2] http://www.vision.caltech.edu/Image_Datasets/Caltech256/
[3] http://www.vision.caltech.edu/visipedia/CUB-200-2011.html
[4] https://cs.stanford.edu/~acoates/stl10/

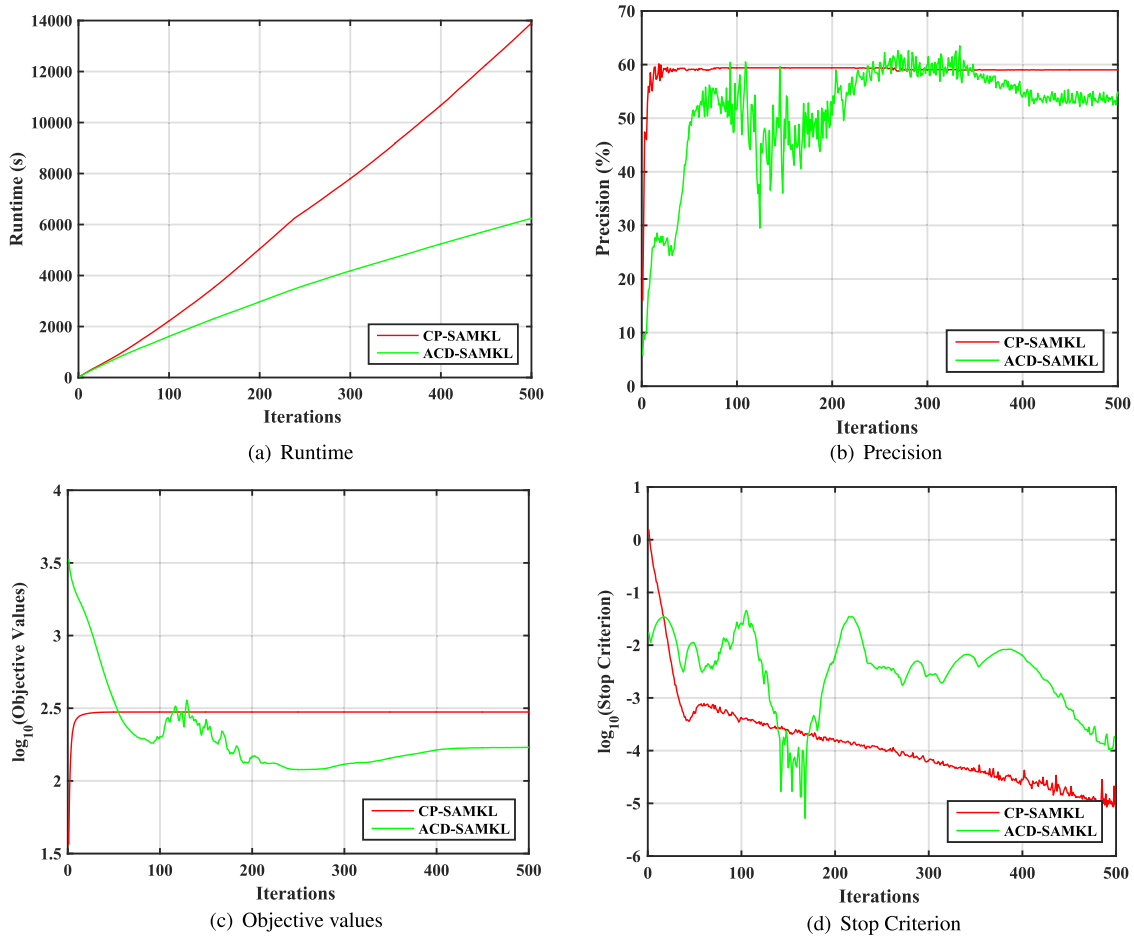


FIGURE 2. Comparison between proposed CP-SAMKL and ACD-SAMKL on 15birds_confuse_hybrid dataset. The regularization parameter C is set to 10^8 and 1 for CP-SAMKL and ACD-SAMKL, respectively. m_0, ε and the maximum iterations is set to 2, 10^{-6} and 500 for these algorithms.

Cifar-100:[5] This dataset has 100 classes containing 600 images each. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. Each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs). A total of 1,100 samples from this dataset are randomly selected in a balanced manner, with each of the 11 classes has 100 samples. The 11 classes are easy to be confused by the human eye, including aquarium fish, crocodile, dolphin, flatfish, otter, ray, seal, shark, trout, turtle, and whale, and they are depicted as Fig. 3(c).

Vgg Pets:[6] It is a collection of pets, covering 37 different breeds of cats and dogs, with roughly 200 images for each class. A total of 1480 samples from this dataset are selected, with each class has 40 samples.

For the image datasets, we extracted features using traditional machine learning methods and deep learning algorithms. Six features are extracted from all images using the toolbox downloaded from.[7] These features are color (dim 420) [46], [47], gist (dim 512) [48], dense hog2 × 2 (dim 420), dense hog3 × 3 (dim 420) [33], [49], lbp (dim 1239) [35] and dense sift (dim 420) [34]. Texture features are extracted using the statxture function by

TABLE 1. Image datasets used in the experiments and their feature descriptions.

Dataset	Subset	Samples	classes
Caltech256	air animal	1032	9
Birds200	15 birds	882	15
STL-10	STL dogcat	2600	2
Cifar-100	11 confuse	1100	10
Vgg Pets	vgg pets	1480	37

TABLE 2. Protein datasets used in the experiments.

Dataset	Training samples	Test samples	classes	kernels
psortPos	272	269	4	69
plant	471	469	4	69

[5]https://www.cs.toronto.edu/~kriz/cifar.html
[6]http://www.robots.ox.ac.uk/~vgg/data/pets/
[7]https://github.com/adikhosla/feature-extraction

(a) 15birds selected from Birds200

(b) air animals selected from Caltech256

(c) 11 confuse animals selected from cifar100

FIGURE 3. **Details of dataset.**

matlab [50]. These seven features are denoted as normal features in our experiments. Besides, we extract another seven features using the pretrained models downloaded from,[8] including imagenet-resnet-50-dag model, imagenet-googlenet-dag model, imagenet-vgg-f model, imagenet-vgg-s model, imagenet-vgg-m model, imagenet-caffe-ref model and imagenet-vgg-verydeep16 model. These features are denoted as deep features in our experiments. For each image dataset, we construct 3 subsets with 7 normal features, 7 deep features and 14 hybrid features (7 normal features and 7 deep features) respectively.

For all of these image data sets, we randomly split the data into 10 groups, with 50% : 50% for training and test. In all experiments, the Gaussian kernels were used to build the similarity matrix for each individual view. The standard deviation (parameter σ) was set to the median of the pairwise Euclidean distances between every pair of data points for all datasets.

[8]http://www.vlfeat.org/matconvnet/pretrained/

B. BASELINES

We compare the proposed algorithms with state-of-the-art MKL algorithms.

UMKL: It is a uniformly weighted MKL algorithm. And we implement it based on the LIBSVM [9] package.

SimpleMKL: [6] It is a well-known baseline with max-margin principle. Its Matlab implementation is available from.[10] Its formulation of the MKL problem results in a smooth and convex optimization problem, which is equivalent to other MKL formualtions available in the literature. The main added value of the smoothness of the new objective function is that descent methods become practical and efficient means to solve the optimization problem that wraps a single kernel solver. It provides optimality conditions, analyzes convergence and computational complexity issues for binary classification.

ℓ_q-MKL: [4] It is an efficient algorithm for multiple kernel learning by discussing the connection between MKL and group-lasso regularizer. It calculates the kernel weights by a closed-form formulation, which therefore leverages the dependency of previous algorithms on employing complicated or commercial optimization software. It is a general max-margin MKL framework with ℓ_q-norm constraint on kernel weights. We consider $q = 1, 2, 4$ and use its Matlab implementation.

SAMKL: [3] In this algorithm, the base kernels are allowed to be adaptively switched on/off with respect to each sample. A latent binary variable was assigned to each base kernel when it is applied to a sample. The kernel combination weights and the latent variables are jointly optimized via the margin maximization principle.

C. RESULT ANALYSIS

Following [37], F1-score is used to measure classification performance on psortPos data set, while the matthew correlation coefficient (MCC) is used for the plant data set. The results of SAMKL are reported from the original paper [3], while the others are obtained by us running the released code. For these protein data sets, we randomly split the data into 20 groups, with 50%: 50% for training and test. For our proposed algorithm, C is chosen from $[10^3, 10^4, \ldots, 10^{12}]$ by five-fold cross-validation and m_0 is chosen adaptively in the optimization of the algorithm. As seen in Table 3, our proposed algorithm achieves superior performance to the baselines on the protein data sets.

Precision is used to measure classification performance on the image datasets used in our experiments. Table 4 shows the classification results of the proposed algorithm and the baselines on each data set. Each cell represents mean precision and standard deviation. Boldface means the best one. From this table, we can see that the image datasets using only normal features achieve inferior classification performance compared with that using deep features. It also can be seen

[9]https://www.csie.ntu.edu.tw/~cjlin/libsvm/
[10]https://github.com/maxis1718/SimpleMKL

TABLE 3. Experimental comparison of the proposed 1-slack CP-SAMKL and the baselines on protein data sets. The two rows of each cell represent mean accuracy(standard deviation) and training/test time (in seconds). Boldface means the best one.

Dataset	*UMKL*	*SimpleMKL*	*lq-MKL*			*SAMKL*	*Proposed*			
			q=2	q=4	q=8		q=1	q=2	q=4	q=8
psortPos	85.5(1.1) 0.0/0.0	89.6(1.8) 69.3/0.0	88.4(1.4) 1.6/0.0	86.6(1.6) 1.0/0.0	86.3(1.4) 0.8/0.0	90.6(1.5) 343.0/5.8	**90.6(1.2)** **20.8/0.9**	89.1(0.9) 20.8/1.3	88.1(1.1) 21.2/1.3	88.0(1.1) 21.1/1.3
plant	82.5(2.2) 0.0/0.0	88.2(1.7) 127.9/0.0	88.3(1.4) 3.1/0.0	86.9(1.5) 2.0/0.0	86.5(1.5) 1.5/0.0	89.5(1.7) 864.7/18.4	**89.6(1.0)** **59.6/6.4**	87.5(1.2) 62.5/5.3	86.3(1.4) 62.1/5.1	85.6(1.3) 62.1/6.1

TABLE 4. Experimental comparison of the proposed 1-slack CP-SAMKL and the baselines on data sets.

Dataset	*UMKL*	*SimpleMKL*	*lq-MKL*			*Proposed*		
			q=1	q=2	q=4	q=1	q=2	q=4
air_animal_normal	48.0(1.7)	50.2(2.1)	51.2(2.3)	50.7(2.1)	50.3(1.9)	**51.3(1.8)**	49.7(2.1)	50.3(1.9)
air_animal_deep	87.2(1.3)	87.8(0.7)	88.2(0.7)	87.8(0.9)	87.7(1.1)	**88.6(1.0)**	87.7(1.2)	87.8(1.2)
air_animal_hybrid	84.3(1.0)	88.4(0.9)	88.2(1.0)	86.6(1.1)	85.7(1.2)	**88.5(1.2)**	87.9(1.3)	87.5(1.4)
15birds_confuse_normal	16.3(2.0)	19.5(2.3)	19.9(1.6)	19.6(1.7)	19.2(1.6)	**20.4(2.3)**	19.4(1.6)	19.3(2.6)
15birds_confuse_deep	56.6(1.6)	52.3(1.3)	54.5(2.1)	57.4(1.3)	57.5(1.7)	**60.0(2.3)**	58.6(2.6)	57.8(2.5)
15birds_confuse_hybrid	48.5(1.9)	53.3(2.7)	54.4(3.4)	54.2(3.0)	52.6(2.9)	**59.4(2.0)**	57.0(2.1)	56.1(2.1)
stl_dogcat_normal	65.3(2.1)	78.0(0.8)	79.0(0.9)	78.3(1.1)	78.3(0.8)	79.6(1.2)	**79.7(0.9)**	79.4(0.7)
stl_dogcat_deep	94.7(0.4)	94.4(0.6)	94.4(0.6)	94.5(0.5)	94.5(0.4)	**95.0(0.4)**	94.8(0.5)	94.7(0.3)
stl_dogcat_hybrid	94.5(0.6)	94.2(0.3)	94.3(0.3)	94.0(0.5)	93.8(0.5)	**95.0(0.4)**	95.0(0.4)	94.7(0.4)
cifar100_confuse11_normal	39.6(1.7)	39.3(1.5)	38.6(2.5)	**39.7(2.1)**	39.4(1.9)	39.5(1.3)	39.3(1.2)	38.7(1.6)
cifar100_confuse11_deep	49.3(2.0)	47.2(2.1)	47.9(2.9)	50.1(2.7)	**50.2(1.8)**	50.0(1.7)	49.2(2.4)	49.0(1.4)
cifar100_confuse11_hybrid	48.6(2.1)	46.8(1.9)	48.3(1.5)	48.9(1.5)	48.3(1.6)	**50.4(2.6)**	49.1(2.2)	49.5(2.6)
vgg_pets_normal	16.5(1.1)	18.7(1.1)	19.5(1.3)	**19.8(1.3)**	19.7(1.4)	19.1(1.1)	18.9(1.6)	19.3(1.4)
vgg_pets_deep	83.4(1.1)	84.0(1.1)	84.4(1.3)	83.6(1.0)	83.5(1.0)	**84.8(1.0)**	83.6(1.1)	82.6(1.6)
vgg_pets_hybrid	78.2(1.5)	83.5(0.8)	83.7(0.7)	80.9(1.2)	79.9(1.2)	**84.4(1.1)**	83.4(1.2)	83.1(1.2)

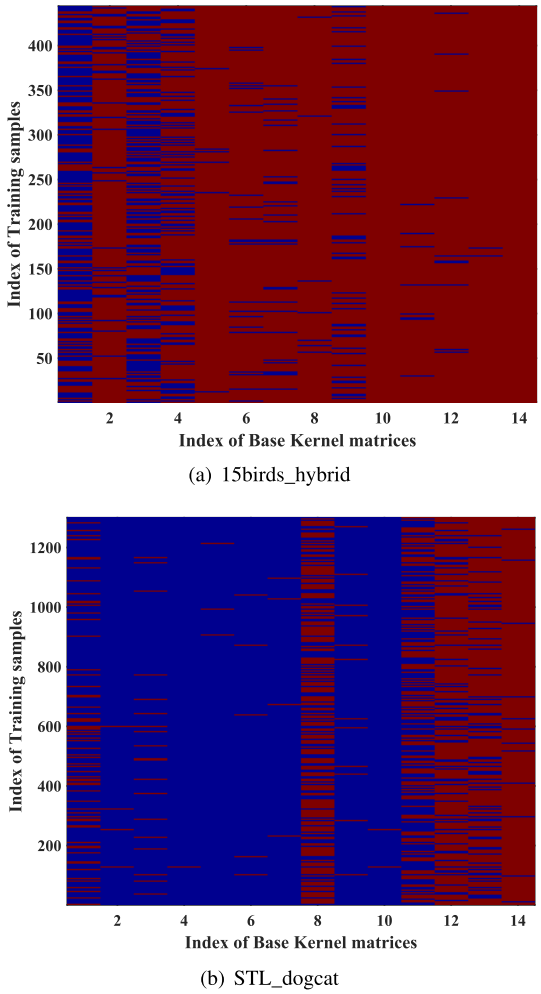
TABLE 5. Experimental comparison of the proposed 1-slack CP-SAMKL and the baselines on data sets with rub kernels.

Dataset	*UMKL*	*SimpleMKL*	*lq-MKL*			*Proposed*		
			q=1	q=2	q=4	q=1	q=2	q=4
15birds_confuse_hybrid_rub	45.3(1.5)	53.0(2.4)	54.4(2.7)	52.7(2.6)	50.0(2.7)	**55.0(1.9)**	53.3(1.5)	52.8(1.8)
air_animal_hybrid_rub	82.6(1.2)	87.5(0.6)	88.1(0.6)	86.1(0.6)	85.2(0.8)	**88.2(1.5)**	87.5(1.0)	87.3(1.3)
cifar100_confuse11_hybrid_rub	45.1(2.2)	42.6(1.7)	44.4(2.1)	45.8(1.4)	45.9(1.5)	**46.6(1.9)**	45.5(2.3)	46.4(1.6)
vgg_pets_hybrid_rub	77.1(1.5)	82.5(1.2)	82.6(1.3)	80.1(1.4)	79.3(1.4)	**83.8(0.9)**	82.6(1.4)	80.9(1.3)
stl_dogcat_hybrid_rub	93.0(0.4)	92.8(0.5)	93.0(0.4)	92.8(0.6)	92.6(0.8)	**94.0(0.3)**	93.7(0.4)	93.5(0.6)

that all the algorithms achieve excellent performance on the datasets using deep features. And the performance achieved by the uniformly weighted MKL is comparable to that of SimpleMKL and Lp-norm MKL. But our proposed algorithm can further improve the classification performance compared to the baselines.

Table 5 shows the classification results of the proposed algorithm and the baselines on data sets with rub kernels. These rub kernels are generated by setting a ratio of samples of several views to 0. 30% of the samples are selected randomly and their values of the randomly selected 50% views are set to 0. From this table, we can see that the proposed algorithm can further improve the classification performance compared to the baselines.

The learned latent variable $\mathbf{h}$ is shown in 4. The $\mathbf{h}$ on each classification task is shown as an $n \times m$ matrix, where n and m are the number of training samples and base kernels, respectively. As can be seen, the active latent variables indicating "1" are in blue while the others indicating "0" are in red. The blue color indicates those latent variables which switch off the base kernels whole weights are nonzeros. As shown, $\mathbf{h}$ switches on/off the base kernels differently across training samples. Due to the constraint $||\mathbf{h}_i - \mathbf{h}_0|| \leq m_0, \forall i$, each row of these matrices has a fixed number of "0"s. They are 6 and 10 for 15birds_hybird and STL_dogcat, respectively. It also can be seen that the blue area is on the left side of Fig. 4(a) and Fig. 4(b), which means most of the kernels extracted from the normal features are switched off. That means, combining the kernels extracted using deep features,



(a) 15birds_hybrid

(b) STL_dogcat

FIGURE 4. The latent varible h learned for each sample of a training group on different data sets.

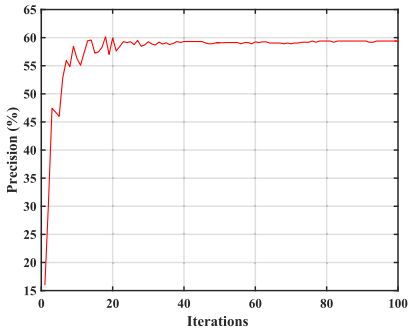
we can get superior classification performance. This rule can be seen from Table. 4. Besides, our proposed algorithm achieves comparable results on the data sets using hybrid features. These experiments preliminarily demonstrate the effectiveness and the properties of the proposed ℓ_q-norm SAMKL.

D. PARAMETER SELECTION

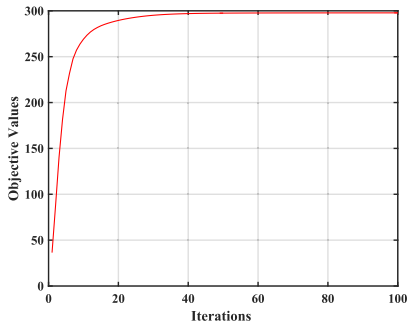
For the image data sets using only normal features or deep features, m for our proposed algorithm is chosen from $[0, 1, 2, 3, 4]$ by 5-fold cross-validation. For the data sets using hybrid features, m_0 is selected from $[0, 2, 4, 6, 8]$. The penalty parameter C for our proposed 1-slack CP-SAMKL is set to a fixed value of 10^8. Each base kernel matrix is normalized to have a unit trace.

We perform 5-fold cross-validation on training data sets to select the regularization parameter $C \in \{10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4\}$ for UMKL, SimpleMKL and ℓ_q-norm MKL.

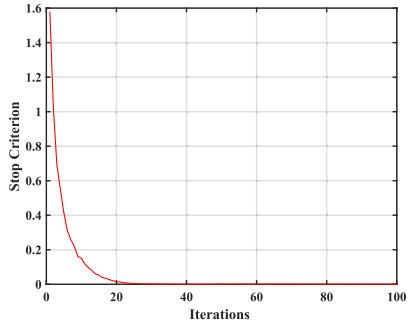
Fig. 5 shows the effect of the iterations on the 1st split of 15 birds data set using hybrid features. And in this setting, the value of regularization parameter C and selected channels



(a) Precision vs. Iterations



(b) Objective Values vs. Iterations



(c) Stop Criterions vs. Iterations

FIGURE 5. Effect of the iterations on the 15 birds dataset using hybrid features.

m_0 are fixed to 10^8 and 2 for convenience, respectively. As the Fig. 5(a) shows, the classification precision increases as the number of iterations of the proposed algorithm increases. And the performance is relative stable when the iteration is too large. It can seen from Fig. 5(b) that the objective function value increases with the number of iteration monotonically increasing. Fig. 5(c) the stop criterion (the value of $\frac{1}{n}\sum_{i=1}^{n}\Delta(y_i,\hat{y}_i^{\star})-\frac{1}{n}\boldsymbol{\alpha}^{\top}\sum_{i=1}^{n}\sum_{p=1}^{m}\left(\gamma_p h_{ip}^{\star}\mathbf{b}_p-\gamma_p\hat{h}_{ip}^{\star}\mathbf{a}_p\right)-\xi$) monotonically decreases dramatically with the number of iteration increases. So, we set the value of iterations to 20 for our proposed algorithm 1-slack CP-SAMKL.

V. CONCLUSION

This work proposes an efficient ℓ_q-norm SAMKL problem which jointly performs MKL and infers the base kernel subsets that are useful for the classification of each sample. By allowing each sample to adaptively switch on/off each base kernel, ℓ_q-norm SAML achieves clear improvement over the comparable MKL algorithms in recent literature. In this paper, we solve the optimization problem using cutting plane methods, and construct datasets using mainstream machine learning methods and deep learning methods. Extensive experiments exhibit the effectiveness of our proposed algorithm. Further improving the classification performance of the proposed SAMKL is another piece of our future work.

REFERENCES

[1] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, And Beyond*. Cambridge, MA, USA: MIT Press, 2002.
[2] Q. Wang, Y. Dou, X. Liu, Q. Lv, and S. Li, "Multi-view clustering with extreme learning machine," *Neurocomputing*, vol. 214, pp. 483–494, Nov. 2016.
[3] X. Liu, L. Wang, J. Zhang, and J. Yin, "Sample-adaptive multiple kernel learning," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2014, pp. 1975–1981.
[4] Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu, "Simple and efficient multiple kernel learning by group lasso," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 1175–1182.
[5] 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. (ICML)*, 2004, p. 6.
[6] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, Nov. 2008.
[7] A. J. Pinar, J. Rice, L. Hu, D. T. Anderson, and T. C. Havens, "Efficient multiple kernel classification using feature and decision level fusion," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1403–1416, Dec. 2017.
[8] Y. Lu, L. Wang, J. Lu, J. Yang, and C. Shen, "Multiple kernel clustering based on centered kernel alignment," *Pattern Recognit.*, vol. 47, no. 11, pp. 3656–3664, Nov. 2014.
[9] H. Zhang, J. Yang, Y. Zhang, and T. S. Huang, "Image and video restorations via nonlocal kernel regression," *IEEE Trans. Cybern.*, vol. 43, no. 3, pp. 1035–1046, Jun. 2013.
[10] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[11] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 606–613.
[12] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 221–228.
[13] S. S. Bucak, R. Jin, and A. K. Jain, "Multiple kernel learning for visual object recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1354–1369, Jul. 2014.
[14] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, Jan. 2004.
[15] 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.
[16] Z. Xu, R. Jin, I. King, and M. Lyu, "An extended level method for efficient multiple kernel learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1825–1832.
[17] F. Orabona and L. Jie, "Ultra-fast optimization algorithm for sparse multi kernel learning," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 249–256.
[18] A. Afkanpour, A. György, C. Szepesvári, and M. Bowling, "A randomized mirror descent algorithm for large scale multiple kernel learning," in *Proc. ICML*, 2013, pp. 374–382.
[19] M. Varma and B. R. Babu, "More generality in efficient multiple kernel learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 1065–1072.
[20] M. Kloft, U. Brefeld, P. Laskov, K.-R. Müller, A. Zien, and S. Sonnenburg, "Efficient and accurate lp-norm multiple kernel learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 997–1005.
[21] M. Gonen, "Bayesian efficient multiple kernel learning," 2012, *arXiv:1206.6465*. [Online]. Available: http://arxiv.org/abs/1206.6465

[22] C. Cortes, M. Mohri, and A. Rostamizadeh, "Multi-class classification with maximum margin multiple kernel," in *Proc. ICML*, 2013, pp. 46–54.
[23] A. Kumar, A. Niculescu-Mizil, K. Kavukcuoglu, and H. Daume, III, "A binary classification framework for two-stage multiple kernel learning," 2012, *arXiv:1206.6428*. [Online]. Available: http://arxiv.org/abs/1206.6428
[24] Y. Lei, A. Binder, Ü. Dogan, and M. Kloft, "Localized multiple kernel Learning—A convex approach," 2015, *arXiv:1506.04364*. [Online]. Available: http://arxiv.org/abs/1506.04364
[25] S. Niazmardi, B. Demir, L. Bruzzone, A. Safari, and S. Homayouni, "Multiple kernel learning for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1425–1443, 2017.
[26] J. Moeller, S. Swaminathan, and S. Venkatasubramanian, "A unified view of localized kernel learning," in *Proc. SIAM Int. Conf. Data Mining*, Aug. 2016, pp. 252–260.
[27] A. Rakotomamonjy and S. Chanda, "ℓ_p-norm multiple kernel learning with low-rank kernels," *Neurocomputing*, vol. 143, pp. 68–79, Nov. 2014.
[28] Y. Han, K. Yang, Y. Ma, and G. Liu, "Localized multiple kernel learning via sample-wise alternating optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 137–148, Jan. 2014.
[29] G. Fu, Q. Wang, H. Wang, and D. Bai, "Group based non-sparse localized multiple kernel learning algorithm for image classification," in *Proc. 4th Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Aug. 2016, pp. 191–195.
[30] M. Gönen and E. Alpaydın, "Localized algorithms for multiple kernel learning," *Pattern Recognit.*, vol. 46, no. 3, pp. 795–807, 2013.
[31] M. Gönen and E. Alpaydın, "Localized multiple kernel learning," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 352–359.
[32] T. Joachims, T. Finley, and C.-N.-J. Yu, "Cutting-plane training of structural SVMs," *Mach. Learn.*, vol. 77, no. 1, pp. 27–59, May 2009.
[33] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.
[34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[35] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
[36] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[37] Y. Wang, F. Yan, K. Zhu, B. Chen, and H. Wu, "A new practical robust control of cable driven manipulators using time-delay estimation," *Int. J. Robust Nonlinear Control*, vol. 29, no. 11, pp. 3405–3425, Apr. 2019.
[38] C. A. Micchelli and M. Pontil, "Learning the kernel function via regularization," *J. Mach. Learn. Res.*, vol. 6, pp. 1099–1125, Jul. 2005.
[39] L. Cao, H. Li, G. Dong and R. Lu, "Event-triggered control for multi-agent systems with sensor faults and input saturation," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published, doi: 10.1109/TSMC.2019.2938216.
[40] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "More efficiency in multiple kernel learning," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 775–782.
[41] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "Non-sparse regularization and efficient training with multiple kernels," vol. 186, pp. 189–190, 2010. [Online]. Available: https://arxiv.org/abs/1003.0079
[42] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. Int. Conf. Comput. Learn. Theory*. Berlin, Germany: Springer, 2001, pp. 416–426.
[43] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, pp. 1453–1484, Sep. 2005.
[44] V. N. Vapnik and V. Vapnik, *Statistical Learning Theory*, vol. 1. New York, NY, USA: Wiley, 1998.
[45] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2007-001, 2007.
[46] J. van de Weijer, C. Schmid, and J. Verbeek, "Learning color names from real-world images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[47] R. Khan, J. van de Weijer, F. S. Khan, D. Muselet, C. Ducottet, and C. Barat, "Discriminative color descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2866–2873.
[48] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
[49] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and Web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 157–173, Oct. 2007.
[50] R. C. E. Gonzalez, S. L. Woods, R. E. R. E. R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*, no. 4. New Delhi, India: Pearson, 2004.

QIANG WANG received the B.S. degree in computer science and technology from Jilin University, China, in 2011, and the M.S. and Ph.D. degrees in computer science and technology from the National University of Defense Technology, China, in 2013 and 2018, respectively. He is currently a Research Assistant with the School of Computer, National University of Defense Technology. His research interests include high-performance computing, information security, and machine learning.

XINWANG LIU received the B.S. degree in computer science and technology from Chongqing Technology and Business University, Chongqing, in 2006, and the M.S. and Ph.D. degrees in computer science and technology from the National University of Defense Technology, China, in 2008 and 2013, respectively. In October 2010, he spent one year in visiting the engineering and computer science at The Australia National University, supported by the China Scholarship Council. From November 2011 to October 2012, he was a Visiting Student with the School of Computer Science and Software Engineering, University of Wollongong. He is currently an Associate Professor with the College of Computer, National University of Defense Technology. His research interests focus on kernel learning and feature selection.

JIAQING XU received the B.S., M.S., and Ph.D. degrees in computer science and technology from the National University of Defense Technology, China, in 2005, 2007, and 2012, respectively. He is currently a Research Assistant with the College of Computer, National University of Defense Technology. His research interests focus on high-performance computing and high-speed interconnect.