Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 346496, 13 pages
http://dx.doi.org/10.1155/2015/346496

Research Article

Online Multikernel Learning Based on a Triple-Norm Regularizer for Semantic Image Classification

Shuangping Huang,[1,2] Lianwen Jin,[1] and Yunyu Li[3]

[1] *School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China*
[2] *School of Engineering, South Agricultural University of China, Guangzhou 510642, China*
[3] *Department of Computer and Information Science, University of Macau, Macau*

Correspondence should be addressed to Lianwen Jin; lianwen.jin@gmail.com

Received 19 January 2015; Revised 9 April 2015; Accepted 13 April 2015

Academic Editor: Dan Simon

Copyright © 2015 Shuangping Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Currently image classifiers based on multikernel learning (MKL) mostly use batch approach, which is slow and difficult to scale up for large datasets. In the meantime, standard MKL model neglects the correlations among examples associated with a specific kernel, which makes it infeasible to adjust the kernel combination coefficients. To address these issues, a new and efficient multikernel multiclass algorithm called TripleReg-MKL is proposed in this work. Taking the principle of strong convex optimization into consideration, we propose a new triple-norm regularizer (TripleReg) to constrain the empirical loss objective function, which exploits the correlations among examples to tune the kernel weights. It highlights the application of multivariate hinge loss and a conservative updating strategy to filter noisy samples, thereby reducing the model complexity. This novel MKL formulation is then solved in an online mode using a primal-dual framework. A theoretical analysis of the complexity and convergence of TripleReg-MKL is presented. It shows that the new algorithm has a complexity of $O(CMT)$ and achieves a fast convergence rate of $O(\log T/T)$. Extensive experiments on four benchmark datasets demonstrate the effectiveness and robustness of this new approach.

1. Introduction

Image semantic classification is a challenging task in computer vision field. Researchers are constantly searching for efficient learning methods with good scalability to categorize large and complex image datasets [1–6]. Among the current approaches used for image categorization, multikernel learning (MKL) [7–11] has been the subject of many recent studies and it delivers the state-of-the-art performance by solving a joint optimization problem, which comprises sample coefficients for the base kernel classifier and the optimal weights for combining multiple kernels associated with multiple clues [5].

However, most MKL methods use batch learning methods [6, 12–16], which are slow at classification, and they do not scale up well with large training datasets. To this end, various online methods have been proposed to facilitate efficient learning and real-time application [17–23]. These different online MKL methods involve different regularization techniques and updating rules. For example, Hoi et al. [17] used a Perceptron algorithm [23] to learn about a base classifier for a given kernel before applying the Hedge algorithm [24] to combine multiple classifiers in a linear manner. However, regularization was not considered in this formulation. Cavallanti et al. [22] proposed an ℓ_p-norm multiview perception algorithm, where differences in the clue-related kernel space were neglected.

Given the complex parameter structures of multikernel models, more researchers are using mixed norm regularization items to integrate sophisticated prior knowledge and to handle the parameter mutation caused by data noise. The $(2,1)$-norm was first proposed as a regularizer for multiclass MKL [9]. This approach induced absolute sparsity in the domain of the kernels, but it might weaken the convexity of the optimization problem or lead to poor performance [25]. Thus, a general type of group norm, $(2,p)$-norm $(1 \le p \le 2)$, was proposed in [19, 26] to provide greater flexibility when tuning the level of sparsity required for a task. However, the algorithm has difficulty achieving convergence when p is close to 1. In addition, an elastic net form of regularization is available for MKL, which allows the solution to obtain exact mathematical zeros and is effective for filtering invalid kernels [20, 27]. In summary, the aforementioned norms impose a constraint on kernels or classes, whereas they neglect the correlations among examples.

In this paper, a new algorithm called TripleReg-MKL is proposed. It defines a triple-norm regularizer (abbreviated as TripleReg) with strong convexity to constrain the empirical risk upon the current incoming samples. An online solution is derived using primal-dual framework for this new MKL formulation. As the correlations among examples are considered in TripleReg-MKL to tune the sparsity of model, the updating of kernel weights involves the historical cumulative effects of the overall online training procedure [19, 20]. It also highlights the combination of multivariate hinge loss and a conservative updating strategy to filter noisy samples. A theoretical derivation is presented and an analysis of the complexity and convergence is conducted as well. Extensive experiments delivered on four benchmark datasets verify the claims.

2. TripleReg-MKL Algorithm

2.1. Multiclass Multikernel Problem. Suppose that we are given a set of training samples $\mathbf{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where each sample is represented as an instance-label pair $(\mathbf{x}_i, y_i)$. Here, $y_i \in \mathbf{Y} = \{1, \ldots, C\}$, $C \geq 2$, denotes the label of a sample. The instance $\mathbf{x}_i = [\mathbf{x}_i^1, \ldots, \mathbf{x}_i^M]$ denotes the corresponding M measurements of features and each $\mathbf{x}_i^m$ $(m = 1, \ldots, M)$ is a multivariate feature vector, which describes a visual characteristic of the ith sample. Then, the multiclass classifier is defined by

$$\mathbf{x} \longmapsto \tilde{y}(\mathbf{x}) = \arg\max_{y \in \mathbf{Y}} f_y(\mathbf{x}), \tag{1}$$

where $f_y(\mathbf{x})$ is the value of the score function when the instance $\mathbf{x}$ is assigned to the class y. $\tilde{y}$ is the predicted class for which the function achieves the highest score. Based on a consideration of multikernel integration, the score function $f_y(\mathbf{x})$ is defined as

$$f_y(\mathbf{x}) = \sum_{m=1}^{M} \langle \boldsymbol{\omega}^{m,y}, \Phi^m(\mathbf{x}, y) \rangle = \langle\langle \boldsymbol{\omega}^{\cdot,y}, \boldsymbol{\Phi}(\mathbf{x}, y) \rangle\rangle, \tag{2}$$

where $\boldsymbol{\omega}^{\cdot,y} = (\boldsymbol{\omega}^{1,y}; \boldsymbol{\omega}^{2,y}; \ldots; \boldsymbol{\omega}^{M,y})$ $(y \in \mathbf{Y})$ are model parameters and $\boldsymbol{\Phi} = (\Phi^1; \ldots; \Phi^M)$ is the nonlinear mapping function that transforms the M features into M arbitrary high-dimensional reproducing kernel Hilbert space (RKHS). $\langle\langle \cdot, \cdot \rangle\rangle$ is the inner product operation between matrices $\mathbf{A}$ and $\mathbf{B}$, which is defined by

$$\langle\langle \mathbf{A}, \mathbf{B} \rangle\rangle = \sum_i \sum_j a_{ij} \times b_{ij}. \tag{3}$$

In the multiclass setup, the concept of class is introduced in the definition of a mapping function. This is different from traditional kernel machines, which ignore the class label information in the kernel definition. Specifically, we define

$$\Phi^m(\mathbf{x}, y) = \left(\mathbf{0}, \ldots, \mathbf{0}, \underbrace{\psi^m(\mathbf{x})}_{y}, \mathbf{0}, \ldots, \mathbf{0} \right), \quad \forall m = 1, \ldots, M, \tag{4}$$

where $y \in \mathbf{Y}$, $\psi^m(\cdot)$ is a label-free feature map [19]. Correspondingly, the model parameter comprises C blocks in each feature space; that is, $\boldsymbol{\omega}^{m,\cdot} = (\boldsymbol{\omega}^{m,1}, \ldots, \boldsymbol{\omega}^{m,C})$. That is to say, both $\boldsymbol{\omega}$ and Φ carry class information and feature clues. Therefore,

$$\langle \boldsymbol{\omega}^{m,\cdot}, \Phi^m(x, y) \rangle = \langle \boldsymbol{\omega}^{m,y}, \psi^m(x) \rangle. \tag{5}$$

Now, the goal is to learn about the multiple score function $f_y(\mathbf{x})$ parameterized by C parameter matrices, each of which is denoted by $\boldsymbol{\omega}^{\cdot,c} = (\boldsymbol{\omega}^{1,c}, \ldots, \boldsymbol{\omega}^{M,c})$, $c = 1, \ldots, C$.

To obtain the solution of the optimization problem, the primal objective function that needs to be minimized is defined as

$$\rho(\boldsymbol{\omega}) = c_t F(\boldsymbol{\omega}) + \min_{\boldsymbol{\omega}} \sum_{t=1}^{T} \ell_t(\boldsymbol{\omega}), \tag{6}$$

where $F(\boldsymbol{\omega})$ is the triple-norm regularizer (TripleReg) used to measure the complexity of $\boldsymbol{\omega}$ and to constrain the problem in a low-complexity domain. The second item is the global loss that accumulates hinge losses over all possible samples in the training set. $\ell_t(\boldsymbol{\omega})$ is the instantaneous loss function that measures the discrepancy between the predicted answer and the correct answer. Specifically, it can be denoted by $\ell(\boldsymbol{\omega}, (x_t, y_t))$ at the tth iteration. Here a multivariate hinge-loss function with convexity is defined for multiclass categorization as follows:

$$\ell(\boldsymbol{\omega}, (x_t, y_t)) = \Bigg[1 - \Bigg(\langle\langle \boldsymbol{\omega}^{\cdot,y_t}, \Phi(x_t, y_t) \rangle\rangle - \max_{y \neq y_t, y_i \in \mathbf{Y}} \langle\langle \boldsymbol{\omega}^{\cdot,y}, \Phi(x_t, y) \rangle\rangle \Bigg) \Bigg]_+ . \tag{7}$$

c_t $(c_t \geq 0)$ is a parameter that trades off the significance between the empirical loss and regularization item. From (6), the essence of this optimization problem is to learn about the optimal weight $\boldsymbol{\omega}$ to minimize the cumulative loss that occurs during the sequence of observations under regularization.

2.2. Triple-Norm Regularizer. According to Section 2.1, each component of $\boldsymbol{\omega}$ associated with a specific class and kernel, that is, $\boldsymbol{\omega}^{m,j}$ $(m = 1, \ldots, M,\ j = 1, \ldots, C)$, is a coefficient vector. It inherently implies the triple structure of the model parameter. Thus, the triple-norm-based regularization is designed as

$$F(\boldsymbol{\omega}) = \frac{1}{2} \|\boldsymbol{\omega}\|_{2,p_1,p_2}^2, \tag{8}$$

where

$$\|\boldsymbol{\omega}\|_{2,p_1,p_2} := \left\| \left(\left\| \left(\left\|\boldsymbol{\omega}^{1,1}\right\|_2, \ldots, \left\|\boldsymbol{\omega}^{1,C}\right\|_2 \right) \right\|_{p_1}, \ldots, \left\| \left(\left\|\boldsymbol{\omega}^{M,1}\right\|_2, \ldots, \left\|\boldsymbol{\omega}^{M,C}\right\|_2 \right) \right\|_{p_1} \right) \right\|_{p_2} \tag{9}$$

and $p_1, p_2 \in (1, 2]$. From (8), the regularizer (TripleReg) is strongly convex due to the square form of the triple norms and the value range of p_1, p_2. In (9), $\|\cdot\|_2$ is applied to each $\boldsymbol{\omega}^{m,j}$ $(m = 1, \ldots, M,\ j = 1, \ldots, C)$ that indicates sample information to obtain an $M \times C$ real number matrix $\boldsymbol{\omega}_{M\times C}$. Then, $\|\cdot\|_{p_1}$ is applied to each column of $\boldsymbol{\omega}_{M\times C}$ to obtain a vector in $\mathbb{R}^C$ upon which the norm $\|\cdot\|_{p_2}$ is applied to yield the value of $\|\boldsymbol{\omega}\|_{2,p_1,p_2}$. This TripleReg is actually a combination norm of three ℓ_p, each of which is imposed on sample-, class-, or kernel-related coefficients. The selection of the values of the parameters p_1 and p_2 allows the sparsity level of the solution to be determined in a flexible manner. The convexity of TripleReg with respect to $\|\cdot\|_{2,p_1,p_2}$ is proved and its argument $\sigma = 4/(1 + 1/(p_1 - 1) + 1/(p_2 - 1))$ is derived as well in Appendix A.

2.3. Online Solution Using a Primal-Dual Framework. A primal-dual algorithmic framework [28–30] is adopted to derive the optimal solution of (6).

Suppose that $\overline{\boldsymbol{\omega}}_0$ is the notation of the optimal fixed solution to the minimization problem of (6), which is both objective and imaginary. It may be considered objective because it can be selected retrospectively from a class of hypotheses based on complex and varied concepts of progress toward an acceptable competing hypothesis using the entire sequence of training data pairs [28]. It can be considered imaginary because it may require a long training period to be objective. $\overline{\boldsymbol{\omega}}_t$ is the actual model parameter at the tth iteration. The algorithm is expected to satisfy $\overline{\boldsymbol{\omega}}_t = \overline{\boldsymbol{\omega}}_0$ at each iteration. Therefore, (6) of the primal domain can be rewritten with constraints:

$$\begin{aligned} &\inf_{\overline{\omega}_0} \left(\frac{c_t}{2} \|\overline{\boldsymbol{\omega}}_0\|^2_{2,p_1,p_2} + \sum_{i=1}^{t} \ell(\overline{\boldsymbol{\omega}}_i) \right) \\ &\text{s.t.} \quad \overline{\boldsymbol{\omega}}_0 \in S, \quad \forall t \in [T],\ \overline{\boldsymbol{\omega}}_t = \overline{\boldsymbol{\omega}}_0. \end{aligned} \tag{10}$$

In (10), S is the set of all possible hypotheses. Introducing the Lagrangian multiplier λ_i $(i = 1, \ldots, T)$ yields

$$\begin{aligned} &\varsigma(\overline{\boldsymbol{\omega}}_0, \overline{\boldsymbol{\omega}}_1, \ldots, \overline{\boldsymbol{\omega}}_T, \lambda_1, \ldots, \lambda_T) \\ &\quad = \frac{c_t}{2} \|\overline{\boldsymbol{\omega}}_0\|^2_{2,p_1,p_2} + \sum_{i=1}^{t} \ell(\overline{\boldsymbol{\omega}}_i) + \sum_{i=1}^{t} \langle \lambda_i, \overline{\boldsymbol{\omega}}_0 - \overline{\boldsymbol{\omega}}_i \rangle. \end{aligned} \tag{11}$$

Based on the definition of the conjugate function

$$f^*(\theta) = \sup_{\omega} (\langle \omega, \theta \rangle - f(\omega)) \tag{12}$$

and the equation of $\inf f(x) = -\sup(-f(x))$, the dual objective function can be obtained:

$$\begin{aligned} &D(\lambda_1, \ldots, \lambda_T) \\ &\quad = \inf_{\overline{\omega}_0 \in S, \lambda_1, \ldots, \lambda_T} \varsigma(\overline{\boldsymbol{\omega}}_0, \overline{\boldsymbol{\omega}}_1, \ldots, \overline{\boldsymbol{\omega}}_T, \lambda_1, \ldots, \lambda_T) \\ &\quad = \sup c_t \left\{ \left\langle \overline{\boldsymbol{\omega}}_0, -\frac{1}{c_t} \sum_{i=1}^{t} \lambda_i \right\rangle - F(\overline{\boldsymbol{\omega}}_0) \right\} \\ &\qquad - \sum_{i=1}^{t} \sup (\langle \overline{\boldsymbol{\omega}}_i, \lambda_i \rangle - \ell(\overline{\boldsymbol{\omega}}_i, (x_i, y_i))) \\ &\quad = -c_t \cdot F^* \left(-\frac{1}{c_t} \sum_{i=1}^{t} \lambda_i \right) - \sum_{i=1}^{t} \ell^*(\lambda_i). \end{aligned} \tag{13}$$

Up to now, the constrained quadratic programming problem in the primal domain, as in (10), has been converted into a dual objective function, as in (13).

Since

$$F^* \left(-\frac{1}{c_t} \sum_{i=1}^{t} \lambda_i \right) = \sup \left\langle \overline{\boldsymbol{\omega}}_0, -\frac{1}{c_t} \sum_{i=1}^{t} \lambda_i \right\rangle - F(\overline{\boldsymbol{\omega}}_0), \tag{14}$$

we obtain the following equation after differentiating both sides of (14) with respect to $-(1/c_t) \sum_{i=1}^{t} \lambda_i$:

$$\overline{\boldsymbol{\omega}}_0 = \nabla F^* \left(-\frac{1}{c_t} \sum_{i=1}^{t} \lambda_i \right). \tag{15}$$

Similarly, since

$$\ell^*(\lambda_i) = \langle \overline{\boldsymbol{\omega}}_i, \lambda_i \rangle - \ell(\overline{\boldsymbol{\omega}}_i, (x_i, y_i)), \tag{16}$$

then,

$$\lambda_i = \partial \ell(\overline{\boldsymbol{\omega}}_i, (x_i, y_i)). \tag{17}$$

We denote the sum of the current t Lagrangian multipliers by $\overline{\boldsymbol{\theta}}_t$; that is,

$$\overline{\boldsymbol{\theta}}_t = -\frac{1}{c_t} \sum_{i=1}^{t} \lambda_i, \tag{18}$$

where $\overline{\boldsymbol{\theta}}_t$ is the dual variable with the same triple structure as the primal variable $\overline{\boldsymbol{\omega}}_t$. Thus, the solution of the primal objective is obtained in the dual domain, which is formulated as $\overline{\boldsymbol{\omega}}_0 = \nabla F^*(\overline{\boldsymbol{\theta}}_t)$. Specifically, the dual norm of (8) is formulated as

$$F^*(\overline{\boldsymbol{\theta}}_t) = \frac{1}{2} \left\|\overline{\boldsymbol{\theta}}_t\right\|^2_{2,q_1,q_2}, \tag{19}$$

where $1/p_1 + 1/q_1 = 1/p_2 + 1/q_2 = 1$.

A summary of the algorithmic framework of TripleReg-MKL is shown in Algorithm 1.

Algorithm 1 shows that the TripleReg-MKL algo -rithm updates the weight $\overline{\boldsymbol{\omega}}_t$ through line (11). Here,

```
TripleReg-MKL
(1) Input $p_1, p_2 \in (1,2], T$;
(2) Initialize $\overline{\boldsymbol{\theta}}_0 = 0, \overline{\boldsymbol{\omega}}_0 = 0, c_0$
(3) For $t = 1, \ldots, T$ do
(4)     Receive a new instance $\mathbf{x}_t$ randomly from $\mathbf{S}$
(5)     Make a prediction $\tilde{y}_t = \arg\max_{y\in\mathbf{Y}} \langle\langle \overline{\boldsymbol{\omega}}_t, \overline{\boldsymbol{\Phi}}(x_t, y)\rangle\rangle$
(6)     Obtain the correct label $y_t$
(7)     Compute inference label, that is, $\hat{y}_t = \arg\max_{y\neq y_t} \langle\langle \overline{\boldsymbol{\omega}}_t, \overline{\boldsymbol{\Phi}}(x_t, y)\rangle\rangle$
(8)     if $\ell(\overline{\boldsymbol{\omega}}_t, x_t, y_t) > 0$
(9)         $\partial\ell(\overline{\boldsymbol{\omega}}_t, (x_t, \cdot)) = \begin{cases} -\overline{\boldsymbol{\Phi}}(x_t, y_t) \\ \overline{\boldsymbol{\Phi}}(x_t, \hat{y}_t) \\ 0, \text{others} \end{cases}$
(10)        $\overline{\boldsymbol{\theta}}_{t+1} = \frac{1}{c_t}(\overline{\boldsymbol{\theta}}_t + \partial\ell_t)$;
(11)        $\boldsymbol{\omega}_{t+1}^{m,j} = \|\overline{\boldsymbol{\theta}}\|_{2,q1,q2}^{2-q2} \|\boldsymbol{\theta}^m\|_{2,q1}^{q2-q1} \|\boldsymbol{\theta}^{m,j}\|_2^{q1-2} \boldsymbol{\theta}_{t+1}^{m,j}$;   $\forall j = 1, \ldots, C;\ m = 1, \ldots, M$
(12)    end if
(13) end for
```

Algorithm 1: Pseudocode of the TripleReg-MKL algorithm. Parameters $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$ are initially set to zero as online learning algorithms usually do [31]. It is intuitive that model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$ are empty at the starting time when no sample arrives.

$\|\overline{\boldsymbol{\theta}}\|_{2,q1,q2}^{2-q2} \|\boldsymbol{\theta}^m\|_{2,q1}^{q2-q1} \|\boldsymbol{\theta}^{m,j}\|_2^{q1-2}$ is denoted by η_{t+1}. Thus, the updating rule can be simplified to $\boldsymbol{\omega}_{t+1}^{m,j} = \eta_{t+1}\boldsymbol{\theta}_{t+1}^{m,j}$, which indicates that $\boldsymbol{\omega}$ is determined by the kernel coefficient η and the dual parameter $\boldsymbol{\theta}$. The relationship between the parameters during online learning is shown in Figure 1.

In Figure 1, the kernel weight is updated using information from the newly arriving sample and from all previous samples. It allows each sample to make different contributions to the model. In other words, the correlations among samples are introduced to tune the level of sparsity in the domain of the kernels because of the close relationship and high similarity among samples in the same class.

It is imperative for us to apply the kernel trick to avoid the difficult definition of $\Phi(x, y)$ and the expensive calculation of the inner product in a high-dimensional transformation space [32] for the derivation of TripleReg-MKL algorithm. By setting $a_t = \text{sign}(\partial\ell(\overline{\boldsymbol{\omega}}_t))$ and according to the relationship of $K^m(x, x') = \langle\Phi^m(x, y), \Phi^m(x', y')\rangle$, the inner product between $\boldsymbol{\omega}_{t+1}^{m,j}$ and $\Phi^m(x, y)$ can be calculated as

$$\begin{aligned} \left\langle \boldsymbol{\omega}_{t+1}^{m,j}, \Phi^m(x, y) \right\rangle &= \left\langle \eta_{t+1}\boldsymbol{\theta}_{t+1}^{m,j}, \Phi^m(x, y) \right\rangle = \Big\langle \eta_{t+1} \\ &\cdot \frac{1}{c_t}(\lambda_1 + \cdots + \lambda_t), \Phi^m(x, y) \Big\rangle = \eta_{t+1} \\ &\cdot \frac{1}{c_t}\{\text{sign}(\partial\ell(\overline{\boldsymbol{\omega}}_1))\Phi^m(x_1, y_1) + \cdots \\ &+ \text{sign}(\partial\ell(\overline{\boldsymbol{\omega}}_t))\Phi^m(x_t, y_t)\} \cdot \Phi^m(x, y) = \eta_{t+1} \\ &\cdot \frac{1}{c_t}(a_1 K^m(x_1, x) + \cdots + a_t K^m(x_t, x)). \end{aligned} \tag{20}$$

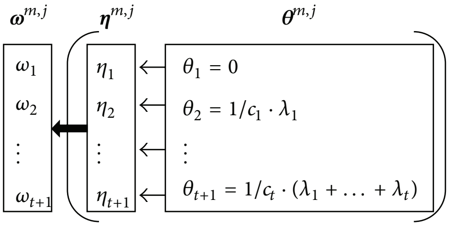

$\boldsymbol{\omega}^{m,j}$
$\boldsymbol{\eta}^{m,j}$
$\boldsymbol{\theta}^{m,j}$
ω_1
ω_2
$\vdots$
ω_{t+1}
η_1
η_2
$\vdots$
η_{t+1}
$\theta_1 = 0$
$\theta_2 = 1/c_1 \cdot \lambda_1$
$\vdots$
$\theta_{t+1} = 1/c_t \cdot (\lambda_1 + \ldots + \lambda_t)$

Figure 1: Relationships between the parameters during online training.

During the training process, TripleReg-MKL algorithm applies a conservative updating strategy. That is to say, the updating is only implemented in the current and the interference class model when the loss function is greater than zero, as shown in line (8) in Algorithm 1. The "one positive and one negative" approach is also used to increase the gap between the correct model and the max interference model.

Algorithm 1 shows that the time required by the TripleReg-MKL is dominated by line (5) in each iteration, which has a complexity of $O(CMT)$ in the worst case. C, M, and T are the numbers of classes, kernels, and previous samples, respectively. This complexity is common to other state-of-the-art online learning MKL algorithms, such as OM-2 and UFO-MKL.

3. Convergence Analysis

In this section, we analyze a theoretical guarantee of the convergence rate of the TripleReg algorithm. Theorem 1 is derived from the regret bound of primal-dual optimization in Theorem 2 in [29] (the proof of Theorem 1 is given in Appendix B).

Theorem 1. $\ell_1, \ldots, \ell_T$ *represents the sequence of the function. For all* $t \in [T]$, $\ell_t = c_t f + g_t$ *and* f *is the* σ*-strongly convex function of* $\|\cdot\|$. $\|\cdot\|_*$ *is the dual norm of* $\|\cdot\|$. w *is the optimal solution to the model. If we set* $R \geq \max_i \|\partial g_i(w_i)\|_*$ *and* $w_t = \nabla f^*(-(1/c_{t-1}) \sum_{i=1}^{t-1} \partial g_i(w_i))$, *then*

$$\frac{1}{T}\sum_{t=1}^{T}\ell_t\left(w_t\right) - \min_{w}\frac{1}{T}\sum_{t=1}^{T}\ell_t\left(w\right) \leq \frac{R^2}{2\sigma T}\left(\log T + 1\right). \quad (21)$$

In the TripleReg-MKL algorithm, $F(\boldsymbol{\omega})$ in (8) is a $\sigma = 4p_2p_3/(p_2p_3 + q_2p_3 + q_3p_2)$-strongly convex function with respect to the norm $\|\cdot\|_{2,p_1,p_2}$ (see Appendix A). Suppose that $X = \max_{j=1,\ldots,M}|\Phi^j(x_t,\cdot)|$; then the gradient of the multiclass hinge-loss function defined in (7) satisfies

$$\begin{aligned} &\left\|\partial\ell\left(\overline{\boldsymbol{\omega}}_t,\left(x_t,y_t\right)\right)\right\|_{2,q_1,q_2} \\ &\quad \leq 2^{1/q_1}M^{1/q_2}\max_{j=1,\ldots,M}\left|\Phi^j\left(x_t,\cdot\right)\right| \leq 2^{1/q_1}M^{1/q_2}X \qquad (22) \\ &\quad \leq \sqrt{2M}X, \end{aligned}$$

where $1/q_1, 1/q_2 \in (0,1/2]$. This means that the upper bound of $\|\partial\ell(\overline{\boldsymbol{\omega}}_t,(x_t,y_t))\|_{2,q_1,q_2}$ is $R = \sqrt{2M}X$.

The Markov inequality ($P[Z \geq a] \leq E(Z)/a$) [33] was introduced in consideration of the random choice of the sample sequence during the online learning procedure. Let $\delta \in (0,1)$; inequality (23) will be satisfied with a probability of at least $1 - \delta$ over the choice of a random sample after T iterations of the TripleReg-MKL algorithm:

$$H\left(\overline{\boldsymbol{\omega}}_{T+1}\right) - H\left(\overline{\boldsymbol{\omega}}_0\right) \leq \frac{MX^2\left(1 + \log T\right)}{\delta\sigma T}, \quad (23)$$

where $H(\overline{\omega}) = (c_{T+1}/2)\|\overline{\boldsymbol{\omega}}\|_{2,p_1,p_2}^2 + \ell_{T+1}(\overline{\boldsymbol{\omega}})$ and $\overline{\boldsymbol{\omega}}_0$ is the optimal hypothetical solution to (6) (see Appendix C). Equation (23) shows that the upper bound of algorithm convergence decreases gradually when T increases infinitely. That is to say, the model parameter becomes increasingly close to the optimal hypothetical solution with increasing number of iterations.

4. Experiment

Experimental evaluation of TripleReg-MKL is presented in terms of classification performance and capacity to combine features. A comparison with four state-of-the-art online MKL algorithms, that is, OM-2 [19], UFO-MKL [20], OMCL [21], and Perceptron [23], is performed on the benchmark Caltech-101 [34], Caltech-256 [35], Oxford Flowers (102) [36], and MNIST [37] datasets. Caltech-101 [34] is a collection of 9144 images from 102 object categories. The number of images in each category varies from 40 to 800. Most of the object categories contain 50 images. Caltech-256 [35] is an extension of Caltech-101 containing 29781 images from 256 object categories. The minimum, average, and maximum number of images in each category are 80, 119, and 827, respectively. Oxford Flowers (102) [36] contains 8189 images that cover 102 flower categories. Each class contains 40 to 258 images. MNIST [37] is a large dataset of 60000 training examples and 10000 test examples from 10 handwritten digit categories. The digits have been size-normalized to 28 × 28 gray-scale images and they are centered in the fixed size images. This dataset is good for testing learning techniques using real-world data since it requires minimal preprocessing and formatting effort. These four datasets are characterized by their high image diversity, large sample volumes and number of categories, or great vagueness among classes, which presents great challenges for classification. The codes of the three comparison algorithms are obtained from DOGMA [31].

Complex but effective features including self-similarity (SSIM) [38], geometric blur (GB) [39], CSIFT [40–43], and Oriented-PDF [44] were applied to the medium or large class datasets of Caltech-101 and Caltech-256. For the same reasons stated in a previous study [45], SPHOG [46], local binary pattern (LBP) [47], and GIST [48] were used to describe the handwritten digits of MNIST. For Oxford Flowers (102), a χ^2-distance matrix [36, 49] was used to measure the similarity associated with four different features of flowers, that is, "D_SIFTint," "D_SIFTbdy," "D_HSV histogram," and "D_HOG." The corresponding kernel matrix was computed using $\exp(-\gamma^{-1}d(x,x'))$, where d was the distance and γ was the kernel parameter determined by cross-validation.

4.1. Experimental Setup. Thirty images of each category were selected randomly for training from Caltech-101 and Caltech-256, and the rest were used for testing. For the Oxford Flowers (102) dataset, the predefined training and testing splits recommended in previous studies [36, 49] were used in this experiment: that is, only 10 of each class are from Oxford Flowers (102). Unless stated otherwise, the experimental process was replicated 10 times using a different random test set or sample sequence. The averages and standard deviations are reported. To obtain better experimental results, model parameters such as p_1 and p_2 were determined using a fivefold cross-validation procedure. Given the fact that online learning has a relatively slow convergence rate, we attempted to increase the training dataset by cycling the training examples through multiple epochs [45].

4.2. Experiment Results

4.2.1. Comparing the Effect of Using Single Kernels or Combining-All. In this experiment, TripleReg-MKL with a combined kernel is compared to that of a single kernel. The experiment results are shown in Table 1.

Table 1 shows that "combining-all" using the TripleReg-MKL algorithm results in significant improvement in the classification performance for any dataset compared with a single kernel. For example, the test accuracy increased by approximately 9.38% and 10.0% on the two large object class datasets, that is, Caltech-101 and Caltech-256, compared with that obtained using the best single kernel. With the Oxford

Table 1: Comparison of the results obtained with single kernel or combining-all.

(a) Caltech-101 and Caltech-256

Accuracy (%)	GB	SSIM	C-SIFT	Oriented-PDF	Combining-all
Caltech-101	81.13 ± 0.02	77.49 ± 0.06	76.45 ± 0.08	80.62 ± 0.14	**88.74** ± 0.09
Caltech-256	34.33 ± 0.05	33.92 ± 0.07	33.99 ± 0.10	45.13 ± 0.15	**49.64** ± 0.08

(b) Oxford Flowers (102)

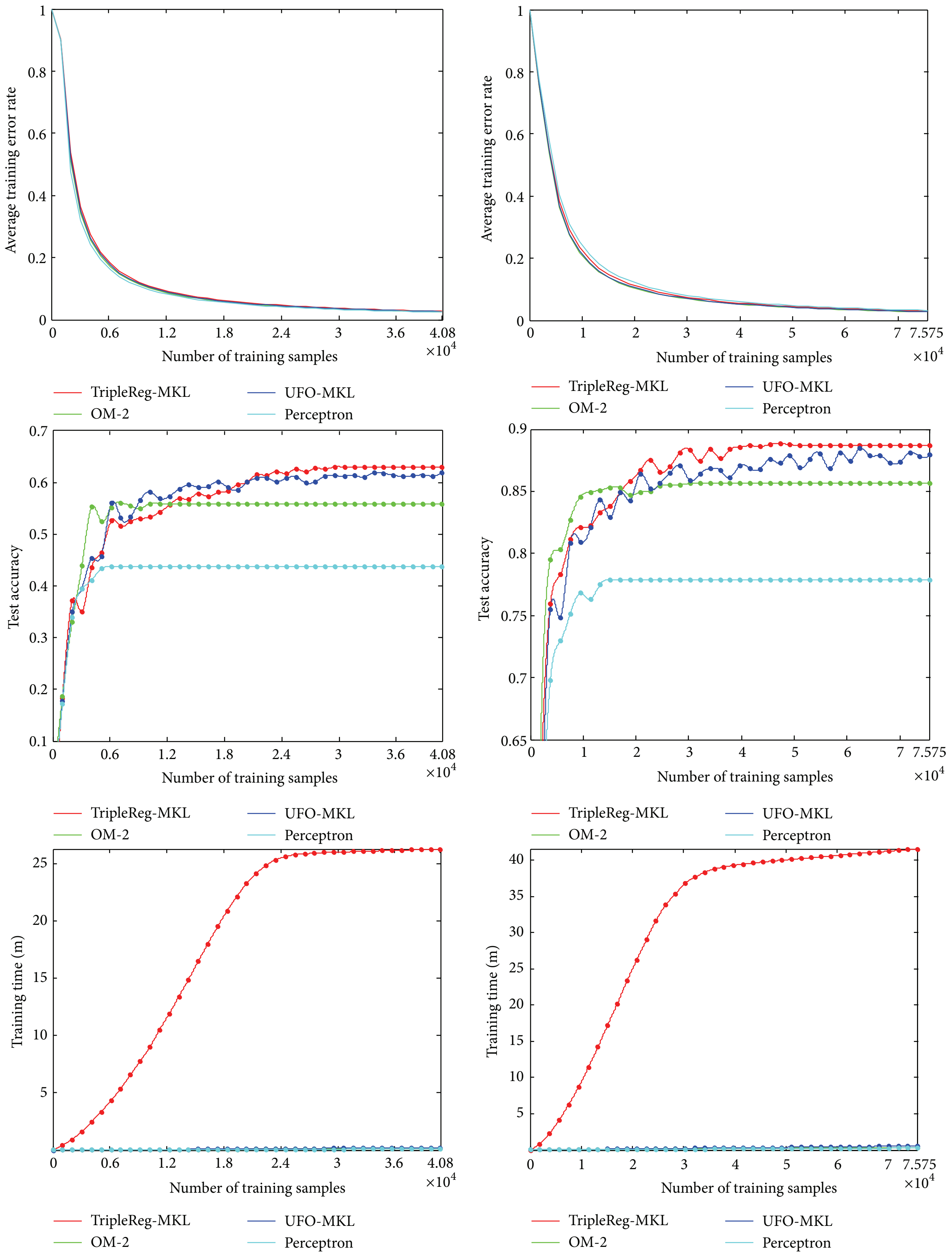
Feature	Accuracy (%)
D_HSV	32.28 ± 0.16
D_HOG	39.12 ± 0.12
D_SIFTint	46.11 ± 0.44
D_SIFTbdy	27.05 ± 0.66
Combining-all	**62.51** ± 0.13

(c) MNIST

Feature	Error rate (%)
SPHOG	0.72 ± 0.02
GIST	0.88 ± 0.06
LBP	15.27 ± 0.13
Combining-all	**0.65** ± 0.02

Flowers (102) dataset, "combining-all" outperformed the best single kernel by approximately 35.57%. For the largest scale dataset MNIST, "error rate" is used as the performance index to clarify the numerical comparison. It is observed that a single SPHOG had a low error rate of 0.72% with MNIST, but "combining-all" resulted in a lower error rate of 0.65%. That is to say, 9.72% reduction in the error rate is achieved using TripleReg-MKL algorithm to fuse all the features of SPHOG, GIST, and LBP.

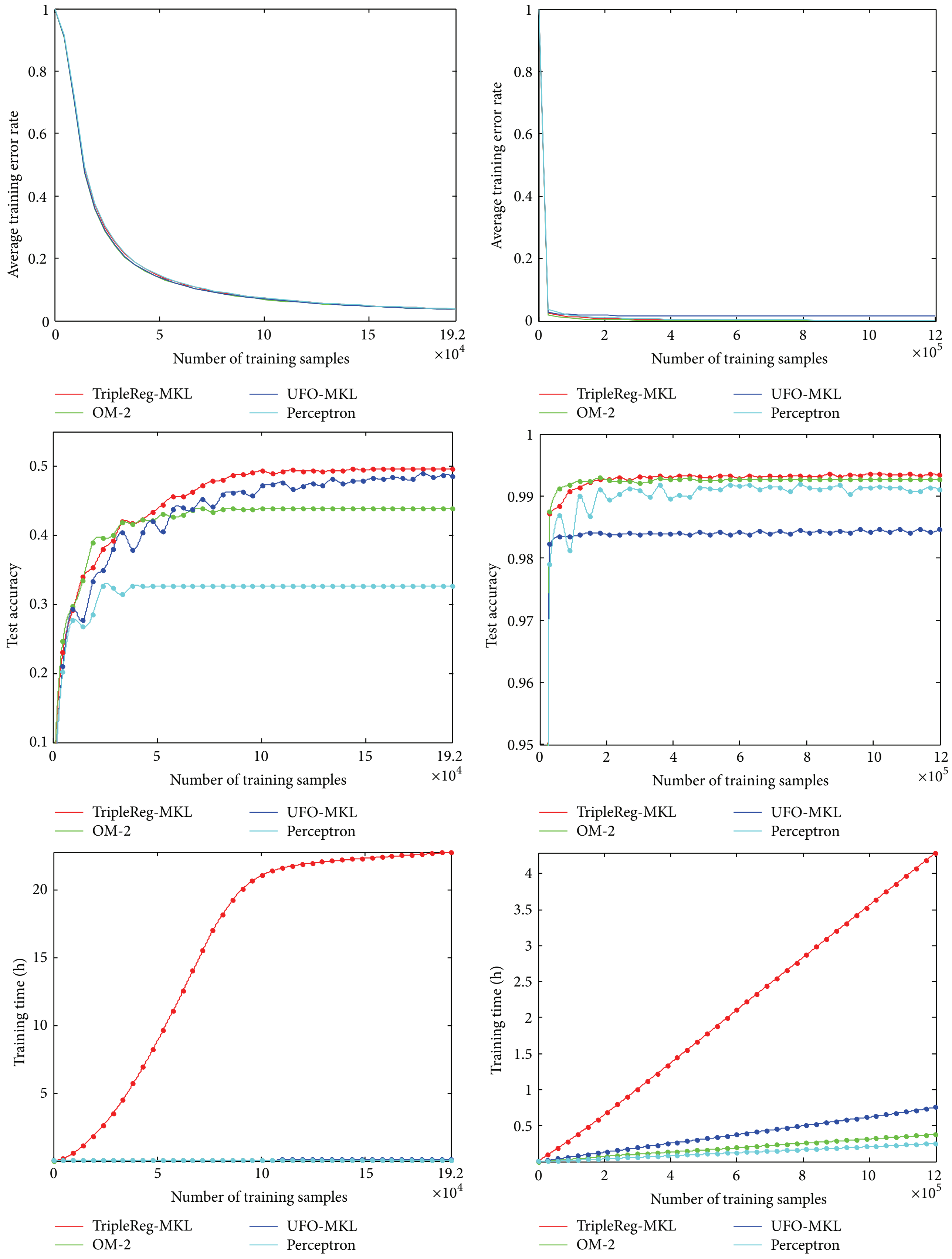
4.2.2. Comparing with the State-of-the-Art Online MKL Methods. We compared the performance of TripleReg-MKL (the source code is available at https://github.com/huangshuangping/TripleReg-MKL) with three state-of-the-art online multiclass MKL algorithms, that is, OM-2 [19], UFO-MKL [20], and Perceptron [23]. Figures 2(a)–2(d) show the training sample size versus average training error rate, training sample size versus test accuracy, and training sample size versus training time curves for the four benchmark datasets. From the figure of "training sample size versus average training error rate," the online training error rate of all of the four algorithms had the same trend with an increasing number of iterations. That is to say, the average training error rate decreased sharply as the number of iterations increased during the early part of the online training period and it gradually stabilized around zero as the learning process continued. This implied that no new prediction errors appeared with the subsequent training examples from the epoch repetitions after the model reached the steady state. In contrast, the curve of "training sample size versus test accuracy" tends to rise with increasing numbers of iterations during online updating after early variability until it reaches a steady state. This trend agrees with the gradual optimization law of online learning. A comparison of the results obtained with all four approaches showed that TripleReg-MKL had the best classification performance after reaching the steady state. That is indicated by the position of the red curve above all of the other curves. The UFO-MKL algorithm ranked the second in terms of classification performance with Caltech-101, Caltech-256, and Oxford Flowers (102). However, it had poor stability, which is demonstrated by the fluctuation in the blue curve. This occurred because UFO-MKL adopts an elastic net form of regularization, thus providing a direct kernel reset. This type of sparsity approach can simplify the model significantly. In the meantime, it may cause significant fluctuations in the performance and slow convergence due to its aggressive reset policy. OM-2 and Perceptron delivered far lower accuracy on the three datasets with more than 100 classes compared with TripleReg-MKL. By contrast, TripleReg-MKL achieved a slightly better performance than OM-2 on MNIST that has relatively simple examples and small classes. This indicates the high adaptability of TripleReg-MKL to relatively large class problems. As for the curve of "training sample size versus training time," it shows the runtime performance as a function of the number of training samples. It can be seen that OM-2, UFO-MKL, and Perceptron are faster than TripleReg-MKL for all four datasets. This can be explained by the fact that our TripleReg-MKL algorithm considers information from not only the newly arriving sample but also all previous historical samples to update kernel weights. Therefore, TripleReg-MKL sacrifices runtime performance to achieve a superior classification performance. In addition, the trend curve of the training time is different for each dataset. This is because the quality and distribution of training samples from the four datasets vary, and these factors determine the number of noisy samples to be filtered and thus the runtime. Regardless, it is also observed from Figure 2 that the TripleReg-MKL algorithm is well suited to



(a) Oxford Flowers (102)

(b) Caltech-101

FIGURE 2: Continued.



(c) Caltech-256

(d) MNIST

Figure 2: Performance of different online learning algorithms as a function of the number of training examples: average online training error rate (top), classification rate using the test set (middle), and training time (bottom) (s denotes seconds, m denotes minutes, and h denotes hours).

Table 2: Test accuracy at the last iteration.

Accuracy (%)	TripleReg-MKL	OM-2	UFO-MKL	Perceptron
Oxford Flowers (102)	**62.56**	55.91	61.95	43.75
Caltech-101	**88.75**	85.69	87.98	77.85
Caltech-256	**49.70**	43.81	48.98	32.59
MNIST	**99.38**	99.27	98.49	99.10

Table 3: Comparison of the classification accuracies (%) on the Caltech-101 and Caltech-256 datasets.

Accuracy (%)	Caltech-101	Caltech-256
MKL-SRC [14]	75.7	—
MKSR [15]	82.9	46.9
SM1MKL [16]	88.42	47.7
SM2MKL [16]	88.51	**49.76**
Proposed	**88.75**	49.70

learning with more than one million samples. To be noted, the data size of million is simulated by means of multipass strategy in the experiment.

Given the fact that some of the results cannot be seen clearly in Figure 2, the test accuracy of the four algorithms at the final iteration is presented in Table 2. It can be seen that test accuracy close to 50% was achieved using only about 25% of the images from Caltech-256 for training. It is the best performance achieved using MKL methods on the Caltech-256 dataset to the best of our knowledge. The test accuracy of 88.75% on Caltech-101 is also the highest result obtained with an online algorithm. Oxford Flowers (102) is characterized by its great variation within classes and vague gaps between classes. Only 12.5% of the images were used for training and an accuracy of 62.56% was obtained using TripleReg-MKL on this benchmark dataset.

4.2.3. Comparing with the State-of-the-Art Batch MKL Methods. We compared the performance of TripleReg-MKL with some recent batch MKL methods including MKL-SRC [14], MKSR [15], and Soft Margin Multiple Kernel Learning [16] (abbreviated as SM1MKL and SM2MKL, corresponding to different setup of hinge loss and squared hinge loss, resp.). The comparison experiments are delivered on Caltech-101 and Caltech-256 as the literatures [14, 15] provide the results for the different training conditions on these two benchmark datasets. We download SMMKL implementation code (https://sites.google.com/site/xinxingxu666/) and use one-versus-all strategy for its multiclass extension. The optimal SVM regularization parameters for SM1MKL and SM2MKL are searched within $[1, 100]$ using median method. To be concrete, the optimal SVM regularization parameter for SM1MKL and SM2MKL is set as 10 and 2.5, respectively. Table 3 shows the test accuracy of TripleReg-MKL and all the three batch MKL baseline algorithms. Comparison is relatively striking between TripleReg-MKL and MKL-SRC or MKSR on both datasets, demonstrating the generation performance of our algorithm. A similar case is with comparison between TripleReg-MKL and SM1MKL on the larger classes of Caltech-256 object dataset. The superiority is on the contrary not obvious when comparing test accuracy of TripleReg-MKL with that of SM1MKL and SM2MKL on Caltech-101. To summarize, the proposed TripleReg-MKL compares well with the state-of-the-art batch MKL methods in terms of test accuracy.

5. Conclusions

In this paper, a new TripleReg-MKL algorithm was proposed. In this approach, a novel triple-norm regularizer (TripleReg) was designed for MKL and an efficient online solution is derived. TripleReg introduced a constraint among correlated examples and makes the kernel weight updated using all previous historical sample information. It yielded greater flexibility for tuning the level of sparsity in the domain of kernels. This online solution allows the algorithm to be readily adapted to learning cases with millions of cases in an efficient manner. To examine the empirical performance of the proposed TripleReg-MKL algorithm, extensive experiments were conducted on a testbed using four diverse real datasets. Experiment results verified its high capacity for heterogeneous feature fusion, which is particularly important for recognizing large classes with crowded spaces and vague gaps between classes. It also achieved the highest classification performance compared with four state-of-the-art online MKL methods, that is, OM-2, UFO-MKL, OMCL, and Perceptron.

Appendices

A. Proof of TripleReg's Strong Convexity

Let $\mathbf{X} = (\mathbf{X}_1 \ \ \mathbf{X}_2 \ \ \cdots \ \ \mathbf{X}_z)$ be a three-dimensional real matrix that comprises z pages. The jth page $\mathbf{X}_j = (\mathbf{X}_j^1 \ \ \mathbf{X}_j^2 \ \ \cdots \ \ \mathbf{X}_j^n)$ is a matrix of $m \times n$ and each column $\mathbf{X}_j^i \in R^m$. By denoting $\|\mathbf{X}_j\|_{p_1,p_2}$ by $\|\mathbf{X}_j\|_{p_1,p_2} := \|(\|\mathbf{X}_j^1\|_{p_1}, \dots, \|\mathbf{X}_j^n\|_{p_1})\|_{p_2}$, the triple-norm $\|\mathbf{X}\|_{p_1,p_2,p_3}$ will be defined as $\|\mathbf{X}\|_{p_1,p_2,p_3} := \|(\|\mathbf{X}_1\|_{p_1,p_2}, \dots, \|\mathbf{X}_z\|_{p_1,p_2})\|_{p_3}$. Based on the definition of the triple norm, we define the triple-norm regularizer, that is, TripleReg $F : \mathbf{R}^{m \times n \times z} \rightarrow \mathbf{R}$, as

$$F(\mathbf{X}) = \frac{1}{2} \|\mathbf{X}\|_{p_1,p_2,p_3}^2, \qquad \text{(A.1)}$$

where $p_1, p_2, p_3 \in (1, 2]$. The Fenchel-conjugate function of (A.1) is

$$F^*(\mathbf{X}) = \frac{1}{2} \|\mathbf{X}\|_{q_1,q_2,q_3}^2, \qquad \text{(A.2)}$$

where $1/p_1 + 1/q_1 = 1/p_2 + 1/q_2 = 1/p_3 + 1/q_3 = 1$.

Next, we present the analysis of the strong convexity of (A.1) and its argument. The proof begins by giving some mathematical definition and tools, which is followed by the derivation of the strong convexity argument.

Definition A.1. A function f is β-strongly smooth with respect to a norm $\|\cdot\|$, if f is differentiable everywhere and if, for all $\mathbf{x}, \mathbf{y} \in \{\mathbf{v} : f(\mathbf{v}) < \infty\}$ and $\alpha \in (0,1)$, one has

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \geq \alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y}) - \frac{1}{2}\beta\alpha(1-\alpha)\|\mathbf{x}-\mathbf{y}\|^2. \tag{A.3}$$

The following theorem states that strong convexity and strong smoothness are dual properties [50].

Theorem A.2 (strong convexity/strong smoothness duality). *Assume that f is a closed and convex function. The Fenchel conjugate of f is denoted by f^*. Then, f is β-strongly convex with respect to a norm $\|\cdot\|$ if and only if f^* is $1/\beta$-strongly smooth with respect to the dual norm $\|\cdot\|_*$.*

The following lemma relates to the strong convexity of the vector l_p norm. Its proof is standard and can be found, for example, in a study by Kakade et al. [50].

Lemma A.3. *Let $p \in (1,2]$. A function $f : \mathbf{R}^d \rightarrow \mathbf{R}$ defined as $f(\omega) = \|\omega\|_p^2$ is $2(p-1)$-strongly convex with respect to $\|\cdot\|_p$.*

Let $Q_i = \|\mathbf{X}\|_{p_i}^2$ ($p_i \in (1,2]$, $i \in \{1,2,3\}$) be absolutely symmetric norms on $\mathbf{R}^m, \mathbf{R}^n, \mathbf{R}^z$, respectively. Their dual norms are $Q_i^* = \|\mathbf{X}\|_{q_i}^2$, where $1/p_i + 1/q_i = 1$. Using the duality properties between strong convexity and smoothness stated in Theorem A.2, Q_i^* are σ_i-strongly smooth with respect to $\|\cdot\|_{q_i}$, where their constants are $\sigma_i = 1/2(p_i - 1)$.

Next, we analyze the smoothness of (A.1) and obtain the argument. Thus, we need to prove

$$\begin{aligned}&\|\alpha\mathbf{X} + (1-\alpha)\mathbf{Y}\|_{q_1,q_2,q_3}^2 \\ &\quad \geq \alpha\|\mathbf{X}\|_{q_1,q_2,q_3}^2 + (1-\alpha)\|\mathbf{Y}\|_{q_1,q_2,q_3}^2 \\ &\quad - \frac{1}{2}(\sigma_1+\sigma_2+\sigma_3)\alpha(1-\alpha)\|\mathbf{X}-\mathbf{Y}\|_{q_1,q_2,q_3}^2.\end{aligned} \tag{A.4}$$

According to Theorem 13 in [50], the bi-norm regularizer function $Q_{1,2}^* = \|\mathbf{X}\|_{q_1,q_2}^2$ is $\sigma_1 + \sigma_2$-strongly smooth with respect to $\|\mathbf{X}\|_{q_1,q_2}$. Following the equivalent definition of strong smoothness in (A.3), we obtain

$$\begin{aligned}&Q_{1,2}^* : \|\alpha\mathbf{X} + (1-\alpha)\mathbf{Y}\|_{q_1,q_2}^2 \\ &\quad \geq \alpha\|\mathbf{X}\|_{q_1,q_2}^2 + (1-\alpha)\|\mathbf{Y}\|_{q_1,q_2}^2 \\ &\quad - \frac{1}{2}(\sigma_1+\sigma_2)\alpha(1-\alpha)\|\mathbf{X}-\mathbf{Y}\|_{q_1,q_2}^2.\end{aligned} \tag{A.5}$$

Combining the smoothness of Q_3^*, that is,

$$\begin{aligned}&Q_3^* : \|\alpha\mathbf{X} + (1-\alpha)\mathbf{Y}\|_{q_3}^2 \\ &\quad \geq \alpha\|\mathbf{X}\|_{q_3}^2 + (1-\alpha)\|\mathbf{Y}\|_{q_3}^2 \\ &\quad - \frac{1}{2}\sigma_3\alpha(1-\alpha)\|\mathbf{X}-\mathbf{Y}\|_{q_3}^2,\end{aligned} \tag{A.6}$$

the left side of (A.4) can be rewritten as

$$\begin{aligned}\|\alpha\mathbf{X} + (1-\alpha)\mathbf{Y}\|_{q_1,q_2,q_3}^2 &= \left\|\ldots, \left\|\alpha\mathbf{X}_j + (1-\alpha)\mathbf{Y}_j\right\|_{q_1,q_2}, \ldots\right\|_{q_3}^2 = \left\|\ldots, \underbrace{\sqrt{\left\|\alpha\mathbf{X}_j + (1-\alpha)\mathbf{Y}_j\right\|_{q_1,q_2}^2}}_{}, \ldots\right\|_{q_3}^2 \\ &\geq \left\|\ldots, \sqrt{\alpha\left\|\mathbf{X}_j\right\|_{q_1,q_2}^2 + (1-\alpha)\left\|\mathbf{Y}_j\right\|_{q_1,q_2}^2 - \frac{1}{2}(\sigma_1+\sigma_2)\alpha(1-\alpha)\left\|\mathbf{X}_j - \mathbf{Y}_j\right\|_{q_1,q_2}^2}, \ldots\right\|_{q_3}^2 \\ &\geq \left\|\ldots, \sqrt{\alpha\left\|\mathbf{X}_j\right\|_{q_1,q_2}^2 + (1-\alpha)\left\|\mathbf{Y}_j\right\|_{q_1,q_2}^2}, \ldots\right\|_{q_3}^2 \\ &\quad - \frac{1}{2}(\sigma_1+\sigma_2)\alpha(1-\alpha)\left\|\ldots, \sqrt{\left\|\mathbf{X}_j - \mathbf{Y}_j\right\|_{q_1,q_2}^2}, \ldots\right\|_{q_3}^2 \\ &= \left\|\ldots, \sqrt{\alpha\left\|\mathbf{X}_j\right\|_{q_1,q_2}^2 + (1-\alpha)\left\|\mathbf{Y}_j\right\|_{q_1,q_2}^2}, \ldots\right\|_{q_3}^2 - \frac{1}{2}(\sigma_1+\sigma_2)\alpha(1-\alpha)\|\mathbf{X}-\mathbf{Y}\|_{q_1,q_2,q_3}^2.\end{aligned} \tag{A.7}$$

Next, for any $x, y \geq 0$ and $\alpha \in [0,1]$, we have $\sqrt{\alpha x^2 + (1-\alpha) y^2} \geq \alpha x + (1-\alpha) y$. Thus, we can obtain

$$\begin{aligned} &\left\| \ldots, \sqrt{\alpha \left\| \mathbf{X}_j \right\|_{q_1,q_2}^2 + (1-\alpha) \left\| \mathbf{Y}_j \right\|_{q_1,q_2}^2}, \ldots \right\|_{q_3}^2 \\ &\quad \geq \underbrace{\left\| \ldots, \alpha \left\| \mathbf{X}_j \right\|_{q_1,q_2} + (1-\alpha) \left\| \mathbf{Y}_j \right\|_{q_1,q_2}, \ldots \right\|_{q_3}^2} \\ &\quad \geq \alpha \left\| \mathbf{X} \right\|_{q_1,q_2,q_3}^2 + (1-\alpha) \left\| \mathbf{Y} \right\|_{q_1,q_2,q_3}^2 \\ &\qquad - \frac{1}{2} \sigma_3 \alpha (1-\alpha) \left\| \mathbf{X} - \mathbf{Y} \right\|_{q_1,q_2,q_3}^2 . \end{aligned} \tag{A.8}$$

Combining (A.7) and (A.8) proves (A.4). Thus, our dual triple-norm regularizer as (A.2) is a $(1/2)(\sigma_1 + \sigma_2 + \sigma_3)$-strongly smooth function. Based on the duality of the strong smoothness and convexity stated in Theorem A.2, the convexity of the triple-norm regularization in (A.1) is

$$\begin{aligned} \sigma &= \frac{2}{\sigma_1 + \sigma_2 + \sigma_3} \\ &= \frac{4}{1/(p_1 - 1) + 1/(p_2 - 1) + 1/(p_3 - 1)}. \end{aligned} \tag{A.9}$$

In particular, the strong convexity argument of the TripleReg, as in (8), is the same as (A.9). As $p_1 = 2$ and $1/p_i + 1/q_i = 1$ $(i = 1, 2, 3)$, (A.9) is equivalent to $4p_2p_3/(p_2p_3 + q_2p_3 + q_3p_2)$.

B. Proof of Theorem 1

To complete the proof, Theorem 2 in [29] is rewritten as follows.

Let $\ell_1, \ldots, \ell_T$ be a sequence of functions such that, for all $t \in [T] = \{1, 2, \ldots, T\}$ and $\ell_t = c_t f + g_t$, where f is σ-strongly convex with respect to a norm $\|\cdot\|$, g_t is a convex and closed function. Then, any algorithm that can be derived from "template algorithm for online strongly convex optimization" [29] satisfies

$$\sum_{t=1}^{T} \ell_t(\omega_t) - \min_{\omega} \sum_{t=1}^{T} \ell_t(\omega) \leq \frac{1}{2} \sum_{t=1}^{T} \frac{\|v_t\|_*^2}{c_{1:t}}, \tag{B.1}$$

where $v_t = \partial g_t(\omega_t)$ and $\|\cdot\|_*$ is the norm dual to $\|\cdot\|$.

If we let $R = \max_t \|v_t\|_*$ and suppose that $c_t \geq \sigma$ $(t \in [T])$, the right side of (B.1) yields the following conclusion:

$$\begin{aligned} \frac{1}{2} \sum_{t=1}^{T} \frac{\|v_t\|_*^2}{c_{1:t}} &\leq \frac{1}{2} \sum_{t=1}^{T} \frac{R^2}{c_{1:t}} = \frac{R^2}{2} \sum_{t=1}^{T} \frac{1}{c_{1:t}} \\ &= \frac{R^2}{2} \sum_{t=1}^{T} \frac{1}{c_1 + \cdots + c_t} \leq \frac{R^2}{2} \sum_{t=1}^{T} \frac{1}{t\sigma} \\ &= \frac{R^2}{2} \left(\sum_{t=1}^{T} \frac{1}{t\sigma} \right) = \frac{R^2}{2\sigma} \left(\sum_{t=1}^{T} \frac{1}{t} \right) \\ &\leq \frac{R^2}{2\sigma} (\log T + 1). \end{aligned} \tag{B.2}$$

Combining (B.1) and (B.2) yields

$$\sum_{t=1}^{T} \ell_t(\omega_t) - \min_{\omega} \sum_{t=1}^{T} \ell_t(\omega) \leq \frac{R^2}{2\sigma} (\log T + 1). \tag{B.3}$$

Theorem 1 is proved by dividing the left and right sides of (B.3) by T.

C. Proof of the Convergence Rate

Assume that $H(\overline{\boldsymbol{\omega}}) = (c_{T+1}/2) \|\overline{\boldsymbol{\omega}}\|_{2,p_1,p_2}^2 + \ell_{T+1}(\overline{\boldsymbol{\omega}})$ and $\overline{\boldsymbol{\omega}}_0$ is the optimal hypothetical solution to (6). Theorem 1 can be rewritten in a probabilistic sense as

$$E_t \left[H(\overline{\boldsymbol{\omega}}_t) - H(\overline{\boldsymbol{\omega}}_0) \right] \leq \frac{R^2}{2\sigma T} (\log T + 1). \tag{C.1}$$

The Markov inequality is as follows:

$$P[Z \geq a] \leq \frac{E(Z)}{a}, \tag{C.2}$$

where a is a constant.

In this case, we denote $H(\overline{\boldsymbol{\omega}}_t) - H(\overline{\boldsymbol{\omega}}_0)$ by a random variable Z and we set $E[Z]/a = \delta$. Then, the upper bound of a is obtained as

$$a = \frac{E[Z]}{\delta} \leq \frac{R^2 (\log T + 1)}{2\delta\sigma T}. \tag{C.3}$$

Since $P[Z \geq a] \leq E[Z]/a = \delta$, then $P[Z \leq a] \geq 1 - \delta$. Thus, the following inequality holds with a probability of at least $1 - \delta$ over the choice of random samples:

$$H(\overline{\boldsymbol{\omega}}_t) - g(\overline{\boldsymbol{\omega}}_0) \leq \frac{R^2}{2\delta\sigma T} (\log T + 1). \tag{C.4}$$

Plugging $R = \sqrt{2M}X$ into (C.4) yields the specific conclusion for the TripleReg-MKL algorithm; that is,

$$H(\overline{\boldsymbol{\omega}}_{T+1}) - H(\overline{\boldsymbol{\omega}}_0) \leq \frac{MX^2 (\log T + 1)}{\delta\sigma T}. \tag{C.5}$$

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported in part by the Guangdong Natural Science Funds (Grant no. S2013010014240), NSFC (nos. 31101087, 31201129, 61075021, and 61302120), National Science and Technology Support Plan (2013BAH65F01, 2013BAH65F04, and 2013BAJ13B05-01), GDSTP (no. 2012A010701001), and Research Fund for the Doctoral Program of Higher Education of China (nos. 20120172110023 and 20124404120005).

References

[1] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Metric learning for large scale image classification: generalizing to new classes at near-zero cost," in *Proceedings of the European Conference on Computer Vision*, pp. 488–501, 2012.

[2] C.-Y. Yeh, W.-P. Su, and S.-J. Lee, "Employing multiple-kernel support vector machines for counterfeit banknote recognition," *Applied Soft Computing*, vol. 11, no. 1, pp. 1439–1447, 2011.

[3] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher Kernel for large-scale image classification," in *Computer Vision—ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5–11, 2010, Proceedings, Part IV*, vol. 6314 of *Lecture Notes in Computer Science*, pp. 143–156, Springer, Berlin, Germany, 2010.

[4] B.-Y. Liu, F. Sadeghi, M. Tappen, O. Shamir, and C. Liu, "Probabilistic label trees for efficient large scale image classification," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 843–850, June 2013.

[5] X.-Z. Qi and Q. Wang, "An image classification approach based on sparse coding and multiple kernel learning," *Acta Electronica Sinica*, vol. 40, no. 4, pp. 773–779, 2012.

[6] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *The Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[7] E. Al Daoud and H. Turabieh, "New empirical nonparametric kernels for support vector machine classification," *Applied Soft Computing*, vol. 13, no. 4, pp. 1759–1765, 2013.

[8] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proceedings of the 21st International Conference on Machine Learning (ICML '04)*, pp. 41–48, July 2004.

[9] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, pp. 1–8, October 2007.

[10] A. Zien and C. S. Ong, "Multiclass multiple kernel learning," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, pp. 1191–1198, June 2007.

[11] L. Jie, T. Tommasi, and B. Caputo, "Multiclass transfer learning from unconstrained priors," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1863–1870, IEEE, Barcelona, Spain, November 2011.

[12] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simple MKL," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[13] Z. Wang and X. Sun, "Multiple kernel local Fisher discriminant analysis for face recognition," *Signal Processing*, vol. 93, no. 6, pp. 1496–1509, 2013.

[14] A. Shrivastava, V. M. Patel, and R. Chellappa, "Multiple kernel learning for sparse representation-based classification," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 3013–3024, 2014.

[15] J. J. Thiagarajan, K. N. Ramamurthy, and A. Spanias, "Multiple kernel sparse representations for supervised and unsupervised learning," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2905–2915, 2014.

[16] X. Xu, I. W. Tsang, and D. Xu, "Soft margin multiple kernel learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 749–761, 2013.

[17] S. C. Hoi, R. Jin, P. Zhao, and T. Yang, "Online multiple kernel classification," *Machine Learning*, vol. 90, no. 2, pp. 289–316, 2013.

[18] R. Jin, S. C. H. Hoi, and T.-B. Yang, "Online multiple Kernel learning: algorithms and mistake bounds," in *Algorithmic Learning Theory: 21st International Conference, ALT 2010, Canberra, Australia, October 6–8, 2010. Proceedings*, vol. 6331 of *Lecture Notes in Computer Science*, pp. 390–404, Springer, Berlin, Germany, 2010.

[19] J. Luo, F. Orabona, M. Fornoni, B. Caputo, and N. Cesa-Bianchi, "OM-2: an online multi-class multi-kernel learning algorithm," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW '10)*, pp. 43–50, June 2010.

[20] F. Orabona and J. Luo, "Ultra-fast optimization algorithm for sparse multi kernel learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML '11)*, pp. 249–256, July 2011.

[21] J. Luo, F. Orabona, and B. Caputo, "An online framework for learning novel concepts over multiple cues," in *Proceeding of the Asian Conference on Computer Vision*, pp. 269–280, Xi'an, China, September 2009.

[22] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, "Linear algorithms for online multitask classification," *Journal of Machine Learning Research*, vol. 11, pp. 2901–2934, 2010.

[23] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[24] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, part 2, pp. 119–139, 1997.

[25] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien, "Efficient and accurate l_p-norm multiple kernel learning," in *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS '09)*, pp. 997–1005, December 2009.

[26] F. Orabona, J. Luo, and B. Caputo, "Online-batch strongly convex multi kernel learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 787–794, IEEE, San Francisco, Calif, USA, June 2010.

[27] R. Tomioka and T. Suzuki, "Sparsity-accuracy trade-off in MKL," in *Proceedings of the NIPS Workshop: Understanding Multiple Kernel Learning Methods*, 2009.

[28] S. Shalev-Shwartz, *Online learning: theory, algorithms, and applications [Ph.D. thesis]*, Hebrew University, 2007.

[29] S. M. Kakade and S. Shalev-Shwartz, "Mind the duality gap: logarithmic regret algorithms for online optimization," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 1457–1464, 2008.

[30] S. M. Kakade, S. Shalev-Shwartz, and A. Tewari, "On the duality of strong convexity and strong smoothness: learning applications and matrix regularization," Tech. Rep., TTI, 2009.

[31] DOGMA, MATLAB toolbox, http://dogma.sourceforge.net/.

[32] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, 2008.

[33] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan, "Stochastic convex optimization," in *Proceedings of the 22nd Conference on Learning Theory (COLT '09)*, June 2009.

[34] F. F. Li, R. Fergus, and P. Perona, "Caltech 101 datasets," 2003, http://www.vision.caltech.edu/Image_Datasets/Caltech101/.

[35] G. Griffin, A. Holub, and P. Perona, "Caltech 256 object category dataset," Tech. Rep. UCB/ CSD-04-1366, California Institue of Technology, 2007, http://www.vision.caltech.edu/Image_Datasets/Caltech256/images/.

[36] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proceedings of the 6th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP '08)*, pp. 722–729, December 2008.

[37] The MNIST DATABASE of handwritten digits, http://yann.lecun.com/exdb/mnist/index.html.

[38] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, June 2007.

[39] A. C. Berg, T. L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 26–33, June 2005.

[40] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[41] A. E. Abdel-Hakim and A. A. Farag, "CSIFT: a SIFT descriptor with color invariant characteristics," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 1978–1983, June 2006.

[42] J.-M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, S. Member, and H. Geerts, "Color invariance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 12, pp. 1338–1350, 2001.

[43] G. J. Burghouts and J.-M. Geusebroek, "Performance evaluation of local colour invariants," *Computer Vision and Image Understanding*, vol. 113, no. 1, pp. 48–62, 2009.

[44] T. Kobayashi, "BFO meets HOG: feature extraction based on histograms of oriented p.d.f. gradients for image classification," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 747–754, June 2013.

[45] S.-P. Huang, L.-W. Jin, Y. Fang, and X.-X. Wei, "Online heterogeneous feature fusion machines for visual recognition," *Neurocomputing*, vol. 123, pp. 100–109, 2014.

[46] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR '07)*, pp. 401–408, July 2007.

[47] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multi-resolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[48] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[49] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 1447–1454, IEEE, June 2006.

[50] S. M. Kakade, S. Shalev-Shwartz, and A. Tewari, "Regularization techniques for learning with matrices," *Journal of Machine Learning Research*, vol. 13, pp. 1865–1890, 2012.

Hindawi

Submit your manuscripts at
http://www.hindawi.com

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization