Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2010, Article ID 461450, 13 pages
doi:10.1155/2010/461450

Research Article

Per-Sample Multiple Kernel Approach for Visual Concept Learning

Jingjing Yang,[1,2,3] Yuanning Li,[1,2,3] Yonghong Tian,[2] Ling-Yu Duan,[2] and Wen Gao[1,2]

[1] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China*
[2] *National Engineering Laboratory for Video Technology, School of EE & CS, Peking University, Beijing 100871, China*
[3] *Graduate University, Chinese Academy of Sciences, Beijing 100039, China*

Correspondence should be addressed to Yonghong Tian, yhtian@pku.edu.cn

Received 1 May 2009; Revised 22 November 2009; Accepted 19 January 2010

Academic Editor: Benoit Huet

Learning visual concepts from images is an important yet challenging problem in computer vision and multimedia research areas. Multiple kernel learning (MKL) methods have shown great advantages in visual concept learning. As a visual concept often exhibits great appearance variance, a canonical MKL approach may not generate satisfactory results when a uniform kernel combination is applied over the input space. In this paper, we propose a per-sample multiple kernel learning (PS-MKL) approach to take into account intraclass diversity for improving discrimination. PS-MKL determines sample-wise kernel weights according to kernel functions and training samples. Kernel weights as well as kernel-based classifiers are jointly learned. For efficient learning, PS-MKL employs a sample selection strategy. Extensive experiments are carried out over three benchmarking datasets of different characteristics including Caltech101, WikipediaMM, and Pascal VOC'07. PS-MKL has achieved encouraging performance, comparable to the state of the art, which has outperformed a canonical MKL.

1. Introduction

Visual concept learning is an important topic in image and video indexing and retrieval. Advanced machine learning techniques have been widely employed to map low-level visual features to visual concepts, such as scenes (e.g., indoor/outdoor [1], natural scenes [2]) and objects (e.g., airplane/motorbike/face) [3, 4]. Generally, a visual concept classifier is learnt from manually labeled images in a supervised manner and unseen images are categorized into one of the learnt concepts with a classifier. However, a well-trained concept classifier on a small dataset may not be expected to work fairly well on a much larger-scale image or video corpus due to the well-known semantic gap [5].

Learning visual concepts from numerous images is a challenging problem in real applications. For a concept, its image instances are often assumed to produce similarity in different attributes (e.g., scale, shape, color, and texture). As shown in Figure 1(a), several "*airplane*" samples exhibit good similarity in color and shape. On the other hand, an instance of a concept may produce various appearances due to the imaging issues like viewpoint, luminance, or occlusion. Moreover, different instances of a concept could produce intra-class variance in appearance (see Figure 1(b)) from the pattern classification point of view. In other words, training instances of a visual concept could be redundant while in different feature spaces a bag of instances would produce distinct intra-class variations. So we have to model the invariance as well as the intra-class diversity in appearance to train a concept classifier as shown in Figure 1.

To train a concept classifier, we would like to maximize the distance (or interval) between positive and negative samples. Several advanced learning techniques such as distance metrics [6] or kernels [7, 8] can be employed. Recently, multiple kernels learning (MKL) methods [9] have shown great advantages in visual concept learning [8, 10]. Instead of using a single kernel in a standard support vector machine (SVM) [11], MKL learns an optimal kernel combination as well as a classifier jointly, thereby selecting informative features and discriminative kernels. However, most of existing MKL methods apply a uniform similarity measure (i.e., a uniform kernel combination) over the input

(a)

(b)

Figure 1: (a) Samples of "*airplane*" in Caltech101 [15], and (b) samples of "*military aircraft*" in WikipediaMM [16].

space, so that the intra-class diversity is difficult to model when the instances of a concept is featured by significant appearance variance.

In this paper, we present a per-sample multiple kernel learning (PS-MKL) method that introduces a sample-wise kernel combination into an MKL framework. PS-MKL is to learn sample-wise multiple kernel combination for different training samples rather than for each concept uniformly. Different from most of sample-based methods [12–14], such sample-wise kernel combination works on the sparse samples consisting of a classifier's support vectors. In learning phases, the sample-wise kernel combination and the associated kernel-based classifiers are jointly optimized through solving a Max-Min problem. So the contributions of different kernels are learnt over individual training samples. The intra-class diversity is accordingly modeled by applying sample-wise kernel combinations.

In PS-MKL, the number of sample-wise kernel weights increases with the number of training samples. When a training set is large or even huge, PS-MKL would probably be deficient due to the high computational complexity. To reduce the computational complexity without losing discriminative power, we introduce an informative samples selection method.

Extensive comparison experiments are carried out over three benchmarking datasets including Caltech101 [15], WikipediaMM [16], and Pascal VOC'07 [17]. Although numerous object categories are involved in Caltech101, their images produce relatively less variation in pose or scale, which may not produce more complete challenges from a corpus of real-world images. So we extend our experiments to WikipediaMM datasets (some 150 k images crawled from Wiki with a wide coverage of concepts) and Pascal VOC'07 (some 10 k images from our daily life). We have achieved promising results comparable to the state-of-the-art methods [8, 10, 13, 14, 18, 19] on Caltech101. Moreover, we show a promising discriminative power of PS-MKL over real-world large-scale images corpus, that is, WikipediaMM and Pascal VOC'07.

Our main contributions are summarized as follow.

(i) We propose a novel PS-MKL approach to visual concept learning by applying kernel-based learning to model the intra-class diversity of a concept.

(ii) We provide a tractable solution for learning optimal sample-wise kernel combinations and kernel-based classifiers in a joint manner.

(iii) We present an effective sample selection method to reduce the computational complexity without losing the discriminative power of PS-MKL.

The remainder of this paper is organized as follows. Section 2 reviews the related work. In Section 3, we present the PS-MKL model. The learning procedure is detailed in Section 4. A sample selection approach for PS-MKL is presented in Section 5. The empirical results of object recognition and image retrieval are presented in Section 6. Future extensions are discussed in Section 7, followed by a conclusion in Section 8.

This paper is an extended version of [20]. The main extensions include a sample selection method for PS-MKL, the adding of two shape features, performance comparisons with the state-of-the-art methods and other multiple kernel-based method, and more extensive experiments on Pascal VOC'07 datasets.

2. Related Works

Many research efforts have been made in modeling visual concepts [8, 10, 13, 14, 18, 19, 21]. Below we review the related works in visual concepts learning by three categories: generative, distance-based, and kernel-based approaches.

2.1. Generative Methods. In earlier years generative approaches are prevalent in visual concept learning [3, 4, 15, 22]. A joint distribution of a concept and low level features is inferred by the Bayesian rule. In a generative model, latent variables can be introduced to fuse multiple cues. For example, part-based methods [22, 23] and bag of words methods [24] introduce a spatial variable to incorporate shape invariance of a visual concept. Ng and Jordan [25] have shown that in a 2-class setting a generative approach often outperforms a discriminative one over a small number of training samples.

A generative model is usually built up upon intermediate results precomputed from low level features, which would be limited in seeking an explicit representation of low-level features (e.g., shape, appearance, and texture). Due to the model complexity, a promising performance could not be guaranteed in large-scale concept learning.

2.2. Distance-Based Methods. On the other hand, researchers try to develop proper distance functions to distinguish visual characteristics of different concepts. In [13, 14], image-to-image or region-to-region distance functions are represented as a linear combination of various distances on different features. Likewise a weighted combination of different features is applied to compute distance functions. However, these methods focus on learning a uniform distance function for each concept. Like a canonical MKL, those distance-based works would be deficient in modeling the intra-class diversity of a concept. Recently, a sample-based distance function [12] is presented to measure the visual similarity between images (using appearance patches and shapes). A serious limitation in distance-based methods lies in that a distance function is usually based on an explicit feature representation whereas a desired representation, is often not available in generic concept learning.

2.3. Kernel-Based Methods. A kernel-based method is discriminative, which can effectively find the decision boundaries in a kernel space and generalize well on unseen data [26]. Generally speaking, a kernel-based method is advantageous in two aspects. First, a kernel explicitly defines a visual similarity measure between samples and implicitly represents the mapping from an input space to a feature space [11], thereby avoiding to seek an explicit feature representation and possible curse of dimension. Second, a kernel method can find out the optimal separating hyperplane between positive and negative samples efficiently by SVM.

Below we review kernel-based methods by two categories, namely, single kernel and multiple kernels.

2.3.1. Single Kernel Methods. In computer vision, various kernels have been carefully designed to measure different visual clues. A multiresolution histogram-based kernel is proposed in [27] to measure the image similarity at different granularities. A spatial pyramid matching kernel is proposed in [7] to enforce loose spatial information that allows the image similarity with local spatial coordinates. A kernel-based on the local feature distribution is presented in [28] to model the local context of an image. A kernel-based on the pyramid histogram of orientated gradients (PHOGs) is presented in [29] to capture the shape similarity by a spatial layout. These kernels are designed to operate on certain features that represent particular visual characteristics. Our idea is to incorporate kernels into a multikernel learning framework to systematically investigate the collaboration of different basic kernels in concept learning.

2.3.2. Multiple Kernel Methods. Much progress has been made in the field of multiple kernel learning [30, 31]. Most of MKL methods follow a similar framework as a linear combination of basic kernels but differ in the cost function to optimize. The combination of basic kernels helps to avoid a high-dimensional feature space from a simple catenation of low-level features. With MKL, different types of features can be formulated in a unifying formula to lower the risk of overfitting.

More recently, MKL has yielded promising results of learning visual concepts [8, 10]. In [8], six descriptors are combined optimally in a kernel learning framework. In [10], 12 kernels (i.e., PMK and SPK with different hyperparameters) are incorporated into an MKL framework. Like the canonical MKL [9], these methods adopt a uniform kernel combination strategy. That is, the weights of basic kernels are learned at the concept level, so that intra-class diversity is ignored in learning a concept classifier.

Our proposed PS-MKL is meant to keep the invariance in visual appearance and accommodate the intra-class diversity based on the sample images of a concept. In PS-MKL, classifier learning is combined with the optimization of sample-wise kernel combinations, which works at a finer granularity than most of the previous multiple kernel methods.

3. Per-Sample Multiple Kernel Learning

Without loss of generality, we cast visual concept detection to a binary classification problem, based on a given visual concept lexicon C. Let $L = \{x_i, y_i\}_{i=1}^{N}$ denote a training image dataset, where x_i is the ith sample, $y_i = \{\pm 1\}$ denotes the binary label of a given visual concept $c \in C$, and N is the number of training samples. Our objective is to train a kernel-based classifier $f_c(x)$ (we simply use $f(x)$ subsequently) to predict a visual concept c in image x.

Figure 2 depicts a three-tier flowchart of three related kernel-based methods (i.e., a standard SVM, a canonical MKL, and our proposed PS-MKL) to detect visual concepts

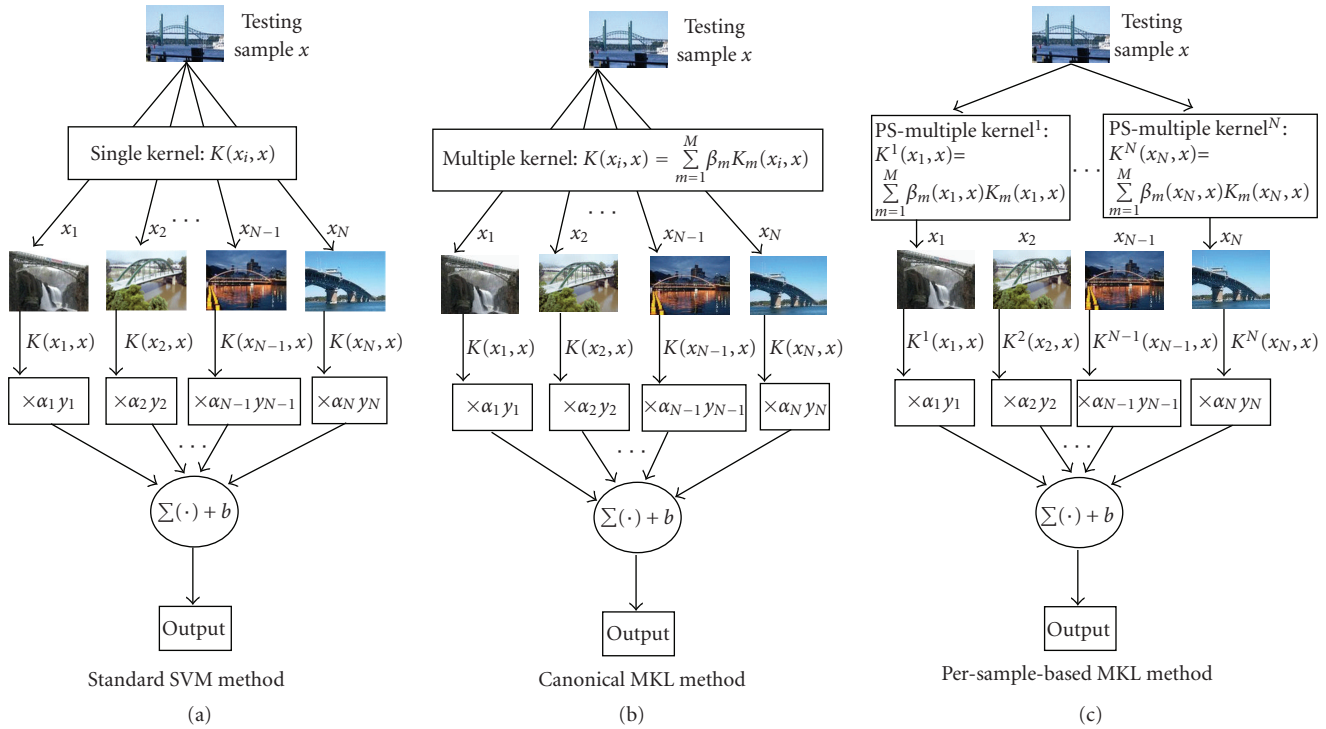

Figure 2: Three paradigms of learning visual concepts from images using: (a) Standard SVM method; (b) Canonical MKL method; (c) PS-MKL method.

in an unseen image. Three layers are involved, that is, input layer, middle layer, and decision layer. Three methods adopt a similar framework but differ in kernel structure. In the input layer, x represents a test sample to be fed into a kernel-based classifier. In the middle layer, the similarities between a test sample x and training samples $\{x_i\}_{i=1}^N$ are measured via different kernel structures, respectively. As shown in Figure 2(a), a standard SVM method employs a single kernel to measure the sample similarity. But both canonical MKL (see Figure 2(b)) and PS-MKL (see Figure 2(c)) combine multiple basic kernels to measure the similarities between samples. A canonical MKL employs a uniform multiple kernel combination whereas PS-MKL employs a sample-wise kernel combination. In PS-MKL, the kernel weights not only depend on basic kernel functions, but also on each sample pair to compare. In the bottom layer, a decision function is used to determine whether a test sample x contains a given concept.

In this section, we first briefly review a standard SVM and a canonical MKL in Section 3.1 and Section 3.2. And then PS-MKL is presented in Section 3.3.

3.1. A Standard SVM. A standard SVM sets up a separating hyperplane to classify samples where a feature map $\phi(x)$ is employed to project the original data from an input space to a feature space. To avoid an explicit representation $\phi(x)$ of the feature space, a so-called "kernel trick" is applied to define a kernel function $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$. For binary classification, the decision function of a standard SVM is expressed as follows:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b, \tag{1}$$

where $K(\cdot, \cdot)$ is a positive semidefinite kernel function associated with a reproducing kernel Hilbert space (RKHS), $\{\alpha_i\}_{i=1}^N$ and b are the classifier parameters to learn from L.

3.2. A Canonical MKL. It is well known that SVM is an efficient tool for solving a classification problem. However, its discriminative power heavily depends on the kernel function which is at most cases selected by cross-validation. Instead of using a single kernel, MKL [9] learns a linear kernel combination and the associated classifier simultaneously. In a canonical MKL, the multikernel combination is defined as

$$K(x_i, x) = \sum_{m=1}^{M} \beta_m K_m(x_i, x) \tag{2}$$

with $\sum_{m=1}^{M} \beta_m = 1$ and $\beta_m \geq 0$ for all m, where $K_m(\cdot, \cdot)$ is a positive semidefinite kernel function (referred to as a *basic kernel*), M is the total number of basic kernels, and $\{\beta_m\}_{m=1}^M$ are kernel weights to optimize during training. Each kernel $K_m(\cdot, \cdot)$ can employ different kernel functions based on different feature subsets or representations.

For binary classification, the decision function of a canonical MKL is given as follows:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \sum_{m=1}^{M} \beta_m K_m(x_i, x) + b, \tag{3}$$

where $\{\alpha_i\}_{i=1}^{N}$ and b are the classifier parameters like the parameters of decision function in a standard SVM. In MKL, the coefficients $\{\alpha_i\}_{i=1}^{N}$ and the kernel weights $\{\beta_m\}_{m=1}^{M}$ can be learnt by solving a joint optimization problem. Readers are referred to [9, 30, 31] for more details.

3.3. The Proposed PS-MKL. Instead of a uniform kernel combination in a canonical MKL, we propose a sample-based formulation of MKL, namely, PS-MKL.

In PS-MKL, the uniform multikernel combination in (2) can be rewritten as

$$K(x_i, x) = \sum_{m=1}^{M} \beta_m(x_i, x) K_m(x_i, x), \tag{4}$$

where $\beta_m(x_i, x)$is a *sample-wise kernel weight* with respect to x_i and x, rather than a fixed β_m in a canonical MKL. Then the decision function in (3) can be reformulated as

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \sum_{m=1}^{M} \beta_m(x_i, x) K_m(x_i, x) + b. \tag{5}$$

Accordingly, our goal is to optimize the coefficients $\boldsymbol{\alpha}$ and the *sample-wise kernel weights* $\boldsymbol{\beta}$ so as to construct a decision function $f(x)$.

It is worth to note that a basic kernel $K_m(x_i, x)$ can utilize distinct feature sets x_i and x to represent the similarity of images in different visual features. Basic kernels can take different kernel functions. In addition to the classical kernels (e.g., Gaussian and polynomial kernels) with different parameters, several specific kernels in image domain (e.g., SPK [7] and PDK [28]) are often preferred. Further details on basic kernels are discussed in the experiments (See Section 6.2).

4. Learning PS-MKL

In this section, we present how to learn the parameters of PS-MKL. In Section 4.1, we briefly describe the PS-MKL primal problem which is in spirit similar to the SVM primal problem. The consequent dual problem is described in Section 4.2, and the learning procedure is detailed in Section 4.3.

4.1. The PS-MKL Primal Problem. In PS-MKL, a sample x is translated by a feature mapping $\{\phi_m(x) \mapsto \mathbb{R}^{D_m}\}_{m=1}^{M}$ from an input space to M feature spaces $(\phi_1(x), \ldots, \phi_M(x))$, where $D_m$denotes the dimensionality of the mth feature space. Each feature map is associated with a weight vector $\mathbf{w}_m$. So we have the linear combination of those corresponding output functions:

$$f(x) = \sum_{m=1}^{M} \beta_m(x) \langle \mathbf{w}_m, \phi_m(x) \rangle + b, \tag{6}$$

where $\beta_m(x)$ is a mixing coefficient to count the contribution of feature map$\phi_m(x)$ for the classification task.

Inspired by the standard SVM learning process, training can be implemented by solving the following optimization problem which involves maximizing the margin between positive/negative training samples as well as minimizing the classification error:

$$\begin{aligned} \min_{\beta, w, b, \xi} \quad & \frac{1}{2} \sum_{m=1}^{M} \|\mathbf{w}_m\|^2 + C \sum_{i=1}^{N} \xi_i, \\ \text{s.t.} \quad & y_i \left(\sum_{m=1}^{M} \beta_m(x_i) \langle \mathbf{w}_m, \varphi_m(x_i) \rangle + b \right) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \forall i, \\ & \sum_{m=1}^{M} \beta_m(x) = 1, \quad \beta_m(x) \geq 0, \forall m, \end{aligned} \tag{7}$$

where $\|\mathbf{w}_m\|^2$ is a regularization term that is inversely related to the margin, $\sum_{i=1}^{N} \xi_i$ measures the total classification error, and C is a tuning parameter to seek a tradeoff between margin maximization and classification error minimization. The L_1 regulation on $\boldsymbol{\beta}$ is meant to promote the sparsity of sample-wise kernel weights.

4.2. The PS-MKL Dual Problem. By introducing Lagrange multipliers $\{\alpha_i\}_{i=1}^{N}$ into the inequality constraints in (7), we can formulate a Lagrangian dual function that satisfies the Karush-Kuhn-Tucker (KKT) conditions [30]. Correspondingly, the PS-MKL primal problem is finally formulated as a max-min problem:

$$\begin{aligned} \max_{\boldsymbol{\beta}} \min_{\boldsymbol{\alpha}} \quad & J \\ & J = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\sum_{m=1}^{M} \beta_m\left(x_i, x_j\right) K_m\left(x_i, x_j\right) \right) \\ & \quad - \sum_{i=1}^{N} \alpha_i, \\ \text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \forall i, \\ & \sum_{m=1}^{M} \beta_m\left(x_i, x_j\right) = 1, \quad \beta_m\left(x_i, x_j\right) \geq 0, \forall i, \forall j, \forall m. \end{aligned} \tag{8}$$

This max-min problem is subsequently called as a PS-MKL dual problem. Different from β_m in the canonical MKL dual problem [30], $\beta_m(x_i, x_j)$ in our PS-MKL dual problem is sample-wise. That is, the value of $\beta_m(x_i, x_j)$ not only depends on the kernel function $K_m(x_i, x_j)$, but also on the sample pair x_i and x_j.

$J(\cdot)$ is a multiobject function for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Fixing $\boldsymbol{\beta}$, to minimize $J(\cdot)$ over coefficients $\boldsymbol{\alpha}$ is to minimize the

global classification error and maximize the **inter-class** intervals. When $\boldsymbol{\alpha}$ is fixed, maximizing $J(\cdot)$ over sample-wise kernel weights $\boldsymbol{\beta}$ is meant to maximize the global **intra-class** similarity and minimize the **inter-class** similarity simultaneously. Solving this Max-Min problem is a typical saddle point problem. We next present the solution.

4.3. An Efficient Learning Algorithm. Similar to the parameter learning in a canonical MKL, we adopt a two-stage alternant optimization procedure.

4.3.1. The Computation of $\boldsymbol{\alpha}$ Given a Fixed $\boldsymbol{\beta}$. Fixing $\boldsymbol{\beta}$, the classifier coefficients $\boldsymbol{\alpha}$ can be estimated by minimizing J under the constraints of $0 \le \alpha_i \le C$ for all i and $\sum_{i=1}^{N} \alpha_i y_i = 0$. Minimization of J is identical to solving a standard SVM dual problem with the kernel combination

$$K\left(x_i, x_j\right) = \sum_{m=1}^{M} \beta_m\left(x_i, x_j\right) K_m\left(x_i, x_j\right). \tag{9}$$

So minimizing J over $\boldsymbol{\alpha}$ can be easily accomplished by existing efficient SVM solvers.

4.3.2. The Computation of $\boldsymbol{\beta}$ Given a Fixed $\boldsymbol{\alpha}$. Fixing $\boldsymbol{\alpha}$, optimizing the sample-wise kernel weights $\beta_m(x_i, x_j)$ is to explore the individual contribution of a feature (or a basic kernel) to the classification, with respect to the sample-pair x_i and x_j.

However, it is difficult to obtain an analytical solution about a generalized form of $\beta_m(x_i, x_j)$. For simplicity, we assume that $\beta_m(x_i, x_j)$ can be decoupled into $\beta_m(x_i)$ and $\beta_m(x_j)$, where $\beta_m(x_i)$ depends on sample x_i and the mth basic kernel only, and similarly for $\beta_m(x_j)$. Under this assumption, $\beta_m(x_i, x_j)$ can be expressed as $\beta_m(x_i)$. $\beta_m(x_j)$ as mentioned in [32]. However, this optimization function is not convex. Alternatively, we define $\beta_m(x_i, x_j) = (\beta_m^i + \beta_m^j)/2$ in our work to preserve the convexity of the optimization function J, where β_m^i corresponds to $\beta_m(x_i)$. Then the sample-wise kernel weights can be expressed by $\boldsymbol{\beta} = (\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^n, \ldots, \boldsymbol{\beta}^N)$ and $\boldsymbol{\beta}^n = (\beta_1^n, \ldots, \beta_m^n, \ldots, \beta_M^n)^T$, where $\beta_m^n \in \mathbb{R}$.

The learning process of our sample-wise kernel weights is different from $\beta_m(x_i, x_j)$ used in the local MKL [32]. Firstly, the objective function in [32] is not convex when solving the optimization problem of kernel combination, thereby leading to a local optimum. Secondly, the optimization of kernel combination in [32] resorts to an explicit feature representation, which would be subject to the curse of dimensionality.

To optimize the sample-wise kernel weights $\boldsymbol{\beta}$ with fixed $\boldsymbol{\alpha}$, the objective function in (8) can be expressed as

$$J(\boldsymbol{\beta}) = \sum_{i=1}^{N}\sum_{m=1}^{M} \beta_m^i S_m^i(\boldsymbol{\alpha}) + \sum_{j=1}^{N}\sum_{m=1}^{M} \beta_m^j S_m^j(\boldsymbol{\alpha}) - \sum_{i=1}^{N} \alpha_i, \tag{10}$$

where

$$\begin{aligned} S_m^i(\boldsymbol{\alpha}) &= \frac{1}{4}\alpha_i y_i \sum_{j=1}^{N} \alpha_j y_j K_m\left(x_i, x_j\right), \\ S_m^j(\boldsymbol{\alpha}) &= \frac{1}{4}\alpha_j y_j \sum_{i=1}^{N} \alpha_i y_i K_m\left(x_i, x_j\right). \end{aligned} \tag{11}$$

Without loss of generality, the objective function with respect to $\boldsymbol{\beta}$ can be rewritten as

$$\max_{\boldsymbol{\beta}} 2\sum_{i=1}^{N}\sum_{m=1}^{M} \beta_m^i S_m^i(\boldsymbol{\alpha}) - \sum_{i=1}^{N} \alpha_i, \tag{12}$$

then the optimization of J over $\boldsymbol{\beta}$ turns to

$$\begin{aligned} \max \quad & \theta \\ \text{w.r.t.} \quad & \theta \in \mathbb{R}, \quad \boldsymbol{\beta} \in \mathbb{R}^{N \cdot K}, \\ \text{s.t.2} \quad & \sum_{i=1}^{N}\sum_{m=1}^{M} \beta_m^i S_m^i(\boldsymbol{\alpha}) - \sum_{i=1}^{N} \alpha_i \ge \theta, \\ & \sum_{m=1}^{M} \beta_m^i = 1, \quad \beta_m^i \ge 0, \forall i, \forall m, \\ & \forall \boldsymbol{\alpha} \in \mathbb{R}^N, \quad \text{with } 0 \le \alpha_i \le C, \quad \sum_{i=1}^{N} \alpha_i y_i = 0. \end{aligned} \tag{13}$$

Given fixed $\boldsymbol{\alpha}$, the optimization of J over $\boldsymbol{\beta}$ is a linear program (LP) problem as θ and $\boldsymbol{\beta}$ are only linearly constrained. Nevertheless, the constraint on θ has to hold for every compatible $\boldsymbol{\alpha}$ resulting from infinite constraints. To solve this so-called semiinfinite linear program (SILP) problem, a column generation strategy is employed. That is, by solving the former QP with fixed $\boldsymbol{\beta}$ we produce a special $\boldsymbol{\alpha}$, which then adds a constraint on θ. In this way we add new constraints iteratively and consequently solve the LP with the help of all the gained constraints. This procedure has been proven to converge in [33].

In summary, we solve the saddle point problem by two alternant processes: (1) using an off-the-shelf SVM solver to learn the classifier coefficients $\boldsymbol{\alpha}$ and (2) using the simplex LPs to learn sample-wise kernel weights $\boldsymbol{\beta}$. In the training phase, sample-wise kernel weights of the training samples are leant by optimizing the object function $J(\boldsymbol{\beta})$. During the test phase, we simply adopt a flat distribution of kernel weights $\boldsymbol{\beta}$ for the nth testing sample. The basic reason why we fix the kernel weights of testing samples lies in the computational simplicity and the suppression on the potential impacts of testing samples over the learnt optimal sample-wise kernel combination. Our empirical results over diverse benchmarking datasets show such strategy has been effective for PS-MKL. It is worthy to note some promising learning strategies like local learning [21] may provide extensions to further optimize $\boldsymbol{\beta}$ during test phase.

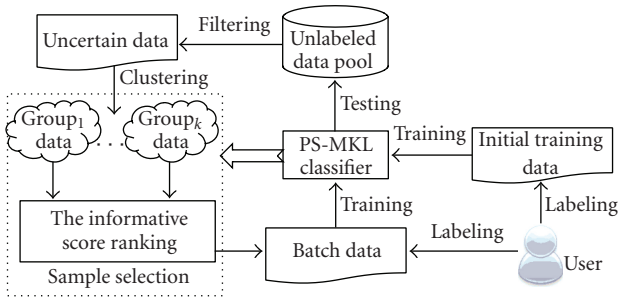

Figure 3: The flowchart of sample selection for PS-MKL.

4.3.3. Learning Algorithm for PS-MKL. The optimization algorithm of PS-MKL is summarized in Algorithm 1. The termination criteria can be the consistency of $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ between two consecutive steps, or a predefined iteration upper bound. Optimizing the classifier coefficients and the sample-wise kernel combination is a linear programming wrapping canonical SVM solver process.

5. Sample Selection for PS-MKL

In PS-MKL, the computational complexity involves two major processes: (1) computing multiple kernel functions for each sample-pair over the training set and (2) optimizing the classifier parameters and the sample-wise kernel weights in an alternate manner, which incurs intensive computation for an optimal solution. In particular, as the size of training set increases, the higher complexity of learning parameters would be a bottleneck in efficiently training PS-MKL. Intuitively, there would probably exist considerable data redundancy in a large-scale training dataset. By removing redundant data and keeping informative samples, we can reduce the size of actually used training data while the classifier's discriminative power is kept comparable.

In the community of machine learning, active learning is one of widely used techniques to reduce the labeling cost in supervised learning tasks. That is, by repeatedly querying the unlabeled samples, we may select the most informative samples to label, so that the demand for a large quantity of labeled data is alleviated [34]. In this paper, we incorporate the process of active learning into PS-MKL to select informative training samples for training. Accordingly, the calculation of kernel matrices, the learning of sample-wise kernel combination and a kernel-based classifier can be conducted over those selected samples only. So the computational complexity is reduced.

Figure 3 illustrates the basic process. Firstly, a preliminary PS-MKL classifier is trained on an initial labeled dataset. Secondly, the images from an unlabeled data pool will be filtered by the learnt classifier, collecting uncertain samples for further selection. Thirdly, these uncertain samples will go through sample selection to give a batch of most informative samples. Fourthly, a retraining of the classifier continues on the batch of data, together with the initial training data, to refine its discriminative power. Then the active learner jumps to the second step and continue.

5.1. Filtering in Uncertain Samples. Given an initial PS-MKL classifier learnt from a small labeled dataset L, our sample selection aims to collect unused samples that are *uncertain* for the classifier as the candidates of informative samples. These samples could be useful to further improve the classifier [35].

For each input, we can get an output score through computing the decision function of PS-MKL. This score indicates the likelihood that the input sample belongs to a given concept. Moreover, it ranks input samples by the likelihood of an instance belonging to a concept. In a sense, we may think the decision function transforms the original feature space into one-dimensional output score space, and establishes the ordered probabilities of a sample belonging to a concept. Hence, we select the *uncertain samples* according to the output score $f(x)$ of the PS-MKL:

$$\mathrm{Unc}(x) = \begin{cases} 1 & \text{if } T^- \le f(x) \le T^+, \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where $T^-$and $T^+$are the negative and positive bounds, respectively. These two bounds can be predefined or empirically estimated by a heuristic rule.

5.2. Selecting Informative Samples. To reduce the iterations of active learning, we adopt a batch mode to do sample selection. In each iteration, the uncertain samples are first clustered into groups, so that visually similar samples are merged into a group. Then the most informative samples in each group are added into the small labeled dataset L for classifier updating. K-means is utilized in our work, where other clustering methods can also be applied.

Within each group, we sort the candidate samples by three criteria: (1) the distance of a sample from the classification boundary, (2) the representativeness of a sample within the group, and (3) how distinguished a sample is from the training samples of L. The informative score $I(x)$ of a sample $x$can be computed as

$$\begin{aligned} I(x) = \rho \frac{1}{|f(x)|} + \lambda \sum_{x_k \in g} \frac{K(x, x_k)}{N_g} \\ + \ (1 - \rho - \lambda) \left(1 - \max_{x_k \in L} K(x, x_k)\right), \end{aligned} \tag{15}$$

where ρ and λ are two parameters to weight the importance of three components; $|f(x)|$ is the distance from the classification boundary; g denotes the group that sample x belongs to; N_g is the number of samples within the group g; $K(x, x_k)$ represents the similarity between sample x and x_k computed by the unweighted multikernel combination. Finally, the samples with the highest scores $I(x)$ within each group are selected for PS-MKL training.

Init $t = 1$, $\beta_m^i = 1/M$, $\forall i\ \forall m$

while the termination criterion NOT met **do**

(a) Compute the classifier coefficients $\boldsymbol{\alpha}^t$ by a standard SVM to solve the problem:

$$\boldsymbol{\alpha}^t = \arg\ \min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i$$

$$\text{s.t.}\sum_{i=1}^{N}\alpha_i y_i = 0, 0 \le \alpha_i \le C, \quad \forall i.$$

with $K(x_i, x_j) = \sum_{m=1}^{M}\beta_m(x_i, x_j)K_m(x_i, x_j)$.

(b) Update the object function

$$J^t = 2\sum_{i=1}^{N}\sum_{m=1}^{M}\beta_m^i S_m^i(\boldsymbol{\alpha}^t) - \sum_{i=1}^{N}\alpha_i^t,$$

where $S_m^i(\boldsymbol{\alpha}^t) = \frac{1}{4}\alpha_i^t y_i \sum_{j=1}^{N}\alpha_j^t y_j K_m(x_i, x_j)$.

(c) Compute the kernel weights $\boldsymbol{\beta}$ by

$$(\boldsymbol{\beta}, \theta) = \arg\max \theta$$

$$\text{w.r.t. } \boldsymbol{\beta} \in \mathbb{R}^{N\cdot K}, \quad \theta \in \mathbb{R}$$

$$\text{s.t.}\sum_{m=1}^{M}\beta_m^i = 1, \beta_m^i \ge 0, \quad \forall i, \forall m$$

$$2\sum_{i=1}^{N}\sum_{m=1}^{M}\beta_m^i S_m^i(\alpha^h) - \sum_{i=1}^{N}\alpha_i^h \ge \theta, \quad \text{for } h = 1, \cdots t.$$

(d) $t = t + 1$

end while

Compute the bias of the decision function (5) by

$$b = y_j - \sum_{i=1}^{N}\alpha_i^{t-1} y_i \sum_{m=1}^{M}\beta_m(x_i, x_j)K_m(x_i, x_j), \quad \forall j \in \{j \mid \alpha_j^{t-1} > 0\}.$$

Algorithm 1: PS-MKL optimization process.

6. Experiments

6.1. Datasets. Extensive experiments are performed on three well-known benchmarking datasets, that is, Caltech101 [15], WikipediaMM [16], and Pascal VOC'07 [17]. Caltech101 involves 102 object categories, each category containing 31 to 800 images. In WikipediaMM, some 150,000 images on diverse topics are crawled from Wikipedia. Our experiments select 33 topics, each of which has more than 80 positive samples. Pascal VOC'07 consists of 20 object categories from real-world images, where 2501 images are used for training, 2510 images for validation, and 4952 images for test. Our primary goal is to investigate the effectiveness of PS-MKL on open benchmarking data sets for fair comparisons with previous work. In addition, we consider the WikipediaMM from the multimedia community, which is to evaluate PS-MKL over a real-world data from a practical retrieval point of view, and how the performance of concept learning changes with the problem complexity. For each dataset, the *one-vs.-all* setting is adopted for training multiple classifiers of visual concepts.

6.2. Features and Kernels. Two local appearance features (dense-color-SIFT (DCSIFT) and dense-SIFT (DSIFT) [7]) and two shape features (self-similarity (SS) [36] and pyramid histogram of orientated gradients (PHOGs) [29]) are utilized. DCSIFT is computed in CIE-lab 3-channels over a square patch of radii r with the spacing of r. We use $r = 4$, 8, and 12 pixels to allow some scalability. In a similar manner, DSIFT is calculated in a gray image. SS descriptor captures the correlation map of a set of 5×5 patches with their neighbors at every 5th pixel. The correlation map is quantized into 10 orientations and 3 radial bins so as to form a 30 dim descriptor. Based on dense features, we employ k-means to quantize these three descriptors to obtain three codebooks of size k (set as 400, cf. [7]), respectively. For PHOG, two spatial pyramid kernels of gradient orientation are calculated to measure the image similarity in shape. We use the same PHOG setting as [29], that is, 20 orientation bins for PHOG-180 degree and 40 bins for PHOG-360 degree.

We implement SPK [7] and PDK [28]. For SPK, an image is divided into cells and a feature similarity from spatially corresponding cells between images is measured. The resulting kernel is formed as a weighted combination of histogram intersections from coarse cells to fine cells. A 4-level pyramid is used with the grid sizes of 8×8, 4×4, 2×2, and 1×1, respectively. For PDK, the local feature distributions of the K-nearest neighbors are compared between images. The resulting kernel combines the local feature distributions at multiple scales, for example, $K = 1, \ldots, k$, where k is set to (8, 16, 32) from the finest to the coarsest neighborhood. In total we employ eight kernels (different features or basic kernel functions) in our experiments.

Table 1: Mean recognition rates of four multiple kernel-based methods on Clatech101.

Methods	Number of positive samples for training					
	5	10	15	20	25	30
UMK	56.21	65.16	68.38	70.47	71.26	72.69
MKL	55.30	66.45	70.76	73.62	74.73	75.35
PS-MKL	58.59	69.24	74.82	77.37	79.41	*80.67*
PS-MKL-SS	58.59	70.02	76.63	*79.74*	80.91	81.82

Table 2: Comparison of four multiple kernel-based methods on WikiPediaMM by using Mean Average Precision as metric.

Methods	Number of positive samples for training					
	5	10	15	20	25	30
UMK	34.32	38.87	42.04	44.78	47.02	49.21
MKL	38.79	45.03	50.08	54.31	56.14	58.17
PS-MKL	40.63	47.14	53.67	57.53	59.82	*61.36*
PS-MKL-SS	40.63	51.35	57.54	*61.34*	62.23	64.84

6.3. Experiment Result

6.3.1. Experiments on Caltech101

Comparison with Different Multiple Kernel Methods. We compare the performance of four multiple kernel methods with the same setting of training/testing partition. Unweighted multiple kernel (UMK) method applies equal kernel weights to different basic kernels. Canonical MKL is implemented by the Matlab code from [30]. PS-MKL is implemented as introduced in Section 4. PS-MKL-SS employs the sample selection strategy in Section 5, while the other three methods apply a random sampling strategy. The mean recognition rates of four methods on Caltech101 are listed in Table 1. We adopt the same experimental setting as reported in state-of-the-art works [7, 8, 10, 14, 15, 18, 19, 21, 37–40], where different numbers of positive samples, that is, $N_{\text{train}} = \{5, 10, 15, 20, 25, 30\}$ are applied to the training phase and a fixed number of samples ($N_{\text{test}} = 30$) are applied to test phase for each concept.

Compared with UMK and MKL, PS-MKL achieves different improvements over all the training setting as shown in Table 1. We also observe that the sample selection further improves the performance of PS-MKL and obtains the best performance among four multiple kernel methods. Note that with a smaller training set $N_{\text{train}} = 20$, PS-MKL-SS has achieved promising MAPs comparable to that of PS-MKL at $N_{\text{train}} = 30$. Through comparisons, our proposed sample selection method is also shown to be more effective than a random sampling strategy in UMK, MKL, and PS-MKL.

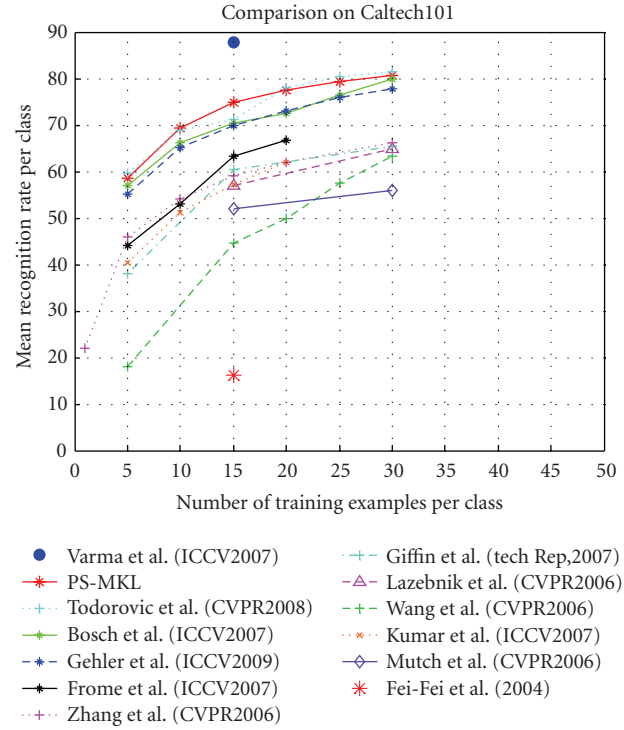

Figure 4: Performances of PS-MKL and other types of methods on Caltech101, with the PS-MKL figures of: 5(58.59), 10 (69.24), 15 (74.82), 20 (77.37), 25 (79.41), 30 (80.67) as (positive training sample number, mean recognition rate).

Comparison with Other Methods. On Caltech101, we compare PS-MKL with other types of recent methods [7, 8, 10, 14, 15, 18, 19, 21, 37–40]. For each concept, we randomly select N_{train} and N_{test} positive images to perform *one-vs.-all* training and testing at different training set sizes, where $N_{\text{train}} = \{5, 10, 15, 20, 25, 30\}$ and $N_{\text{test}} = 15$. As shown in Figure 4, our approach has achieved promising results comparable to the top performances. When $N_{\text{train}} = 30$, the mean recognition rate of PS-MKL reaches up to 80.67%.

As a multiple kernel-based method, the approach of Varma and Ray [8] got the best result on Caltech 101, with a recognition rate of 87.82% when $N_{\text{train}} = 15$. Both PS-MKL and MKL did not catch up with [8]. One of possible reasons is different features. In [8], six shape and appearance features (i.e., geometric blur (GB), GB distance [37], DSIFT,

Table 3: Average Precision of PS-MKL and other methods on the Pascal VOC 2007 dataset.

Categories	[41]	[24]	[42]	[43]	UMK	MKL	PS-MKL
Aero plane	77.5	63.0	65.0	65.0	72.9	74.1	75.0
Bicycle	63.6	22.0	44.3	48.0	52.2	53.9	54.0
Bird	56.1	14.0	48.6	44.0	47.0	46.6	48.6
Boat	71.9	42.0	58.4	60.0	61.8	62.2	65.2
Bottle	33.1	43.0	17.8	20.0	33.5	37.5	41.6
Bus	60.6	50.0	46.4	49.0	49.0	55.6	57.7
Car	78.0	62.0	63.2	70.0	69.5	70.7	71.9
Cat	58.8	32.0	46.8	49.0	45.8	48.4	48.4
Chair	53.5	37.0	42.2	50.0	53.2	54.0	59.0
Cow	42.6	19.0	29.6	32.0	32.7	34.7	36.8
Dining table	54.9	30.0	20.8	39.0	48.3	50.1	52.1
Dog	45.8	29.0	37.7	40.0	44.2	40.7	46.7
Horse	77.5	15.0	66.6	72.0	70.3	76.6	72.6
Motor-bike	64.0	31.0	50.3	59.0	55.0	59.8	63.6
Person	85.9	43.0	78.1	81.0	85.1	82.5	86.5
Potted plant	36.3	33.0	27.2	32.0	32.4	38.3	44.3
Sheep	44.7	41.0	32.1	35.0	36.8	40.0	43.0
Sofa	50.6	37.0	26.8	42.0	45.9	48.2	54.2
Train	79.2	29.0	62.8	68.0	67.5	68.1	70.1
TV monitor	53.2	62.0	33.3	49.0	47.5	47.2	49.2
Mean AP	59.4	36.7	44.9	50.2	52.5	54.5	57.0

DCSIFT, PHOG180, and PHOG 360) are incorporated into an extended MKL framework.

6.3.2. Experiments on WikiPediaMM. Now let us evaluate the multiple kernel-based methods (i.e., UMK, MKL, PS-MKL, and PS-MKL-SS) on WikipediaMM. For each visual concept, we perform six runs with $N_{\text{train}} = \{5, 10, 15, 20, 25, 30\}$ positive image samples for *one-vs.-all* training, while the rest of positive images are used for test. The mean of average precisions (MAPs) [16] for six runs are listed in Table 2. At the same empirical setting, PS-MKL outperforms UMK and MKL. That is, even on the large-scale image corpus in practice, PS-MKL shows stronger discriminative power.

From Table 2, we observe that PS-MKL-SS achieves improvements over PS-MKL with more sampling images in each round. At a smaller training size $N_{\text{train}} = 20$, PS-MKL-SS has achieved an MAP 61.34%, which is comparable to the best result of PS-MKL (MAP 61.36% at $N_{\text{train}} = 30$). Beyond PS-MKL, PS-MKL-SS not only achieves better results, but also is able to get comparable performance with less training samples and comparatively lower computation. So our proposed PS-MKL equipped with a sample selection is more effective and efficient.

To illustrate the effects of PS-MKL on WikipediaMM, we list the top correctly returned positive and negative images of ten concepts in Figure 5. With a finer category, some negative and positive images of a concept not only have similar appearances, but also produce semantic correlations, for example, hunting dog versus pet dog, and race car versus vehicle, whereas our PS-MKL works well.

6.3.3. Experiments on Pascal VOC'07. For each concept, we employ 5011 images for training and 4952 images for test. Table 3 compares the performances of PS-MKL to two multiple kernel methods (UMK and MKL) and other recent methods [24, 41–43]. An official performance metric Average Precision (AP) [17] is used.

At the same setting of training/testing sets, PS-MKL has obtained relative improvements of 7.9% and 4.4% over UMK and MKL. More or less improvements have been achieved over all 20 concepts. In particular, "chair" and "potted plant" receive over 10% improvements. By investigating the intra-class diversity of each concept in Pascal VOC'07, we find that PS-MKL allows better discriminative power than UMK and MKL in the dataset.

From Table 3, PS-MKL has achieved a promising MAP 57.0%, which is better than [24, 42, 43], and the performance is very close to the best reported performance (MAP 59.4% in [41]) in Pascal VOC'07 challenge.

6.4. Efficiency. Computing kernel matrix is time-consuming due to the intense similarity calculation based on diverse kernel functions. However, these kernel matrices can be pre-computed and loaded at the learning stage. We implemented PS-MKL in C++ on PC (3.0 GHz core, and 2 G RAM). In each loop of training, we need to solve a canonical SVM problem with the sample-wise kernel weights optimized by

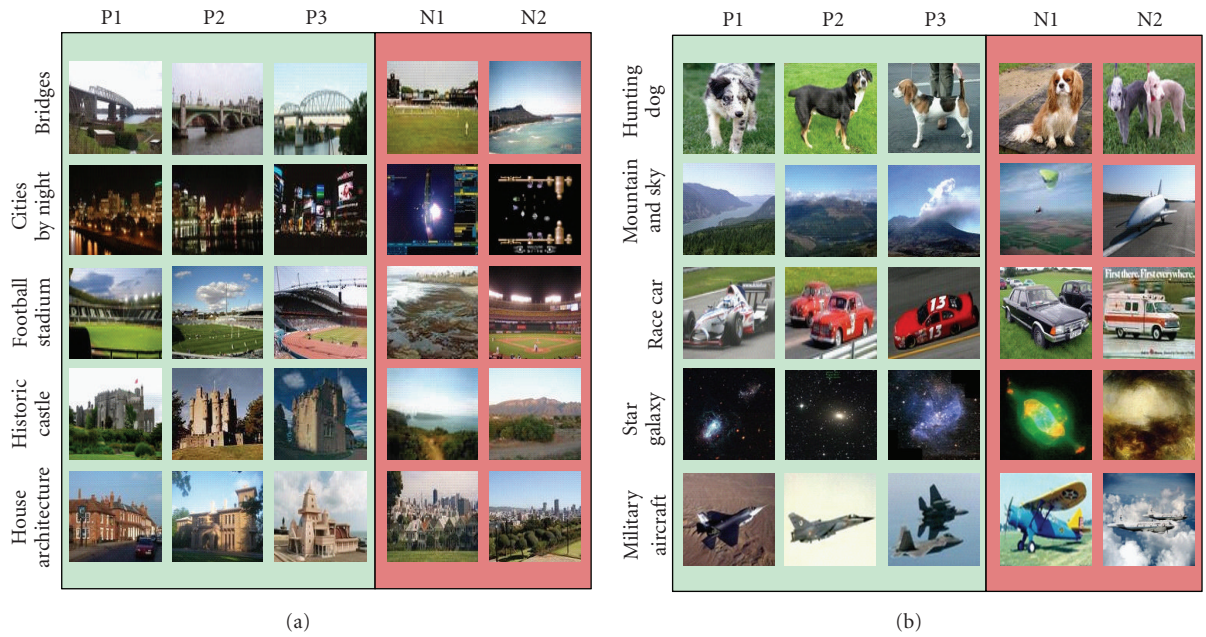

Figure 5: Illustration of top three positive and top two negative samples of correctly returned results on WikipediaMM dataset.

SILP. The time complexity of SILP is ignorable compared to the SVM solver. Using hot start (i.e., providing previous $\boldsymbol{\alpha}$ as an initial input) can accelerate the process of SVM solvers. In the case of learning a concept from 5 k samples on Pascal VOC'07, training a canonical MKL takes some 12 minutes, and PS-MKL needs about 30 minutes to converge for each concept. To compute the decision function during test, we need to calculate the kernel function $K_m(x_i, x)$ only when α_i and β_m^i are nonzero.

7. Discussion

Through working across multiple basic kernel spaces, PS-MKL helps to capture the intra-class diversity of a visual concept. When a canonical MKL method is deficient in dealing with the intra-class diversity using uniform kernel combination, PS-MKL may provide a tractable approach to kernel weighting at the sample level. Our experiments show that PS-MKL achieves significant improvements over a canonical MKL method.

Like other sample-based methods, PS-MKL has to optimize numerous parameters especially when massive training samples are available. For example, in Caltech101, 8 basic kernels and 30 training samples per class lead to some 25 k sample-wise kernel weights. Since the number of classifier parameters is in proportion to the number of (sparse) support vectors, there are only 1~3 k nonzero sample-wise kernel weights to optimize, and then PS-MKL is tractable. In practice, however, when the number of training samples keeps growing, how to solve PS-MKL efficiently would become a critical problem.

In another work [44], we extend PS-MKL and present a group-sensitive multiple kernel learning method (GS-MKL). In GS-MKL, an intermediate representation "group" is introduced between object categories and individual images to seek a tradeoff between capturing the diversity and keeping the invariance for each category. Visually similar training samples within a group are assigned a uniform kernel combination, while distinct training samples may have different kernel combinations. Hence, the number of kernel weights can be greatly reduced by allowing visually similar training samples to share a kernel combination setting.

How to make the optimal allocation of different kernel combinations over more complex training samples is included in our future work to take into account more practical issues such as a discriminative power and a reduced classifier complexity.

8. Conclusion

We have proposed a novel per-sample multiple kernels learning method for visual concept learning. Different from a canonical MKL method, PS-MKL is able to capture the contributions of different basic kernels over individual training samples by a sample-wise kernel combination, rather than the uniform weighting over the whole input space like a canonical MKL. Our proposed PS-MKL approach optimizes the sample-wise kernel weights and the kernel-based classifier in a joint manner, where the optimal parameters can be learned alternately with off-the-shelf SVM solvers and simplex LPs. Moreover, we present an effective sample selection method for PS-MKL to alleviate the computational load in the training process. Extensive experiments show that PS-MKL has achieved encouraging results over three well-known benchmarking datasets.

Acknowledgments

The work is supported by grants from the Chinese National Natural Science Foundation under Contract no. 60605020, no. 60973055, no. 60902057, and no. 90820003, The National Hi-Tech R and D Program (863) of China under Contract 2006AA010105, and the National Basic Research Program of China under Contract no. 2009CB320906. This work was performed when Jingjing Yang and Yuanning Li were visiting NELVT as full-time research assistant.

References

[1] M. Szummer and R. W. Picard, "Indoor-outdoor image classification," in *Proceedings of the International Workshop on Content-Based Access of Image and Video Databases*, 1998.

[2] J. Vogel and B. Schiele, "Natural scene retrieval based on a semantic modeling step," in *Proceedings of the International Conference on Image and Video Retrieval*, pp. 207–215, 2004.

[3] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their location in images," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 1, pp. 370–377, Beijing, China, October 2005.

[4] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from Google's image search," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, vol. 2, pp. 1816–1823, Beijing, China, October 2005.

[5] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[6] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in Neural Information Processing Systems*, 2005.

[7] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 2, pp. 2169–2178, New York, NY, USA, June 2006.

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

[9] 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.

[10] A. Kumar and C. Sminchisescu, "Support Kernel machines for object recognition," in *Proceedings of the 11th IEEE International Conference on Computer Vision*, 2007.

[11] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Learning*, pp. 185–208, MIT Press, Cambridge, Mass, USA, 1998.

[12] O. Chum and A. Zisserman, "An exemplar model for learning object classes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2007.

[13] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning globally-consistent local distance functions for shape-based image retrieval and classification," in *Proceedings of the 11th IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.

[14] T. Malisiewicz and A. A. Efros, "Recognition by association via learning per-exemplar distances," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, June 2008.

[15] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories," in *Proceedings of the Computer Vision and Pattern Recognition (CVPR '04)*, 2004.

[16] http://www.imageclef.org/2008/wikipedia.

[17] M. Everingham, L. VanGool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge 2007 (VOC2007) results," http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/workshop/index.html.

[18] S. Fidler, M. Boben, and A. Leonardis, "Similarity-based cross-layered hierarchical representation for object categorization," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, June 2008.

[19] A. Bosch, A. Zisserman, and X. Muñoz, "Image classification using random forests and ferns," in *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV '07)*, October 2007.

[20] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao, "A new multiple kernel approach for visual concept learning," in *Proceedings of the 15th International Multimedia Modeling Conference*, pp. 250–262, 2009.

[21] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, "Local ensemble kernel learning for object category recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[22] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 10–17, June 2005.

[23] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[24] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, June 2008.

[25] A. Ng and M. Jordan, "On discriminative vs. generative classifiers: a comparison of logistic regression and naive Bayes," in *Advances in Neural Information Processing Systems*, 2002.

[26] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, Mass, USA, 2004.

[27] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *Proceedings of the 10th IEEE International Conference on Computer Vision*, vol. 2, pp. 1458–1465, Beijing, China, October 2005.

[28] H. Ling and S. Soatto, "Proximity distribution kernels for geometric context in category recognition," in *Proceedings of the 11th IEEE International Conference on Computer Vision*, 2007.

[29] 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*, pp. 401–408, 2007.

[30] S. Sonnenburg, G. Raetsch, C. Schaefer, and B. Scholkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

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

[32] M. Gönen and E. Alpaydin, "Localized multiple kernel learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 352–359, July 2008.

[33] S. Sonnenburg, G. Ratsch, and C. Schafer, "A general and efficient multiple kernel learning algorithm," in *Neural Information Processing Systems*, 2005.

[34] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[35] S. C. H. Hoi, J. Zhu, R. Jin, and M. R. Lyu, "Semi-supervised SVM batch mode active learning for image retrieval," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[36] 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*, 2007.

[37] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: discriminative nearest neighbor classification for visual category recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2126–2136, June 2006.

[38] G. Wang, Y. Zhang, and L. Fei-Fei, "Using dependent regions for object categorization in a generative framework," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1597–1604, 2006.

[39] J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 11–18, 2006.

[40] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[41] M. Marszałek, C. Schmid, H. Harzallah, and J. Weijer, "Learning object representations for visual object class recognition," in *Proceedings of the Visual Recognition Challenge Workshop, in Conjunction with IEEE International Conference on Computer Vision*, 2007.

[42] G. Wang, D. Hoiem, and D. Forsyth, "Learning image similarity from Flickr groups using stochastic intersection kernel machines," in *Proceedings of the of International Workshop on Multimedia Information Retrieval*, 2008.

[43] F. Khan, J. Weijer, and M. Vanrell, "Top-down color attention for object recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[44] J. J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao, "Group-sensitive multi-kernel learning for object categorization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009.