A Convex Method for Locating Regions of Interest with Multi-instance Learning

Yu-Feng Li[1], James T. Kwok[2], Ivor W. Tsang[3], and Zhi-Hua Zhou[1]

[1] National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
{liyf,zhouzh}@lamda.nju.edu.cn
[2] Department of Computer Science and Engineering
Hong Kong University of Science and Technology, Hong Kong, China
jamesk@cse.ust.hk
[3] School of Computer Engineering
Nanyang Technological University, Singapore 639798
IvorTsang@ntu.edu.sg

Abstract. In content-based image retrieval (CBIR) and image screening, it is often desirable to locate the regions of interest (ROI) in the images automatically. This can be accomplished with multi-instance learning techniques by treating each image as a bag of instances (regions). Many SVM-based methods are successful in predicting the bag labels, however, few of them can locate the ROIs. Moreover, they are often based on either local search or an EM-style strategy, and may get stuck in local minima easily. In this paper, we propose two convex optimization methods which maximize the margin of concepts via key instance generation at the instance-level and bag-level, respectively. Our formulation can be solved efficiently with a cutting plane algorithm. Experiments show that the proposed methods can effectively locate ROIs, and they also achieve performances competitive with state-of-the-art algorithms on benchmark data sets.

1 Introduction

With the rapid expansion of digital image collections, content-based image retrieval (CBIR) has attracted more and more interest. The main difficulty of CBIR lies in the gap between the high-level image semantics and the low-level image features. Much endeavor has been devoted to bridging this gap, it remains unsolved yet. Generally, the user first poses in the query and relevance feedback process several labeled images that are relevant/irrelevant to an underlying target concept. Then the CBIR system attempts to retrieve all images from the database that are relevant to the concept. It is noteworthy that although the user feeds whole images to the system, usually s/he is only interested in some regions, i.e., *regions of interest* (ROIs), in the images.

For medical and military applications which require a fast scanning of huge amount of images to detect suspect areas, it is very desirable if ROIs can be identified and exhibited when suspected images are presented to the examiner. Even in common CBIR scenarios, considering that the system usually returns a lot of images, the explicit identification of ROIs may help the user in recognizing images s/he really wants more quickly.

W. Buntine et al. (Eds.): ECML PKDD 2009, Part II, LNAI 5782, pp. 15–30, 2009.

In multi-instance learning [6], the training examples are *bags* each containing many instances. A bag is positively labeled if it contains at least one positive instance, and negatively labeled otherwise. The task is to learn a model from the training bags for correctly labeling unseen bags. Multi-instance learning is difficult because that, unlike conventional supervised learning tasks where all the training instances are labeled, here the labels of the individual instances are unknown. It is obvious that if a whole image is regarded as a bag with its regions being regarded as instances, the problem of determining whether an image is relevant to a target concept or not can be viewed as a multi-instance problem. So, it is not surprising that multi-instance learning has been found very useful in tasks involving image analysis.

In general, three kinds of multi-instance learning approaches can be used to locate the ROIs. The first is the Diverse Density (DD) algorithm [15] and its variants, e.g., EM-DD [26] and multi-instance logistic regression [19]. These methods apply gradient search with multiple restarts to identify an instance which maximizes the *diverse density*, that is, an instance close to every positive bags while far from negative bags. The instance is then regarded as the prototype of the target concept. It is obvious that DD can be applied to locate ROIs. A serious problem with this kind of methods is the huge time cost, since they have to perform gradient search starting from every instance in every positive bag.

The second approach is the C*k*NN-ROI algorithm [29], which is a variant of Citation-*k*NN [23]. This approach uses Citation-*k*NN to predict whether a bag is positive or not. It takes the minimum distance between the nearest pair of instances from two bags as the distance between bags, and then utilizes *citers* of the neighbors to improve performance. Subsequently, each instance in a positive bag is regarded as a bag and a score is calculated by considering its distance to other bags, from which the key instance can be decided. The time complexity of C*k*NN-ROI is mainly dominated by the calculation of neighbors, and is much more efficient than DD. However, this algorithm is based on heuristics and the theoretical justification has not been established yet.

The third approach is MI-SVM [1]. While many SVM-based multi-instance learning methods have been developed [1,3,4], to the best of our knowledge, MI-SVM is the only one that can locate the ROIs. The MI-SVM locates ROI (also referred to as the key instance) with an EM-style procedure. It first starts with a SVM using some multi-instance kernel [8] and picks the key instances according to the SVM prediction, and the SVM is then retrained with respect to the key instance assignment; the procedure is repeated until convergence. Empirical study shows that MI-SVM is efficient and works well on many multi-instance data sets. In fact, MI-SVM can be viewed as a *constrained concave-convex programming* (CCCP) method whose convergence has been well-studied [5]. Each MI-SVM iteration only involves the solving of a convex optimization problem, however, the optimization problem as a whole is still non-convex and suffers from local minima.

In this paper, we focus on SVM-based methods and propose the KI-SVM (key-instance support vector machine) algorithm. We formulate the problem as a convex optimization problem. At each iteration, KI-SVM generates a violated key instance assignment and then combines them via efficient multiple kernel learning. It is noteworthy that it involves a series of standard SVM subproblems that can be solved with various

state-of-the-art SVM implementations in a scalable and efficient manner, such as SVM-*perf* [10], LIBSVM [7], LIBLINEAR [9] and CVM [21]. Two variants of the KI-SVM, namely, Ins-KI-SVM and Bag-KI-SVM, are proposed for locating the key instances at the instance-level and bag-level, respectively.

The rest of the paper is organized as follows. Section 2 briefly introduces MI-SVM. Section 3 proposes our KI-SVM method. Experimental results are reported in Section 4. The last section concludes the paper.

2 Multi-instance Support Vector Machines

In the sequel, we denote the transpose of a vector/matrix (in both the input and feature spaces) by the superscript $'$. The zero vector and the vector of all ones are denoted as $\mathbf{0}, \mathbf{1} \in \mathbb{R}^n$, respectively. Moreover, the inequality $\mathbf{v} = [v_1, \cdots, v_k]' \geq \mathbf{0}$ means that $v_i \geq 0$ for $i = 1, \cdots, k$.

In multi-instance classification, we are given a set of training bags $\{(B_1, y_1), \cdots, (B_m, y_m)\}$, where $B_i = \{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \cdots, \mathbf{x}_{i,m_i}\}$ is the ith bag containing instances $\mathbf{x}_{i,j}$'s, m_i is the size of bag B_i, and $y_i \in \{\pm 1\}$ is its bag label. Suppose the decision function is denoted as $f(\mathbf{x})$. As is common in the traditional MI setting, we take $f(B_i) = \max_{1 \leq j \leq m_i} f(\mathbf{x}_{i,j})$. Furthermore, $\mathbf{x}_{i,l} = \arg\max_{\mathbf{x}_{i,j}} f(\mathbf{x}_{i,j})$ is viewed as the key instance of a positive bag B_i. For simplification, we assume that the decision function is a linear model, i.e., $f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x})$, where ϕ is the feature map induced by some kernel k.

The goal is to find f that minimizes the structural risk functional

$$\Omega(\|\mathbf{w}\|_p) + C \sum\nolimits_{i=1}^{m} \ell\left(-y_i \max_{1 \leq j \leq m_i} \mathbf{w}'\phi(\mathbf{x}_{i,j})\right), \tag{1}$$

where Ω can be any strictly monotonically increasing function, $\ell(\cdot)$ is a monotonically increasing loss function, and C is a regularization parameter that balances the empirical risk functional and the model complexity. In this paper, we focus on $\Omega(\|\mathbf{w}\|_p) = \frac{1}{2}||\mathbf{w}||^2$ and the squared hinge loss. So, (1) becomes:

$$\min_{\mathbf{w}, \rho, \boldsymbol{\xi}} \ \frac{1}{2}||\mathbf{w}||_2^2 - \rho + \frac{C}{2}\sum_{i=1}^{m} \xi_i^2 \tag{2}$$

$$\text{s.t } \ y_i \max_{1 \leq j \leq m_i} \mathbf{w}'\phi(\mathbf{x}_{i,j}) \geq \rho - \xi_i, \ i = 1 \cdots, m, \tag{3}$$

where $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_m]'$. This, however, is a non-convex problem because of the max operator for positive bags.

Andrews *et al.* [1] proposed two heuristic extensions of the support vector machines, namely, the mi-SVM and MI-SVM, for this multi-instance learning problem. The mi-SVM treats the MI learning problem in a supervised learning manner, while the MI-SVM focuses on finding the key instance in each bag. Later, Cheung and Kwok [5] proposed the use of the *constrained concave-convex programming* (CCCP) method, which has well-studied convergence properties, for this optimization problem. However, while each iteration only involves the solving of a convex optimization problem,

the optimization problem as a whole is non-convex and so still suffers from the problem of local minima.

3 KI-SVM

In this section, we propose two versions of KI-SVM, namely the Ins-KI-SVM (instance-level KI-SVM) and Bag-KI-SVM (bag-level KI-SVM).

3.1 Mathematical Formulation

Let p be the number of positive bags. Without loss of generality, we assume that the positive bags are ordered before negative bags, i.e., $y_i = 1$ for all $1 \leq i \leq p$ and -1 otherwise. Moreover, let $J_i = \sum_{t=1}^{i} m_t$.

For a positive bag B_i, we use a binary vector $\mathbf{d}_i = [d_{i,1}, \cdots, d_{i,m_i}]' \in \{0,1\}^{m_i}$ to indicate which instance in B_i is its key instance. Here, followed the traditional multi-instance setup, we assume that each positive bag has only one key instance, so $\sum_{j=1}^{m_i} d_{i,j} = 1$[1]. In the following, let $\mathbf{d} = [\mathbf{d}_1, \cdots, \mathbf{d}_p]$, and Δ be its domain. Moreover, note that $\max_{1 \leq j \leq m_i} \mathbf{w}'\phi(\mathbf{x}_{i,j})$ in (3) can be written as $\max_{\mathbf{d}_i} \sum_{j=1}^{m_i} d_{i,j}\mathbf{w}'\phi(\mathbf{x}_{i,j})$ in this case.

For a negative bag B_i, all its instances are negative and the corresponding constraint (3) can be replaced by $-\mathbf{w}'\phi(\mathbf{x}_{i,j}) \geq \rho - \xi_i$ for every instance in B_i. Moreover, we relax the problem by allowing the slack variable ξ_i to be different for different instances of bag B_i. This leads to a set of slack variables $\{\xi_{s(i,j)}\}_{i=1,\cdots,m;j=1,\cdots,m_i}$, where $s(i,j) = J_{i-1} - J_p + j + p$ is the indexing function that numbers these slack variables from $p+1$ to $N = J_m - J_p + p$.

Combining all these together, (2) can be rewritten as:

$$\begin{aligned}
\text{(Ins-KI-SVM)} \quad \min_{\mathbf{w},\rho,\boldsymbol{\xi},\mathbf{d}} \quad & \frac{1}{2}||\mathbf{w}||_2^2 - \rho + \frac{C}{2}\sum_{i=1}^{p}\xi_i^2 + \frac{\lambda C}{2}\sum_{i=p+1}^{m}\sum_{j=1}^{m_i}\xi_{s(i,j)}^2 \\
\text{s.t.} \quad & \sum_{j=1}^{m_i} \mathbf{w}' d_{i,j}\phi(\mathbf{x}_{i,j}) \geq \rho - \xi_i, \quad i = 1, \cdots, p, \\
& -\mathbf{w}'\phi(\mathbf{x}_{i,j}) \geq \rho - \xi_{s(i,j)}, \quad i = p+1, \cdots, m, \\
& \qquad\qquad\qquad\qquad\qquad\qquad j = 1, \cdots, m_i, \qquad (4)
\end{aligned}$$

where λ balances the slack variables from the positive and negative bags.

Note that each instance in a negative bag leads to a constraint in (4). Potentially, this may result in a large number of constraints in optimization. Here, we consider another variant that simply represents each negative bag in the constraint by the mean of its instances. It has been shown that this representation is reasonable and effective in many cases [8,24]. Thus, we have the following optimization problem:

[1] In many cases the standard assumption of multi-instance learning, that is, the positive label is triggered by a key instance, does not hold. Instead, the positive label may be triggered by more than one key instances [22,24,30]. Suppose the number of key instances is v, we can simply set $\sum_{j=1}^{m_i} d_{i,j} = v$, and thus our proposal can also handle this situation with a known v.

$$\text{(Bag-KI-SVM)} \min_{\mathbf{w},\rho,\boldsymbol{\xi},\mathbf{d}} \frac{1}{2}||\mathbf{w}||_2^2 - \rho + \frac{C}{2}\sum_{i=1}^{p}\xi_i^2 + \frac{\lambda C}{2}\sum_{i=p+1}^{m}\xi_i^2$$
$$\text{s.t.} \quad \sum_{j=1}^{m_i}\mathbf{w}' d_{i,j}\phi(\mathbf{x}_{i,j}) \geq \rho - \xi_i, \quad i = 1, \cdots, p,$$
$$-\mathbf{w}'\frac{\sum_{j=1}^{m_i}\phi(\mathbf{x}_{i,j})}{m_i} \geq \rho - \xi_i, \quad i = p+1, \cdots, m. \tag{5}$$

Hence, instead of a total of $\sum_{i=p+1}^{m} m_i$ constraints for the negative bags in (4), there are now only $m-p$ corresponding constraints in (5). As (4) considers each *instance* (in a negative bag) as one constraint, while (5) only represents the whole negative *bag* as a constraint. Therefore, we will refer to the formulations in (4) and (5) as the instance-level KI-SVM (Ins-KI-SVM) and bag-level KI-SVM (Bag-KI-SVM), respectively.

As (4) and (5) are similar in form, we consider in the following a more general optimization problem for easier exposition:

$$\min_{\mathbf{w},\rho,\boldsymbol{\xi},\mathbf{d}} \frac{1}{2}||\mathbf{w}||_2^2 - \rho + \frac{C}{2}\sum_{i=1}^{p}\xi_i^2 + \frac{\lambda C}{2}\sum_{i=p+1}^{r}\xi_i^2$$
$$\text{s.t.} \quad \sum_{j=1}^{m_i}\mathbf{w}' d_{i,j}\phi(\mathbf{x}_{i,j}) \geq \rho - \xi_i, \quad i = 1, \cdots, p,$$
$$-\mathbf{w}'\psi(\hat{\mathbf{x}}_i) \geq \rho - \xi_i, \quad i = p+1, \cdots, r. \tag{6}$$

It is easy to see that both the Ins-KI-SVM and Bag-KI-SVM are special cases of (6). Specifically, when $r = N$, and $\psi(\hat{\mathbf{x}}_{s(i,j)}) = \phi(\mathbf{x}_{i,j})$ for the second constraint, (6) reduces to the Ins-KI-SVM. Alternatively, when $r = m$, and $\psi(\hat{\mathbf{x}}_i) = \frac{\sum_{j=1}^{m_i}\phi(\mathbf{x}_{i,j})}{m_i}$ for the second constraint, then (6) becomes the Bag-KI-SVM.

By using the method of Lagrange multipliers, the Lagrangian can be obtained as:

$$\begin{aligned}
&\mathcal{L}(\mathbf{w}, \rho, \boldsymbol{\xi}, \mathbf{d}, \boldsymbol{\alpha}) \\
&= \frac{1}{2}||\mathbf{w}||_2^2 - \rho + \frac{C}{2}\sum_{i=1}^{p}\xi_i^2 + \frac{\lambda C}{2}\sum_{i=p+1}^{r}\xi_i^2 - \sum_{i=1}^{p}\alpha_i(\sum_{j=1}^{m_i}\mathbf{w}' d_{i,j}\phi(\mathbf{x}_{i,j}) - \rho + \xi_i) \\
&\quad - \sum_{i=p+1}^{r}\alpha_i(-\mathbf{w}'\psi(\hat{\mathbf{x}}_i) - \rho + \xi_i).
\end{aligned}$$

By setting the partial derivatives with respect to the $\mathbf{w}$, ρ, $\boldsymbol{\xi}$ to zeros, we have

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{p}\alpha_i\sum_{j=1}^{m_i} d_{i,j}\phi(\mathbf{x}_{i,j}) + \sum_{i=p+1}^{r}\alpha_i\psi(\hat{\mathbf{x}}_i) = 0,$$
$$\frac{\partial L}{\partial \rho} = -1 + \sum_{i=1}^{r}\alpha_i = 0,$$

$$\frac{\partial L}{\partial \xi_i} = C\xi_i - \alpha_i = 0, \forall i = 1, \cdots, p,$$
$$\frac{\partial L}{\partial \xi_i} = \lambda C\xi_i - \alpha_i = 0, \forall i = p+1, \cdots, r.$$

Then, the dual of (6) can be obtained as

$$\min_{\mathbf{d}\in\Delta} \max_{\boldsymbol{\alpha}\in\mathcal{A}} -\tfrac{1}{2}(\boldsymbol{\alpha} \odot \hat{\mathbf{y}})' \Big(\mathbf{K}^{\mathbf{d}} + \mathbf{E}\Big)(\boldsymbol{\alpha} \odot \hat{\mathbf{y}}), \tag{7}$$

where $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_r]' \in \mathbb{R}^r$ is the vector of Lagrange multipliers, $\mathcal{A} = \{\boldsymbol{\alpha} \mid \sum_{i=1}^r \alpha_i = 1, \alpha_i \geq 0\}$, $\hat{\mathbf{y}} = [\mathbf{1}_p, -\mathbf{1}_{r-p}] \in \mathbb{R}^r$, $\odot$ denotes the element-wise product of two matrices, $\mathbf{E} \in \mathbb{R}^{r\times r}$ is a diagonal matrix with diagonal entries

$$E_{i,i} = \begin{cases} \frac{1}{C} & i = 1, \cdots, p, \\ \frac{1}{\lambda C} & \text{otherwise,} \end{cases}$$

and $\mathbf{K}^{\mathbf{d}} \in \mathbb{R}^{r\times r}$ is the kernel matrix where $\mathbf{K}^{\mathbf{d}}_{ij} = (\boldsymbol{\psi}^{\mathbf{d}}_i)'(\boldsymbol{\psi}^{\mathbf{d}}_j)$ with

$$\boldsymbol{\psi}^{\mathbf{d}}_i = \begin{cases} \sum_{j=1}^{m_i} d_{i,j}\phi(\mathbf{x}_{i,j})' & i = 1, \cdots, p, \\ \psi(\hat{\mathbf{x}}_i) & i = p+1, \cdots, r. \end{cases} \tag{8}$$

Note that (7) is a mixed-integer programming problem, and so is computationally intractable in general.

3.2 Convex Relaxation

The main difficulty of (7) lies in the variables $\mathbf{d}$ which is hard to optimize in general. But once the $\mathbf{d}$ is given, the inner problem of (7) will become a standard SVM which could be solved in an efficient manner. This simple observation motivates us to avoid optimizing $\mathbf{d}$, alternatively, to learn the optimal combination of some $\mathbf{d}$'s. Further observed that each $\mathbf{d}$ corresponds to a kernel $\mathbf{K}^{\mathbf{d}}$, learning the optimal convex combination will become multiple kernel learning (MKL) [13] which is convex and efficient in general.

In detail, we consider a minimax relaxation [14] by exchanging the order of $\min_{\mathbf{d}}$ and $\max_{\boldsymbol{\alpha}}$. According to the minimax inequality [12], (7) can be lower-bounded by

$$\begin{aligned} &\max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\mathbf{d}\in\Delta} \; -\frac{1}{2}(\boldsymbol{\alpha} \odot \hat{\mathbf{y}})' \Big(\mathbf{K}^{\mathbf{d}} + \mathbf{E}\Big)(\boldsymbol{\alpha} \odot \hat{\mathbf{y}}) \\ &= \max_{\boldsymbol{\alpha}\in\mathcal{A}} \Big\{ \max_{\theta} -\theta \\ &\qquad \text{s.t.} \quad \theta \geq \frac{1}{2}(\boldsymbol{\alpha} \odot \hat{\mathbf{y}})' \Big(\mathbf{K}^{\mathbf{d}_t} + \mathbf{E}\Big)(\boldsymbol{\alpha} \odot \hat{\mathbf{y}}), \; \forall \mathbf{d}_t \in \Delta \Big\}. \end{aligned} \tag{9}$$

By introducing the dual variable $\mu_t \geq 0$ for each constraint, then its Lagrangian is

$$-\theta + \sum_{t:\mathbf{d}_t\in\Delta} \mu_t \Big(\theta - \frac{1}{2}(\boldsymbol{\alpha} \odot \hat{\mathbf{y}})' \Big(\mathbf{K}^{\mathbf{d}_t} + \mathbf{E}\Big)(\boldsymbol{\alpha} \odot \hat{\mathbf{y}})\Big).$$

Algorithm 1. Cutting plane algorithm for KI-SVM

1: Initialize $\mathbf{d}$ to $\mathbf{d}_0$, and set $\mathcal{C} = \{\mathbf{d}_0\}$.
2: Run MKL for the subset of kernel matrices selected in $\mathcal{C}$ and obtain $\boldsymbol{\alpha}$ from (10). Let o_1 be the objective value obtained.
3: Find a constraint (indexed by $\hat{\mathbf{d}}$) violated by the current solution and set $\mathcal{C} = \hat{\mathbf{d}} \bigcup \mathcal{C}$.
4: Set $o_2 = o_1$. Run MKL for the subset of kernel matrices selected in $\mathcal{C}$ and obtain $\boldsymbol{\alpha}$ from (10). Let o_1 be the objective value obtained.
5: Repeat steps 3-4 until $|\frac{o_2 - o_1}{o_2}| < \epsilon$.

It can be further noted that $\sum \mu_t = 1$ by setting the derivative w.r.t. θ to zero. Let $\boldsymbol{\mu}$ be the vector of μ_t's, and $\mathcal{M}$ be the simplex $\{\boldsymbol{\mu} \mid \sum \mu_t = 1, \mu_t \geq 0\}$. Then (9) becomes

$$\max_{\boldsymbol{\alpha} \in \mathcal{A}} \min_{\boldsymbol{\mu} \in \mathcal{M}} -\frac{1}{2}(\boldsymbol{\alpha} \odot \hat{\mathbf{y}})' \Big(\sum_{t: \mathbf{d}_t \in \Delta} \mu_t \mathbf{K}^{\mathbf{d}_t} + \mathbf{E} \Big) (\boldsymbol{\alpha} \odot \hat{\mathbf{y}}) \tag{10}$$

$$= \min_{\boldsymbol{\mu} \in \mathcal{M}} \max_{\boldsymbol{\alpha} \in \mathcal{A}} -\frac{1}{2}(\boldsymbol{\alpha} \odot \hat{\mathbf{y}})' \Big(\sum_{t: \mathbf{d}_t \in \Delta} \mu_t \mathbf{K}^{\mathbf{d}_t} + \mathbf{E} \Big) (\boldsymbol{\alpha} \odot \hat{\mathbf{y}}). \tag{11}$$

Here, we can interchange the order of the $\max$ and $\min$ operators as the objective in (10) is concave in $\boldsymbol{\alpha}$ and convex in $\boldsymbol{\mu}$ [13]. It is noteworthy that (11) can be regarded as multiple kernel learning (MKL) [13], where the kernel matrix to be learned is a convex combination of the base kernel matrices $\{\mathbf{K}^{\mathbf{d}_t} : \mathbf{d}_t \in \Delta\}$. However, the number of feasible vectors $\mathbf{d}_t \in \Delta$ is exponential, the set of base kernels is also exponential in size and so direct MKL is still computationally intractable.

In this paper, we apply the cutting plane method [11] to handle this exponential number of constraints. The cutting plane algorithm is described in Algorithm 1. First, as in [1], we initialize $\mathbf{d}_0$ as the average value, i.e., $\{d_{i,j} = 1/m_i, i = 1, \cdots, p; j = 1, \cdots, m_i\}$ and initialize the working set $\mathcal{C}$ to $\{\mathbf{d}_0\}$. Since the size of $\mathcal{C}$ (and thus the number of base kernel matrices) is no longer exponential, one can perform MKL with the subset of kernel matrices in $\mathcal{C}$, obtain $\boldsymbol{\alpha}$ from (10) and record the objective value o_1 in step 2. In step 3, an inequality constraint in (9) (which is indexed by a particular $\hat{\mathbf{d}}$) that is violated by the current solution is then added to $\mathcal{C}$. In step 4, we first set $o_2 = o_1$, then we perform MKL again and record the new objective value o_1. We repeat step 3 and step 4 until the gap between o_1 and o_2 is small enough. ϵ is simply set as 0.001 in our experiments.

Two important issues need to be addressed in the cutting plane algorithm, i.e., how to efficiently solve the MKL problem in Steps 2 and 4 and how to efficiently find the a violated constraint in Step 3? These will be addressed in Sections 3.3 and 3.4, respectively.

3.3 MKL on Subset of Kernel Matrices in $\mathcal{C}$

In recent years, a number of MKL methods have been developed in the literature [2,13,17,18,20,25]. In this paper, an adaptation of the SimpleMKL algorithm [18] is used to solve the MKL problem in Algorithm 1.

Specifically, suppose that the current $\mathcal{C} = \{\mathbf{d}_1, \cdots, \mathbf{d}_T\}$. Recall that the feature map induced by the base kernel matrix $\mathbf{K}^{\mathbf{d}_t}$ is given in (8). As in the derivation of the SimpleMKL algorithm, we consider the following optimization problem that corresponds to the MKL problem in (11).

$$\begin{aligned}\min_{\boldsymbol{\mu}\in\mathcal{M},\mathbf{w},\boldsymbol{\xi}} \ & \frac{1}{2}\sum_{t=1}^{T}\frac{||\mathbf{w}_t||^2}{\mu_t} - \rho + \frac{C}{2}\sum_{i=1}^{p}\xi_i^2 + \frac{\lambda C}{2}\sum_{i=p+1}^{r}\xi_i^2 \\ \text{s.t.} \quad & \sum_{t=1}^{T}\left(\sum_{j=1}^{m_i}\mathbf{w}_t' d_{i,j}^t \phi(\mathbf{x}_{i,j})\right) \geq \rho - \xi_i,\ i = 1, \cdots, p, \\ & -\sum_{t=1}^{T}\mathbf{w}_t'\psi(\hat{\mathbf{x}}_i) \geq \rho - \xi_i, \quad i = p+1, \cdots, r. \end{aligned} \tag{12}$$

It is easy to verify that its dual is

$$\begin{aligned}\max_{\boldsymbol{\alpha}\in\mathcal{A},\theta} \ & -\frac{1}{2}\boldsymbol{\alpha}'\mathbf{E}\boldsymbol{\alpha} - \theta \\ \text{s.t.} \quad & \theta \geq \frac{1}{2}(\boldsymbol{\alpha}\odot\hat{\mathbf{y}})'\mathbf{K}^{\mathbf{d}_t}(\boldsymbol{\alpha}\odot\hat{\mathbf{y}}) \ \ t = 1, \ldots, T, \end{aligned}$$

which is the same as (9). Following SimpleMKL, we solve (11) (or, equivalently, (12)) iteratively. First, by fixing the mixing coefficients $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_T]'$ of the base kernel matrices and we solve the SVM's dual

$$\max_{\boldsymbol{\alpha}\in\mathcal{A}} -\tfrac{1}{2}(\boldsymbol{\alpha}\odot\hat{\mathbf{y}})'\Big(\textstyle\sum_{t=1}^{T}\mu_t\mathbf{K}^{\mathbf{d}_t} + \mathbf{E}\Big)(\boldsymbol{\alpha}\odot\hat{\mathbf{y}}).$$

Then, by fixing $\boldsymbol{\alpha}$, we use the reduced gradient method to update $\boldsymbol{\mu}$. These two steps are iterated until convergence.

3.4 Finding a Violated Constraint

While the cutting plane algorithm only needs to find a violated constraint in each iteration, it is customary to find the most violated constraint. In the context of (9), we then have to find the $\hat{\mathbf{d}}$ that maximizes

$$\max_{\mathbf{d}\in\Delta} \sum\nolimits_{i,j=1}^{r} \alpha_i\alpha_j\hat{y}_i\hat{y}_j(\boldsymbol{\psi}_i^{\mathbf{d}})'(\boldsymbol{\psi}_j^{\mathbf{d}}). \tag{13}$$

However, this is a concave QP and so can not be solved efficiently. Note, however, that while the use of the most violated constraint may lead to faster convergence, the cutting plane method only requires a violated constraint at each iteration. Hence, we propose in the following a simple and efficient method for finding a good approximation of the most violated $\hat{\mathbf{d}}$.

First, note that maximizing (13) could be rewritten as $\| \sum_{i=1}^{r} \alpha_i\ \hat{y}_i\boldsymbol{\psi}_i^{\mathbf{d}} \|_2$. Using the definition of $\boldsymbol{\psi}_i^{\mathbf{d}}$ in (8), this can be rewritten as

$$\max_{\mathbf{d}\in\Delta} \left\| \sum_{i=1}^{p}\alpha_i\sum_{j=1}^{m_i} d_{i,j}\phi(\mathbf{x}_{i,j}) - \sum_{i=p+1}^{r}\alpha_i\psi(\hat{\mathbf{x}}_i) \right\|_2. \tag{14}$$

The key is to replace the ℓ_2-norm above with the infinity-norm. For simplicity, let $\phi(\mathbf{x}) = [x^{(1)}, x^{(2)}, \cdots, x^{(g)}]'$ and $\psi(\hat{\mathbf{x}}) = [\hat{x}^{(1)}, \hat{x}^{(2)}, \cdots, \hat{x}^{(g)}]'$, where g is the dimensionality of $\phi(\mathbf{x})$ and $\psi(\hat{\mathbf{x}})$. Then, we have

$$\begin{aligned}&\max_{\mathbf{d}\in\Delta}\left\|\sum_{i=1}^{p}\alpha_i\sum_{j=1}^{m_i}d_{i,j}\psi(\hat{\mathbf{x}}_{i,j})-\sum_{i=p+1}^{r}\alpha_i\psi(\hat{\mathbf{x}}_i)\right\|_\infty\\&=\max_{l=1,\cdots,g}\max_{\mathbf{d}\in\Delta}\left|\sum_{i=1}^{p}\alpha_i\sum_{j=1}^{m_i}d_{i,j}x_{i,j}^{(l)}-\sum_{i=p+1}^{r}\alpha_i\hat{x}_i^{(l)}\right|.\end{aligned}\tag{15}$$

The absolute sign for each inner subproblem (defined on the lth feature)

$$\max_{\mathbf{d}\in\Delta}\left|\sum_{i=1}^{p}\alpha_i\sum_{j=1}^{m_i}d_{i,j}x_{i,j}^{(l)}-\sum_{i=p+1}^{r}\alpha_i\hat{x}_i^{(l)}\right|.\tag{16}$$

can be removed by writing as the maximum of:

$$\max_{\mathbf{d}\in\Delta}\sum_{i=1}^{p}\alpha_i\sum_{j=1}^{m_i}d_{i,j}x_{i,j}^{(l)}-\sum_{i=p+1}^{r}\alpha_i\hat{x}_i^{(l)},\tag{17}$$

and

$$\max_{\mathbf{d}\in\Delta}-\sum_{i=1}^{p}\alpha_i\sum_{j=1}^{m_i}d_{i,j}x_{i,j}^{(l)}+\sum_{i=p+1}^{r}\alpha_i\hat{x}_i^{(l)}.\tag{18}$$

Recall that each $d_{i,j} \in \{0, 1\}$. Hence, by setting the key instance of (the positive) bag B_i to be the one corresponding to $\arg\max_{1\le j\le m_i} x_{i,j}^{(l)}$, i.e.,

$$d_{i,j}=\begin{cases}1\ j=\arg\max_{1\le j'\le m_i}x_{i,j'}^{(l)},\\0\text{ otherwise,}\end{cases}$$

the maximum in (17) can be obtained as

$$\sum_{i=1}^{p}\alpha_i\max_{1\le j\le m_i}x_{i,j}^{(l)}-\sum_{i=p+1}^{r}\alpha_i\hat{x}_i^{(l)}.\tag{19}$$

Similarly, for (18), we set the key instance of (the positive) bag B_i to be the one corresponding to $\arg\min_{1\le j\le m_i} x_{i,j}^{(l)}$, i.e.,

$$d_{i,j}=\begin{cases}1\ j=\arg\min_{1\le j'\le m_i}x_{i,j'}^{(l)},\\0\text{ otherwise,}\end{cases}$$

then the maximum in (18) is obtained as

$$-\sum_{i=1}^{p}\alpha_i\min_{1\le j\le m_i}x_{i,j}^{(l)}+\sum_{i=p+1}^{r}\alpha_i\hat{x}_i^{(l)}.\tag{20}$$

Algorithm 2. Local search for $\mathbf{d}$. Here, $obj(\mathbf{d})$ is the objective value in (13).

Initialize $\mathbf{d} = \arg\max_{\mathbf{d} \in \{\mathbf{d}_1, \cdots, \mathbf{d}_T, \hat{\mathbf{d}}\}} obj(\mathbf{d})$, $v = obj(\mathbf{d})$.
if $\mathbf{d} = \hat{\mathbf{d}}$ **then**
  return $\mathbf{d}$;
end if
for $i = 1 : p$ **do**
  $\mathbf{d}'_l = \mathbf{d}_l, \forall l \neq i$.
  for $j = 1 : m_i$ **do**
    Set $d'_{i,j} = 1$, $d'_{i,q} = 0 \; \forall q \neq j$
    if $obj(\mathbf{d}') > v$ **then**
      $\mathbf{d} = \mathbf{d}'$ and $v = obj(\mathbf{d}')$.
    end if
  end for
end for
return $\mathbf{d}$;

These two candidate values (i.e., (19) and (20)) are then compared, and the larger value is the solution of the lth subproblem in (16). With g features, there are thus a total of $2g$ candidates for $\hat{\mathbf{d}}$. By evaluating the objective values for these $2g$ candidates, we can obtain the solution of (15) and thus the key instance assignment $\hat{\mathbf{d}}$.

Note that for all the positive bags, both $\max_{1 \leq j \leq m_i} x_{i,j}^{(l)}$ and $\min_{1 \leq j \leq m_i} x_{i,j}^{(l)}$ can be pre-computed. Moreover, this pre-processing takes $O(gJ_p)$ time and space only. When a new $\boldsymbol{\alpha}$ is obtained by SimpleMKL, the processing above takes $O(2gr)$ time. Therefore, $\hat{\mathbf{d}}$ can be solved efficiently without the use of any numeric optimization solver.

However, a deficiency of this infinity-norm approximation is that the $\hat{\mathbf{d}}$ obtained may not always correspond to a violated constraint. As the cutting plane algorithm only requires the addition of a violated constraint at each iteration, a simple local search is used to refine the $\hat{\mathbf{d}}$ solution (Algorithm 2). Specifically, we iteratively update the key instance assignment for each positive bag, while keeping the key instance assignments for all the other positive bags fixed. Finally, the $\mathbf{d}$ that leads to the largest objective value in (13) will be reported.

3.5 Prediction

On prediction, each instance $\mathbf{x}$ can be treated as a bag, and its output from the KI-SVM is given by $f(\mathbf{x}) = \sum_{t=1}^{T} \mu_t \sum_{i=1}^{N} \alpha_i \hat{y}_i (\boldsymbol{\psi}_i^{\mathbf{d}_t})' \phi(\mathbf{x})$.

4 Experiments

In this section, we evaluate the proposed methods on both CBIR image data and benchmark data sets of multi-instance learning.

Table 1. Some statistics of the image data set

concept	#images	average #ROIs per image
castle	100	19.39
firework	100	27.23
mountain	100	24.93
sunset	100	2.32
waterfall	100	13.89

4.1 Locating ROI in Each Image

We employ the image database that has been used by Zhou *et al.* [29] in studying the ROI detection performance of multi-instance learning methods. This database consists of 500 COREL images from five image categories: *castle*, *firework*, *mountain*, *sunset* and *waterfall*. Each category corresponds to a target concept to be retrieved. Moreover, each image is of size 160×160, and is converted to the multi-instance feature representation by using the bag generator SBN [16]. Each region (instance) in the image (bag) is of size 20×20. Some of these regions are labeled manually as ROIs. A summary of the data set is shown in Table 1.

The one-vs-rest strategy is used. In particular, a training set of 50 images is created by randomly sampling 10 images from each of the five categories. The remaining 450 images constitute a test set. The training/test partition is randomly generated 30 times, and the average performance is recorded.

The proposed KI-SVMs are compared with the MI-SVM [1] and two other SVM-based methods in multi-instance learning, namely the mi-SVM [1] and the SVM with a multi-instance kernel (MI-Kernel) [8]. Moreover, we further compare with three state-of-art methods on locating the ROIs, namely, Diverse Density (DD) [15], EM-DD [26] and CkNN-ROI [29]. For the MI-SVM, mi-SVM, MI-Kernel and KI-SVMs, the RBF kernel is used and the parameters are selected using cross-validation on the training sets. Experiments are performed on a PC with 2GHz Intel Xeon(R)2-Duo running Windows XP with 4GB memory.

Following [29], we evaluate the *success rate*, i.e., the ratio of the number of successes divided by the total number of relevant images. For each relevant image in the database, if the ROI returned by the algorithm is a real ROI, then it is counted as a success. For a fair comparison, all the SVM-based methods are only allowed to identify one ROI, which is the region in the image with maximum prediction value.

Table 2 shows the success rates (with standard deviations) of the various methods. Besides, we also show the rank of each method in terms of its success rate. As can be seen, among all the SVM-based methods, Ins-KI-SVM achieves the best performance on all five concepts. As for its performance comparison with the other non-SVM type methods, Ins-KI-SVM is still always better than DD and CkNN-ROI, and is comparable to EM-DD. In particular, EM-DD achieves the best performance on two out of five categories, while Ins-KI-SVM achieves the best performance on the other three. As can be seen, the proposed Bag-KI-SVM also achieves

Table 2. Success rate (%) in locating ROIs. The number in parentheses is the relative rank of the algorithm on the corresponding data set (the smaller the rank, the better the performance).

Method		*castle*	*firework*	*mountain*	*sunset*	*waterfall*	total rank
SVM methods	Ins-KI-SVM	64.74 (2) ±6.64	**83.70 (1)** ±15.43	76.78 (2) ±5.46	**66.85 (1)** ±6.03	**63.41 (1)** ±10.56	7
	Bag-KI-SVM	60.63 (3) ±7.53	54.00 (4) ±22.13	72.70 (3) ±7.66	47.78 (4) ±13.25	45.04 (2) ±21.53	16
	MI-SVM	56.63 (4) ±5.06	58.04 (3) ±20.31	67.63 (5) ±8.43	33.30 (6) ±2.67	33.30 (5) ±8.98	23
	mi-SVM	51.44 (6) ±4.93	40.74 (6) ±4.24	67.37 (6) ±4.48	32.19 (7) ±1.66	22.04 (7) ±4.97	32
	MI-Kernel	50.52 (7) ±4.46	36.37 (8) ±7.92	65.67 (7) ±5.18	32.15 (8) ±1.67	19.93 (8) ±4.65	38
non-SVM methods	DD	35.89 (8) ±15.23	38.67 (7) ±30.67	68.11 (4) ±7.54	57.00 (2) ±18.40	37.78 (4) ±29.61	25
	EM-DD	**76.00 (1)** ±4.63	79.89 (2) ±19.25	**77.22 (1)** ±13.29	53.56 (3) ±16.81	44.33 (3) ±15.13	10
	C*k*NN-ROI	51.48 (5) ±4.59	43.63 (5) ±12.40	60.59 (8) ±4.38	34.59 (5) ±2.57	30.48 (6) ±6.34	29

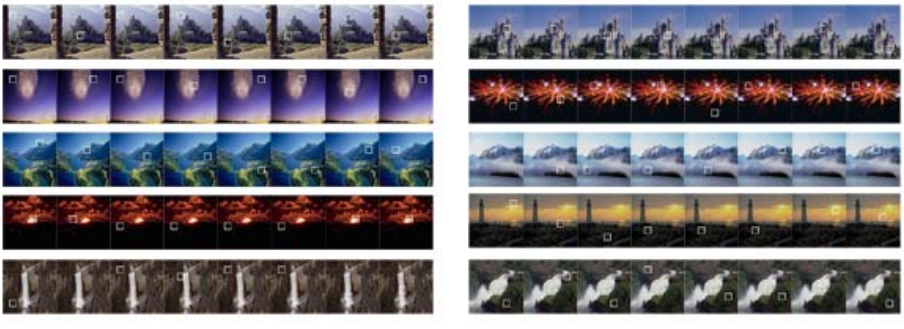

Fig. 1. ROIs located by (from left to right) DD, EM-DD, C*k*NN-ROI, MI-SVM, mi-SVM, MI-Kernel, Ins-KI-SVM and Bag-KI-SVM. Each row shows one category (top to bottom: *castle*, *firework*, *mountain*, *sunset* and *waterfall*).

highly competitive performance with the other state-of-the-art multi-instance learning methods. Fig. 1 shows some example images with the located ROIs. It can be observed that Ins-KI-SVM can correctly identify more ROIs than the other methods.

Each multi-instance algorithm typically has higher confidences (i.e., higher prediction value on the predicted ROI) on some bags than in others. In the next experiment, instead of reporting one ROI in each image, we vary a threshold on the confidence so

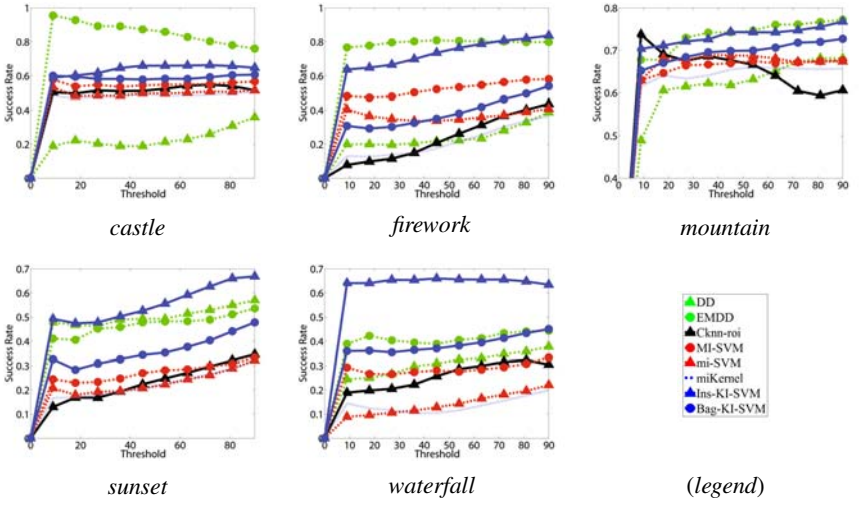

Fig. 2. Success rates when different number of top-confident bags are considered

Table 3. Average wall clock time per query (in seconds)

non-SVM-based methods			SVM-based methods				
DD	EM-DD	CkNN-ROI	MI-SVM	mi-SVM	MI-Kernel	Ins-KI-SVM	Bag-KI-SVM
155.02	15.91	0.003	6.03	6.39	3.04	19.47	5.57

that more than one ROIs can be detected. Fig. 2 shows how the success rate varies when different number of top-confident bags are considered. As can be seen, the proposed Ins-KI-SVM and Bag-KI-SVM achieve highly competitive performance. In particular, Ins-KI-SVM is consistently better than all the other SVM-based methods across all the settings.

Table 3 compares the average query time for the various methods. As can be seen, DD is the slowest since it has to perform gradient descent with multiple restarts. EM-DD is about ten times faster than DD as it involves a much smaller DD optimization at each step. Moreover, note that both the Ins-KI-SVM and mi-SVM work at the instance level while MI-SVM and Bag-KI-SVM work at the bag level. Therefore, MI-SVM and Bag-KI-SVM are in general faster than Ins-KI-SVM and mi-SVM. On the other hand, CkNN-ROI is very efficient as it pre-computes the distances and only needs to compute the citer and reference information when locating ROIs. Moreover, unlike CkNN-ROI which uses the standard Euclidean distance, MI-Kernel needs to compute a small kernel matrix. Therefore, MI-Kernel is slower than CkNN-ROI but is still faster than the other SVM methods in that it only needs to solve the SVM once. However, although CkNN-ROI and MI-Kernel are fast, their performance is much inferior to those of the proposed KI-SVMs, as shown in Table 2.

Table 4. Testing accuracy (%) on the multi-instance classification benchmark data sets

Methods		*Musk1*	*Musk2*	*Elephant*	*Fox*	*Tiger*
SVM-based Methods	Ins-KI-SVM	84.0	84.4	83.5	**63.4**	82.9
	Bag-KI-SVM	**88.0**	82.0	**84.5**	60.5	**85.0**
	MI-SVM	77.9	84.3	81.4	59.4	84.0
	mi-SVM	87.4	83.6	82.0	58.2	78.9
	MI-Kernel	**88.0**	**89.3**	84.3	60.3	84.2
Non-SVM-based Methods	DD	**88.0**	84.0	N/A	N/A	N/A
	EM-DD	84.8	84.9	78.3	56.1	72.1

4.2 Multi-instance Classification

Finally, we evaluate the proposed KI-SVM methods on five multi-instance classification data sets[2] that have been popularly used in the literature [1,5,6,8,28]. These include *Musk1*, *Musk2*, *Elephant*, *Fox* and *Tiger*. The *Musk1* data set contains 47 positive and 45 negative bags, *Musk2* contains 39 positive and 63 negative bags, and each of the remaining three data sets contains 100 positive and 100 negative bags. Details of these data sets can be found in [1,6]. The RBF kernel is used and the parameters are determined by cross-validation on the training set. Comparison is made with the MI-SVM [1], mi-SVM [1], SVM with MI-Kernel [8], DD [15] and EM-DD [26]. Ten-fold cross-validation is used to measure the performance[3]. The average test accuracies of the various methods are shown in Table 4. As can be seen, the performance of KI-SVMs are competitive with all these state-of-the-art methods.

5 Conclusion

Locating ROI is an important problem in many real-world image involved applications. In this paper, we focus on SVM-based methods, and propose two convex optimization methods, Ins-KI-SVM and Bag-KI-SVM, for locating ROIs in images. The KI-SVMs are efficient and based on convex relaxation of the multi-instance SVM. They maximize the margin via generating the most violated key instance step by step, and then combines them via efficient multiple kernel learning. Experiments show that KI-SVMs achieve excellent performance in locating ROIs. The performance of KI-SVMs on multi-instance classification is also competitive with other state-of-the-art methods.

The current work assumes that the bag labels are triggered by single key instances. However, it is very likely that some labels are triggered by several instances together instead of a single key instance. Moreover, some recent studies disclosed that in multi-instance learning the instances should not be treated as i.i.d. samples [27,28]. To identify key instances or key instance groups under these considerations will be studied in the future.

[2] http://www.cs.columbia.edu/∼andrews/mil/datasets.html

[3] The accuracies of these methods were taken from their corresponding literatures. All of them were obtained by ten-fold cross-validation.

Acknowledgements

This research was supported by the National Science Foundation of China (60635030, 60721002), the National High Technology Research and Development Program of China (2007AA01Z169), the Jiangsu Science Foundation (BK2008018), Jiangsu 333 High-Level Talent Cultivation Program, the Research Grants Council of the Hong Kong Special Administrative Region (614907), and the Singapore NTU AcRF Tier-1 Research Grant (RG15/08).

References

1. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Becker, S., Thrun, S., Obermayer, K. (eds.) Advances in Neural Information Processing Systems 15, pp. 577–584. MIT Press, Cambridge (2003)
2. Bach, F.R., Lanckriet, G.R.G., Jordan, M.I.: Multiple kernel learning, conic duality, and the SMO algorithm. In: Proceedings of the 21st International Conference on Machine Learning, Banff, Canada, pp. 41–48 (2004)
3. Bi, J., Chen, Y., Wang, J.Z.: A sparse support vector machine approach to region-based image categorization. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, pp. 1121–1128 (2005)
4. Chen, Y., Wang, J.Z.: Image categorization by learning and reasoning with regions. Journal of Machine Learning Research 5, 913–939 (2004)
5. Cheung, P.M., Kwok, J.T.: A regularization framework for multiple-instance learning. In: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, pp. 193–200 (2006)
6. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence 89(1-2), 31–71 (1997)
7. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using second order information for training support vector machines. Journal of Machine Learning Research 6, 1889–1918 (2005)
8. Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, pp. 179–186 (2002)
9. Hsieh, C.J., Chang, K.W., Lin, C.J., Keerthi, S.S., Sundararajan, S.: A dual coordinate descent method for large-scale linear SVM. In: Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, pp. 408–415 (2008)
10. Joachims, T.: Training linear SVMs in linear time. In: Proceedings to the 12th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, pp. 217–226 (2006)
11. Kelley, J.E.: The cutting plane method for solving convex programs. Journal of the SIAM 8(4), 703–712 (1960)
12. Kim, S.-J., Boyd, S.: A minimax theorem with applications to machine learning, signal processing, and finance. SIAM Journal on Optimization 19(3), 1344–1367 (2008)
13. Lanckriet, G.R.G., Cristianini, N., Bartlett, P., El Ghaoui, L., Jordan, M.I.: Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research 5, 27–72 (2004)
14. Li, Y.-F., Tsang, I.W., Kwok, J.T., Zhou, Z.-H.: Tighter and convex maximum margin clustering. In: Proceeding of the 12th International Conference on Artificial Intelligence and Statistics, Clearwater Beach, FL, pp. 344–351 (2009)

15. Maron, O., Lozano-Pérez, T.: A framework for multiple-instance learning. In: Jordan, M.I., Kearns, M.J., Solla, S.A. (eds.) Advances in Neural Information Processing Systems 10, pp. 570–576. MIT Press, Cambridge (1998)
16. Maron, O., Ratan, A.L.: Multiple-instance learning for natural scene classification. In: Proceedings of the 15th International Conference on Machine Learning, Madison, WI, pp. 341–349 (1998)
17. Rakotomamonjy, A., Bach, F., Canu, S., Grandvalet, Y.: More efficiency in multiple kernel learning. In: Proceedings of the 24th International Conference on Machine Learning, Corvalis, OR, pp. 775–782 (2007)
18. Rakotomamonjy, A., Bach, F.R., Canu, S., Grandvalet, Y.: SimpleMKL. Journal of Machine Learning Research 9, 2491–2521 (2008)
19. Ray, S., Craven, M.: Supervised versus multiple instance learning: an empirical comparison. In: Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, pp. 697–704 (2005)
20. Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large scale multiple kernel learning. Journal of Machine Learning Research 7, 1531–1565 (2006)
21. Tsang, I.W., Kwok, J.T., Cheung, P.: Core vector machines: fast SVM training on very large data sets. Journal of Machine Learning Research 6, 363–392 (2006)
22. Wang, H.Y., Yang, Q., Zha, H.: Adaptive p-posterior mixture-model kernels for multiple instance learning. In: Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, pp. 1136–1143 (2008)
23. Wang, J., Zucker, J.D.: Solving the multiple-instance problem: A lazy learning approach. In: Proceedings of the 17th International Conference on Machine Learning, Stanford, CA, pp. 1119–1125 (2000)
24. Xu, X., Frank, E.: Logistic regression and boosting for labeled bags of instances. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 272–281. Springer, Heidelberg (2004)
25. Xu, Z., Jin, R., King, I., Lyu, M.R.: An extended level method for efficient multiple kernel learning. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems 21, pp. 1825–1832. MIT Press, Cambridge (2009)
26. Zhang, Q., Goldman, S.A.: EM-DD: An improved multiple-instance learning technique. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems 14, pp. 1073–1080. MIT Press, Cambridge (2002)
27. Zhou, Z.-H., Sun, Y.-Y., Li, Y.-F.: Multi-instance learning by treating instances as non-i.i.d. samples. In: Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada (2009)
28. Zhou, Z.-H., Xu, J.-M.: On the relation between multi-instance learning and semi-supervised learning. In: Proceedings of the 24th International Conference on Machine Learning, Corvalis, OR, pp. 1167–1174 (2007)
29. Zhou, Z.-H., Xue, X.-B., Jiang, Y.: Locating regions of interest in CBIR with multi-instance learning techniques. In: Proceedings of the 18th Australian Joint Conference on Artificial Intelligence, Sydney, Australia, pp. 92–101 (2005)
30. Zhou, Z.-H., Zhang, M.-L.: Multi-instance multi-label learning with application to scene classification. In: Schölkopf, B., Platt, J., Hofmann, T. (eds.) Advances in Neural Information Processing Systems 19, pp. 1609–1616. MIT Press, Cambridge (2007)