Optimal Margin Distribution Machine for Multi-Instance Learning

Teng Zhang and **Hai Jin**

National Engineering Research Center for Big Data Technology and System
Services Computing Technology and System Lab, Cluster and Grid Computing Lab
School of Computer Science and Technology, Huazhong University of Science and Technology, China
{tengzhang, hjin}@hust.edu.cn

Abstract

Multi-instance learning (MIL) is a celebrated learning framework where each example is represented as a bag of instances. An example is negative if it has no positive instances, and vice versa if at least one positive instance is contained. During the past decades, various MIL algorithms have been proposed, among which the large margin based methods is a very popular class. Recently, the studies on margin theory disclose that the margin distribution is of more importance to generalization ability than the minimal margin. Inspired by this observation, we propose the multi-instance optimal margin distribution machine, which can identify the key instances via explicitly optimizing the margin distribution. We also extend a stochastic accelerated mirror prox method to solve the formulated minimax problem. Extensive experiments show the superiority of the proposed method.

1 Introduction

Multi-instance learning (MIL) is a celebrated learning framework [Foulds and Frank, 2010; Amores, 2013; Herrera *et al.*, 2016] where each example is represented as a collection (bag) of feature vectors (instances). A bag is negative if it has no positive instances, and vice versa if at least one positive instance is contained. The learner can only access the bag labels, while the instance labels are not available. Compared to the classical supervised learning where each bag just consists of one instance, MIL provides a much more natural representation and is well suited for many complicated problems. For example in drug discovery and development [Dietterich *et al.*, 1997], one molecule (bag) could have many low-energy shapes (instances), and the model should predict whether a new molecule is qualified to make a special drug or not by learning from a set of known molecules. In content-based image retrieval (CBIR) [Zhou *et al.*, 2005], the image (bag) is decompose into several regions (instances), and the system should retrieve all the images that are relevant to the concept queried by users. Besides the prediction of bag labels, detecting the *key instance* which triggers the positive bag label is a more difficult task. For the former, it is to determine which low-energy shapes are responsible for the observed biological activity. For the latter, it is to identify which regions in the image make users have interest in it.

In the past decades, various MIL algorithms have been proposed, e.g., the diverse density (DD) algorithm [Maron and Ratan, 1998] and EM-DD [Zhang and Goldman, 2001], citation-kNN and its variant [Zhou *et al.*, 2005], MI-SVM [Andrews *et al.*, 2003] and its variants [Bi *et al.*, 2005; Li *et al.*, 2009], among which, the large margin based methods have always been popular. More specifically, MI-SVM starts with a SVM using some multi-instance kernel [Gärtner *et al.*, 2002] and identifies the key instances according to the decision values, after that retrains the SVM model based on the new key instance assignments (bag labels). The convergence of this procedure can be easily guaranteed once it is viewed as a specialization of the constrained concave-convex programming (CCCP) method. Although in each iteration, MI-SVM only solves a SVM-like convex optimization, the whole problem is still non-convex and thus it may get stuck in the local minima. On the other hand, the KI-SVM overcomes this difficulty by relaxing the mixed-integer programming as a convex optimization by minimax saddle point theory. It applies the cutting-plane method for optimization by generating a violated key instance assignment (kernel) to the constraint set in each iteration.

Aforementioned methods are based on the large margin principle, i.e., maximizing the minimal distance from the instances to the decision hyperplane. Recently, the studies on margin theory [Gao and Zhou, 2013] show that margin distribution is of more importance to generalization ability than minimal margin, which gives rise to the optimal margin distribution learning ([Zhang and Zhou, 2019]). Due to the superiority to the traditional large margin based methods, this new learning paradigm has quickly attracted a lot of attentions and been extended to many learning settings ([Zhang and Zhou, 2018a; Zhang and Zhou, 2018b; Tan *et al.*, 2020]). These great successes suggest that the existing large margin based MIL algorithms still have enough room for enhancement.

Based on this recognition, we propose the multi-instance optimal margin distribution machine (MI-ODM). It can identify the key instance via explicitly optimizing the margin distribution. Specifically, we characterize the margin distribution by its first- and second-order statistics, i.e., the mar-

gin mean and margin variance. As suggested in [Gao and Zhou, 2013], we maximize the former and minimize the latter simultaneously. To solve the resultant minimax saddle point problem, we extend a stochastic accelerated mirror prox method which enjoys the optimal convergence rate. Extensive experiments verify the superiority of the proposed method.

The rest of the paper is organized as follows. We first introduce some preliminaries, and then present the proposed MI-ODM. After that we detail the optimization techniques, followed by the experimental results and empirical observations. Finally we conclude the paper with future work.

2 Preliminaries

For convenience, we first make some notation conventions. Throughout the paper, we denote scalars with lower case letters (e.g., y), and vectors with bold face letters (e.g., $\boldsymbol{x}$). Sets are designated by upper case letters with mathcal font (e.g., $\mathcal{S}$). Let $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} = \{1, -1\}$ denote the input and output spaces, respectively. For any $m \geq 1$, the set of integers $\{1, \ldots, m\}$ is denoted by $[m]$. The feature mapping associated to some positive definite kernel κ is denoted by $\phi : \mathcal{X} \mapsto \mathbb{H}$.

2.1 ODM

The margin $\gamma(\boldsymbol{x}, y)$ of a labeled instance $(\boldsymbol{x}, y)$ is defined as the signed decision value, i.e., $\gamma(\boldsymbol{x}, y) = y\boldsymbol{w}^\top \phi(\boldsymbol{x})$.[1] This value can be viewed as the confidence (or safety) of the prediction. The larger the margin, the more confidence we have on the predicted label, and $(\boldsymbol{x}, y)$ is misclassified if and only if it produces a negative margin.

It is well known that SVMs employ the large margin principle to pick the decision boundary [Cristianini and Shawe-Taylor, 2000]. As a result, the obtained separating hyperplane just consists of a small amount of instances, a.k.a. the support vectors (SVs), and the rest instances are totally ignored. When noisy instances exist, the learner may be misled and produce a suboptimal decision boundary [Zhou, 2014].

As a counterpart, optimizing the margin distribution is a more robust strategy by exploiting the whole data set and preventing from being cheated by the noisy instances. As for how to characterize the margin distribution, a straightforward way is through the first- and second- statistics, i.e., the margin mean and variance. Moreover, as suggested in [Gao and Zhou, 2013], maximizing the former and minimizing the latter simultaneously can yield a tighter generalization bound, the optimal margin distribution machine (ODM) is initially formulated as:

$$\begin{aligned} \min_{\boldsymbol{w},\bar{\gamma},\xi_i,\epsilon_i} \ & \frac{1}{2}\|\boldsymbol{w}\|^2 - \eta\bar{\gamma} + \frac{\lambda}{m}\sum_{i\in[m]}(\xi_i^2 + \epsilon_i^2), \\ \text{s.t. } & \gamma(\boldsymbol{x}_i, y_i) \geq \bar{\gamma} - \xi_i, \\ & \gamma(\boldsymbol{x}_i, y_i) \leq \bar{\gamma} + \epsilon_i, \ \forall i \in [m], \end{aligned} \tag{1}$$

where η and λ are the parameters for trading-off the regularization, and $\bar{\gamma}$ is the margin mean. Note that ξ_i and ϵ_i are deviations of $\gamma(\boldsymbol{x}_i, y_i)$ from the margin mean, thus the last term $\sum_{i\in[m]}(\xi_i^2 + \epsilon_i^2)/m$ is exactly the margin variance.

To make the model more clean and powerful, ODM further introduces three modifications to Eqn. (1). First is simplifying the formulation by fixing the margin mean as one.[2] Second is assigning different weights to different deviations respectively. Third is tolerating the deviation smaller than the given threshold θ to achieve a sparse solution. Therefore, the final formulation of ODM is:

$$\begin{aligned} \min_{\boldsymbol{w},\xi_i,\epsilon_i} \ & F(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\lambda}{m}\sum_{i\in[m]}\frac{\xi_i^2 + \nu\epsilon_i^2}{(1-\theta)^2}, \\ \text{s.t. } & y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i) \geq 1 - \theta - \xi_i, \\ & y_i\boldsymbol{w}^\top\phi(\boldsymbol{x}_i) \leq 1 + \theta + \epsilon_i, \ \forall i \in [m]. \end{aligned} \tag{2}$$

where ν is the weight for trading-off different deviations, and $(1-\theta)^2$ is to scale the second term as a surrogate loss.

3 The Proposed Method

Given a training set of m bags $\mathcal{S} = \{\mathcal{B}_i, y_i\}_{i\in[m]}$ where $\mathcal{B}_i = \{\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,m_i}\}$ is the i-th bag, $y_i \in \{\pm 1\}$ is the label and m_i is the number of instances in bag $\mathcal{B}_i$, we assume the first p bags are positive and the rest $q = m - p$ bags are negative without loss of generality, i.e., all bags are ordered such that

$$y_i = \begin{cases} 1 & i \in [p], \\ -1 & i \in [m] \setminus [p]. \end{cases}$$

The prediction of a bag is determined by the largest decision value of its instances, i.e., $f(\mathcal{B}_i) = \max_{j\in[m_i]} \boldsymbol{w}^\top\phi(\boldsymbol{x}_{i,j})$. Substituting into Eqn. (2), we get

$$\begin{aligned} \min_{\boldsymbol{w},\xi_i,\epsilon_i} \ & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\lambda_1}{p}\sum_{i=1}^{p}\frac{\xi_i^2 + \nu\epsilon_i^2}{(1-\theta)^2} + \frac{\lambda_2}{q}\sum_{i=p+1}^{m}\frac{\xi_i^2 + \nu\epsilon_i^2}{(1-\theta)^2}, \\ \text{s.t. } & y_i \max_{j\in[m_i]} \boldsymbol{w}^\top\phi(\boldsymbol{x}_{i,j}) \geq 1 - \theta - \xi_i, \\ & y_i \max_{j\in[m_i]} \boldsymbol{w}^\top\phi(\boldsymbol{x}_{i,j}) \leq 1 + \theta + \epsilon_i, \ \forall i \in [m], \end{aligned} \tag{3}$$

where λ_1, λ_2 are the parameters for trading-off empirical losses on positive and negative bags, respectively.

For each positive bag $\mathcal{B}_i$, we introduce a binary vector

$$\boldsymbol{a}_i = [a_{i,1}; \ldots; a_{i,m_i}] \in \{0, 1\}^{m_i}$$

to indicate the key instance with the largest decision value. Following the traditional MIL setting, we assume that each positive bag has only one key instance,[3] and hence we have $\boldsymbol{e}^\top\boldsymbol{a}_i = 1$, where $\boldsymbol{e}$ is an all-one column vector. In the following, let $\boldsymbol{c} = [\boldsymbol{a}_1; \ldots; \boldsymbol{a}_p]$ and $\mathcal{C}$ be its domain, then the constraints $y_i \max_{j\in[m_i]} \boldsymbol{w}^\top\phi(\boldsymbol{x}_{i,j}) \geq 1-\theta-\xi_i$ corresponding to positive bags in Eqn. (3) can be equivalently rewritten as $\max_{\boldsymbol{a}_i} \sum_{j\in[m_i]} a_{i,j}\boldsymbol{w}^\top\phi(\boldsymbol{x}_{i,j}) \geq 1 - \theta - \xi_i$.

[1] Often an offset term b is included, but as this can be implemented by augmenting each $\boldsymbol{x}$ with an additional element whose value is always one, we do not explicitly include it here.

[2] Note that scaling $\boldsymbol{w}$ does not affect the prediction.

[3] Sometimes one may want the positive bag has more than one key instances. The proposed method can be simply extended to this case by setting $\boldsymbol{e}^\top\boldsymbol{a}_i = k$.

For each negative bag $\mathcal{B}_i$ whose instances are all negative, the corresponding constraints Eqn. (3) can be replaced by

$$\begin{cases} -\boldsymbol{w}^\top \phi(\boldsymbol{x}_{i,j}) \geq 1 - \theta - \xi_i, \\ -\boldsymbol{w}^\top \phi(\boldsymbol{x}_{i,j}) \leq 1 + \theta + \epsilon_i, \ \forall j \in [m_i]. \end{cases}$$

Moreover, to make the model more relaxable, we allow the bags have different slack variables, i.e.,

$$\{\xi_{s(i,j)}\}_{i\in[m]\setminus[p],\, j\in[m_i]}, \quad \{\epsilon_{s(i,j)}\}_{i\in[m]\setminus[p],\, j\in[m_i]},$$

where index $s(i,j) = J_{i-1} - J_p + j + p$ ranges from $p+1$ to $J_m - J_p + p$ and $J_i = \sum_{t=1}^{i} m_t$ (J_0 is set to 0). Combining all these together, Eqn. (3) turns into

$$\begin{aligned} \min_{\boldsymbol{c}\in\mathcal{C}} \min_{\boldsymbol{w},\xi_i,\epsilon_i} \ & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\lambda_1}{p}\sum_{i=1}^{p}\frac{\xi_i^2 + \nu\epsilon_i^2}{(1-\theta)^2} \\ & + \frac{\lambda_2}{q}\sum_{i=p+1}^{m}\sum_{j\in[m_i]}\frac{\xi_{s(i,j)}^2 + \nu\epsilon_{s(i,j)}^2}{(1-\theta)^2}, \\ \text{s.t.} \ & \sum_{j\in[m_i]} a_{i,j}\boldsymbol{w}^\top\phi(\boldsymbol{x}_{i,j}) \geq 1 - \theta - \xi_i, \\ & \sum_{j\in[m_i]} a_{i,j}\boldsymbol{w}^\top\phi(\boldsymbol{x}_{i,j}) \leq 1 + \theta + \epsilon_i, \ \forall i \in [p], \\ & -\boldsymbol{w}^\top\phi(\boldsymbol{x}_{i,j}) \geq 1 - \theta - \xi_{s(i,j)}, \\ & -\boldsymbol{w}^\top\phi(\boldsymbol{x}_{i,j}) \leq 1 + \theta + \epsilon_{s(i,j)}, \\ & \qquad \forall i \in [m]\setminus[p], \ \forall j \in [m_i]. \end{aligned} \tag{4}$$

As kernel methods, the inner minimization of Eqn. (4) is usually processed via the dual form due to the underlying infinite dimensional feature mapping. Introduce the dual variables $\boldsymbol{u} = [u_1; \ldots; u_{2(J_m - J_p + p)}] \succeq \boldsymbol{0}$, the Lagrangian of Eqn. (4) leads to

$$\begin{aligned} \min_{\boldsymbol{c}\in\mathcal{C}} \max_{\boldsymbol{u}\in\mathcal{U}} \ & -\frac{1}{2}\boldsymbol{u}^\top \begin{bmatrix} \mathbf{K} & -\mathbf{K} \\ -\mathbf{K} & \mathbf{K} \end{bmatrix} \boldsymbol{u} \\ & -\frac{(1-\theta)^2}{4}\boldsymbol{u}^\top \begin{bmatrix} \frac{1}{\lambda_1}\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\lambda_2\nu}\mathbf{I} \end{bmatrix} \boldsymbol{u} - \begin{bmatrix} (\theta-1)\boldsymbol{e} \\ (\theta+1)\boldsymbol{e} \end{bmatrix}^\top \boldsymbol{u}, \end{aligned} \tag{5}$$

where $\mathcal{U}$ is the non-negative quadrant and $\mathbf{K}_{i,j} = \boldsymbol{\Psi}_i^\top \boldsymbol{\Psi}_j \in \mathbb{R}^{q\times q}$ is the kernel matrix with

$$\boldsymbol{\Psi}_i = \begin{cases} \sum_{j\in[m_i]} a_{i,j}\phi(\boldsymbol{x}_{i,j}) & i \in [p], \ j \in [m_i], \\ -\phi(\boldsymbol{x}_{i,j}) & i \in [m]\setminus[p], \ j \in [m_i]. \end{cases}$$

To overcome the difficulty caused by the mixed-integer programming, some convex relaxation methods should be applied, e.g., the semi-definite programming relaxation [Xu *et al.*, 2004], and the minimax relaxation [Li *et al.*, 2009]. Since the latter is proven to be tighter than the former, we also employ minimax relaxation in this paper. Interchanging the order of $\max_{\boldsymbol{u}\in\mathcal{U}}$ and $\min_{\boldsymbol{c}\in\mathcal{C}}$, we have

$$\max_{\boldsymbol{u}\in\mathcal{U}} \min_{\boldsymbol{c}\in\mathcal{C}} \ D(\boldsymbol{u}, \boldsymbol{c}),$$

where $D(\boldsymbol{u}, \boldsymbol{c})$ denotes the objective function of Eqn. (5). Moreover, with an equivalent rewriting for the inner optimization, the above formulation turns to

$$\max_{\boldsymbol{u}\in\mathcal{U}} \max_{d} \ d \quad \text{s.t. } D(\boldsymbol{u}, \boldsymbol{c}_k) \geq d, \ \forall \boldsymbol{c}_k \in \mathcal{C}. \tag{6}$$

Again introduce the dual variables $\boldsymbol{v} = [v_1; \ldots; v_{|\mathcal{C}|}] \succeq \boldsymbol{0}$ for the inner optimization, the Lagrangian of Eqn. (6) leads to

$$\min_{\boldsymbol{v}\succeq\boldsymbol{0}} \max_{d} \ \{d + \sum_{k:\boldsymbol{c}_k\in\mathcal{C}} v_k(D(\boldsymbol{u}, \boldsymbol{c}_k) - d)\}.$$

Setting the derivative of d to zero, we have $\sum_{k:\boldsymbol{c}_k\in\mathcal{C}} v_k = 1$ and the dual problem turns to

$$\min_{\boldsymbol{v}\in\mathcal{V}} \sum_{k:\boldsymbol{c}_k\in\mathcal{C}} v_k D(\boldsymbol{u}, \boldsymbol{c}_k), \tag{7}$$

where $\mathcal{V} = \{\boldsymbol{v} \in \mathbb{R}_+^{|\mathcal{C}|} \mid \boldsymbol{e}^\top\boldsymbol{v} = 1\}$ is the simplex in $\mathbb{R}^{|\mathcal{C}|}$. For simplicity, denote $\sum_{k:\boldsymbol{c}_k\in\mathcal{C}} v_k D(\boldsymbol{u}, \boldsymbol{c}_k)$ as $G(\boldsymbol{u}, \boldsymbol{v})$, and substitute Eqn. (7) into Eqn. (6), we have

$$\max_{\boldsymbol{u}\in\mathcal{U}} \min_{\boldsymbol{v}\in\mathcal{V}} \ G(\boldsymbol{u}, \boldsymbol{v}).$$

Note that $G(\boldsymbol{u}, \boldsymbol{v})$ is a convex combination of negative definite quadratic functions, thus it is convex in $\boldsymbol{v}$ and concave in $\boldsymbol{u}$, and according to Siont's minimax theorem [Sion, 1958], there exists a saddle point $(\boldsymbol{u}^\star, \boldsymbol{v}^\star) \in \mathcal{U} \times \mathcal{V}$ such that

$$\begin{aligned} \min_{\boldsymbol{v}\in\mathcal{V}} \max_{\boldsymbol{u}\in\mathcal{U}} G(\boldsymbol{u}, \boldsymbol{v}) &\leq \max_{\boldsymbol{u}\in\mathcal{U}} G(\boldsymbol{u}, \boldsymbol{v}^\star) = G(\boldsymbol{u}^\star, \boldsymbol{v}^\star) \\ &= \min_{\boldsymbol{v}\in\mathcal{V}} G(\boldsymbol{u}, \boldsymbol{v}^\star) \leq \max_{\boldsymbol{u}\in\mathcal{U}} \min_{\boldsymbol{v}\in\mathcal{V}} G(\boldsymbol{u}, \boldsymbol{v}). \end{aligned} \tag{8}$$

Combining with the minimax inequality

$$\max_{\boldsymbol{u}\in\mathcal{U}} \min_{\boldsymbol{v}\in\mathcal{V}} G(\boldsymbol{u}, \boldsymbol{v}) \leq \min_{\boldsymbol{v}\in\mathcal{V}} \max_{\boldsymbol{u}\in\mathcal{U}} G(\boldsymbol{u}, \boldsymbol{v}),$$

all the inequalities in Eqn. (8) hold as equations, thus the MIODM finally can be formulated as

$$\min_{\boldsymbol{v}\in\mathcal{V}} \max_{\boldsymbol{u}\in\mathcal{U}} \ G(\boldsymbol{u}, \boldsymbol{v}). \tag{9}$$

and the optimal solution is the saddle point $(\boldsymbol{u}^\star, \boldsymbol{v}^\star)$.

4 Optimization

In this section, we first give a simple introduction to the minimax problem. After that, we detail the stochastic accelerated mirror prox method which can quickly find the optimal solution.

4.1 Minimax Problem

Note that $G(\boldsymbol{u}, \cdot)$ and $-G(\cdot, \boldsymbol{v})$ are both convex functions, according to the first order inequality of convexity, for any pair $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}) \in \mathcal{U} \times \mathcal{V}$, we have

$$\begin{aligned} G(\boldsymbol{u}, \hat{\boldsymbol{v}}) - G(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}) &\leq -\partial_{\boldsymbol{u}} G(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})^\top(\hat{\boldsymbol{u}} - \boldsymbol{u}), \ \forall \boldsymbol{u} \in \mathcal{U}, \\ G(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}) - G(\hat{\boldsymbol{u}}, \boldsymbol{v}) &\leq \partial_{\boldsymbol{v}} G(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})^\top(\hat{\boldsymbol{v}} - \boldsymbol{v}), \ \forall \boldsymbol{v} \in \mathcal{V}. \end{aligned}$$

Adding the above two inequalities together and augmenting $\boldsymbol{u}$ and $\boldsymbol{v}$, we have

$$G(\boldsymbol{u}, \hat{\boldsymbol{v}}) - G(\hat{\boldsymbol{u}}, \boldsymbol{v}) \leq g(\hat{\boldsymbol{w}})^\top(\hat{\boldsymbol{w}} - \boldsymbol{w}), \ \forall \boldsymbol{u}, \boldsymbol{v}, \tag{10}$$

where $\boldsymbol{w} = [\boldsymbol{u}; \boldsymbol{v}]$ and $g(\hat{\boldsymbol{w}}) = [-\partial_{\boldsymbol{u}} G(\hat{\boldsymbol{w}}); \partial_{\boldsymbol{v}} G(\hat{\boldsymbol{w}})]$. Compared to the general convex optimization, it can be found that $g(\hat{\boldsymbol{w}})$ plays a similar role as "gradient". Since Eqn. (10) holds for any $\boldsymbol{u}$ and $\boldsymbol{v}$, particularly we have

$$\max_{\boldsymbol{u}\in\mathcal{U}} G(\boldsymbol{u}, \hat{\boldsymbol{v}}) - \min_{\boldsymbol{v}\in\mathcal{V}} G(\hat{\boldsymbol{u}}, \boldsymbol{v}) \leq g(\hat{\boldsymbol{w}})^\top(\hat{\boldsymbol{w}} - \boldsymbol{w}). \tag{11}$$

The LHS of Eqn. (11) can be further decomposed as two gaps between current point $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}})$ and saddle point $(\boldsymbol{u}^\star, \boldsymbol{v}^\star)$:

$$\begin{aligned}&\max_{\boldsymbol{u}\in\mathcal{U}} G(\boldsymbol{u}, \hat{\boldsymbol{v}}) - G(\boldsymbol{u}^\star, \boldsymbol{v}^\star) + G(\boldsymbol{u}^\star, \boldsymbol{v}^\star) - \min_{\boldsymbol{v}\in\mathcal{V}} G(\hat{\boldsymbol{u}}, \boldsymbol{v}) \\ &= \underbrace{\max_{\boldsymbol{u}\in\mathcal{U}} G(\boldsymbol{u}, \hat{\boldsymbol{v}}) - \min_{\boldsymbol{v}\in\mathcal{V}} \max_{\boldsymbol{u}\in\mathcal{U}} G(\boldsymbol{u}, \boldsymbol{v})}_{\geq 0} \\ &\qquad + \underbrace{\max_{\boldsymbol{u}\in\mathcal{U}} \min_{\boldsymbol{v}\in\mathcal{V}} G(\boldsymbol{u}, \boldsymbol{v}) - \min_{\boldsymbol{v}\in\mathcal{V}} G(\hat{\boldsymbol{u}}, \boldsymbol{v})}_{\geq 0}.\end{aligned}$$

Since the two gaps are both non-negative, and the smaller the two gaps, the closer to the saddle point, the LHS of Eqn. (11) can be viewed as the "duality gap" in the general convex optimization and serves as a stopping criteria for the algorithm design.

4.2 Stochastic Accelerated Mirror Prox

The feasible field of $\boldsymbol{u}$ and $\boldsymbol{v}$ are box and simplex respectively. To exploit this structural information, we resort to the mirror descent method [Beck and Teboulle, 2003]. Specifically, for variable $\boldsymbol{u}$, the common Euclidean distance mirror map $\psi_{\mathcal{U}}(\boldsymbol{u}) = \|\boldsymbol{u}\|_2^2/2$ can work well, while for variable $\boldsymbol{v}$, the negative entropy mirror map $\psi_{\mathcal{V}}(\boldsymbol{v}) = \sum_k v_k \log v_k$ is most suitable, since it can make the time complexity only have a logarithmic dependence on the dimension.

The mirror descent style methods perform gradient descent in the dual space induced by the mirror maps. To make the minimax structure more easily handled like the general optimization problem, we introduce the joint mirror map $\psi(\boldsymbol{w}) = a\psi_{\mathcal{U}}(\boldsymbol{u}) + b\psi_{\mathcal{V}}(\boldsymbol{v})$, where $a = \sqrt{2}/\tau\sqrt{J_m - J_p + p}$ and $b = 1/\sqrt{\log|\mathcal{C}|}$, respectively. It can be shown that $\nabla\psi_{\mathcal{U}}(\boldsymbol{u}) = \boldsymbol{u}$ and $\nabla\psi_{\mathcal{V}}(\boldsymbol{v}) = \log \boldsymbol{v} + \boldsymbol{e}$. Combining the two components together we have $\nabla\psi(\boldsymbol{w}) = [a\boldsymbol{u}; b\log \boldsymbol{v} + b\boldsymbol{e}]$.

As shown in Figure 1, at the t-th iteration, we first map the current point $\boldsymbol{w}_t = [\boldsymbol{u}_t; \boldsymbol{v}_t]$ into the dual space $\nabla\psi(\boldsymbol{w}_t) = [a\boldsymbol{u}_t; b\log \boldsymbol{v}_t + b\boldsymbol{e}]$ and perform one step of gradient descent

$$\begin{aligned}&\nabla\psi(\hat{\boldsymbol{w}}_t) = \nabla\psi(\boldsymbol{w}_t) - \eta g(\boldsymbol{w}_t) \\ &= [a\boldsymbol{u}_t + \eta\partial_{\boldsymbol{u}} G(\boldsymbol{u}_t, \boldsymbol{v}_t); b\log \boldsymbol{v}_t + b\boldsymbol{e} - \eta\partial_{\boldsymbol{v}} G(\boldsymbol{u}_t, \boldsymbol{v}_t)],\end{aligned}$$

where η is the step size, then map the $\nabla\psi(\hat{\boldsymbol{w}}_t)$ back to primal space, i.e., to find $\hat{\boldsymbol{w}}_t = [\hat{\boldsymbol{u}}_t; \hat{\boldsymbol{v}}_t]$ such that

$$\begin{bmatrix} a\hat{\boldsymbol{u}}_t \\ b\log \hat{\boldsymbol{v}}_t + b\boldsymbol{e} \end{bmatrix} = \begin{bmatrix} a\boldsymbol{u}_t + \eta\partial_{\boldsymbol{u}} G(\boldsymbol{u}_t, \boldsymbol{v}_t) \\ b\log \boldsymbol{v}_t + b\boldsymbol{e} - \eta\partial_{\boldsymbol{v}} G(\boldsymbol{u}_t, \boldsymbol{v}_t) \end{bmatrix},$$

which implies that $\hat{\boldsymbol{u}}_t = \boldsymbol{u}_t + \eta\partial_{\boldsymbol{u}} G(\boldsymbol{u}_t, \boldsymbol{v}_t)/a$ and $\hat{\boldsymbol{v}}_t = \boldsymbol{v}_t \exp(-\eta\partial_{\boldsymbol{v}} G(\boldsymbol{u}_t, \boldsymbol{v}_t)/b)$. Finally, we project $[\hat{\boldsymbol{u}}_t; \hat{\boldsymbol{v}}_t]$ back to $\mathcal{U}\times\mathcal{V}$ based on the Bregman distance induced by the mirror maps. To be specific, the Euclidean distance mirror map induces the common Euclidean distance, while the negative entropy mirror map induces the Kullback-Leibler divergence. This can be formulated as the following two optimization sub-problems:

$$\begin{aligned}\boldsymbol{u}_{t+1} &= \arg\min_{\boldsymbol{u}\in\mathcal{U}} \|\boldsymbol{u} - \hat{\boldsymbol{u}}_t\|_2^2, \\ \boldsymbol{v}_{t+1} &= \arg\min_{\boldsymbol{v}\in\mathcal{V}} \boldsymbol{v}^\top \log \frac{\boldsymbol{v}}{\hat{\boldsymbol{v}}_t}.\end{aligned}$$

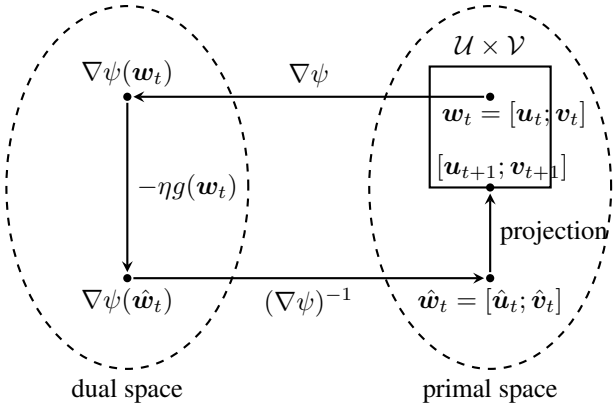

Figure 1: One iteration of mirror descent.

Fortunately, both two problems have a closed-form solution. The former is to project $\hat{\boldsymbol{u}}_t$ onto the non-negative quadrant, thus $\boldsymbol{u}_{t+1} = \max\{\hat{\boldsymbol{u}}_t, \boldsymbol{0}\}$. For the latter, we introduce the dual variable z, and the Lagrangian leads to

$$\max_z \min_{\boldsymbol{v}} \ \boldsymbol{v}^\top \log(\boldsymbol{v}/\hat{\boldsymbol{v}}_t) + z(\boldsymbol{e}^\top \boldsymbol{v} - 1).$$

Setting the derivative of $\boldsymbol{v}$ to zero, we have $\log(\boldsymbol{v}/\hat{\boldsymbol{v}}_t) + \boldsymbol{e} + z\boldsymbol{e} = \boldsymbol{0}$, which implies that $\boldsymbol{v}_{t+1} = \hat{\boldsymbol{v}}_t \exp(-1 - z)$. Note that $\boldsymbol{v}_{t+1}$ belongs to a simplex, hence

$$1 = \boldsymbol{e}^\top \boldsymbol{v}_{t+1} = \boldsymbol{e}^\top \hat{\boldsymbol{v}}_t \exp(-1 - z) = \|\hat{\boldsymbol{v}}_t\|_1 \exp(-1 - z).$$

Substituting with $\exp(-1-z) = 1/\|\hat{\boldsymbol{v}}_t\|_1$, we get the closed-form solution $\boldsymbol{v}_{t+1} = \hat{\boldsymbol{v}}_t/\|\hat{\boldsymbol{v}}_t\|_1$.

Once we have $\boldsymbol{y}_{t+1} \triangleq [\boldsymbol{u}_{t+1}; \boldsymbol{v}_{t+1}]$, repeat the above procedure from $\boldsymbol{w}_t$ one more time, except that when performing gradient descent in the dual space, we use the gradient at $\boldsymbol{y}_{t+1}$ rather than $\boldsymbol{w}_t$. In other words, a two-step mirror descent is carried out in each iteration, which starts from the same point but the gradient used in the second time is evaluated at the ending point of the first time. This is exactly the mirror prox method [Nemirovski, 2005], which has been proved enjoying better convergence rate. Figure 2 illustrates one iteration of this method.

The mirror prox method can be further accelerated via the Nesterov accelerated technique [Nesterov, 2003]. The intuition is that besides $\{\boldsymbol{w}_t\}$ and $\{\boldsymbol{y}_t\}$, we also maintain another two sequences $\{\underline{\boldsymbol{w}}_t\}$ and $\{\overline{\boldsymbol{w}}_t\}$, which are the convex combination of $\{\boldsymbol{w}_t\}$ and $\{\boldsymbol{y}_t\}$. Specifically, at the t-th iteration, first update $\underline{\boldsymbol{w}}_t = (1-\gamma_t)\overline{\boldsymbol{w}}_t + \gamma_t \boldsymbol{y}_t$, where γ_t is the Nesterov accelerated coefficient, usually set as $2/(t+1)$. After that, perform two-step mirror descent based on $\underline{\boldsymbol{w}}_t$ to get $\boldsymbol{y}_{t+1}$ and $\boldsymbol{w}_{t+1}$. Finally, update update $\overline{\boldsymbol{w}}_{t+1} = (1-\gamma_t)\overline{\boldsymbol{w}}_t + \gamma_t \boldsymbol{w}_{t+1}$.

Moreover, to make the method scale well for big data, we also extend a stochastic version of our method, and the key problem turns to finding the unbiased noisy gradient $\partial_{\boldsymbol{u}}\widetilde{G}(\boldsymbol{u}_t, \boldsymbol{v}_t)$ and $\partial_{\boldsymbol{v}}\widetilde{G}(\boldsymbol{u}_t, \boldsymbol{v}_t)$. Note that $G(\boldsymbol{u}, \boldsymbol{v}) = \sum_{k:\boldsymbol{c}_k\in\mathcal{C}} v_k D(\boldsymbol{u}, \boldsymbol{c}_k)$, we have

$$\begin{aligned}\partial_{\boldsymbol{u}} G(\boldsymbol{u}_t, \boldsymbol{v}_t) &= [\partial_{\boldsymbol{u}} D(\boldsymbol{u}_t, \boldsymbol{c}_1), \ldots, \partial_{\boldsymbol{u}} D(\boldsymbol{u}_t, \boldsymbol{c}_{|\mathcal{C}|})]\boldsymbol{v}_t, \\ \partial_{\boldsymbol{v}} G(\boldsymbol{u}_t, \boldsymbol{v}_t) &= [D(\boldsymbol{u}_t, \boldsymbol{c}_1), \ldots, D(\boldsymbol{u}_t, \boldsymbol{c}_{|\mathcal{C}|})].\end{aligned}$$

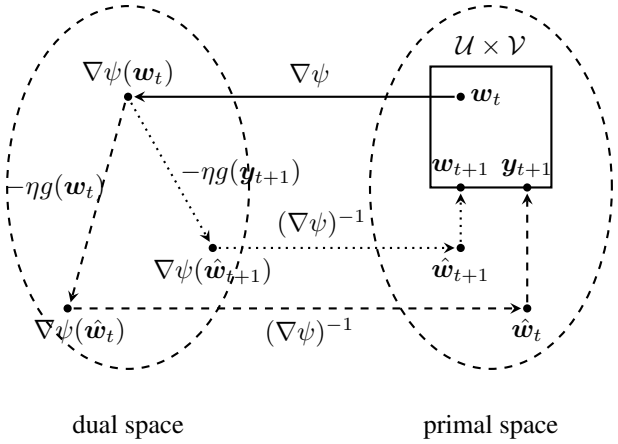

Figure 2: One iteration of mirror prox. The two-step mirror descent both starts from the same point but with gradients evaluated at different points.

Randomly sampling an index i_t according to the distribution $\boldsymbol{v}_t$ on $\{1, 2, \ldots, |\mathcal{C}|\}$, we can obtain $\partial_{\boldsymbol{u}}\widetilde{G}(\boldsymbol{u}_t, \boldsymbol{v}_t, i_t) = \partial_{\boldsymbol{u}} D(\boldsymbol{u}_t, \boldsymbol{c}_{i_t})$. On the other hand, uniformly sampling an index j_t from $\{1, 2, \ldots, |\mathcal{C}|\}$, we can obtain $\partial_{\boldsymbol{v}}\widetilde{G}(\boldsymbol{u}_t, \boldsymbol{v}_t, j_t) = [0, \ldots, |\mathcal{C}| D(\boldsymbol{u}_t, \boldsymbol{c}_{j_t}) \ldots, 0]$. Thus it can be shown that

$$\mathbb{E}[\partial_{\boldsymbol{u}}\widetilde{G}(\boldsymbol{u}_t, \boldsymbol{v}_t, i_t) \mid \boldsymbol{u}_t, \boldsymbol{v}_t] = \partial_{\boldsymbol{u}} G(\boldsymbol{u}_t, \boldsymbol{v}_t),$$
$$\mathbb{E}[\partial_{\boldsymbol{v}}\widetilde{G}(\boldsymbol{u}_t, \boldsymbol{v}_t, j_t) \mid \boldsymbol{u}_t, \boldsymbol{v}_t] = \partial_{\boldsymbol{v}} G(\boldsymbol{u}_t, \boldsymbol{v}_t),$$

and $\widetilde{g}(\boldsymbol{w}_t) = [-\partial_{\boldsymbol{u}}\widetilde{G}(\boldsymbol{u}_t, \boldsymbol{v}_t, i_t); \partial_{\boldsymbol{v}}\widetilde{G}(\boldsymbol{u}_t, \boldsymbol{v}_t, j_t)]$ is the targeted unbiased estimation of $g(\boldsymbol{w}_t)$.

Putting the above all together, we come up with the stochastic accelerated mirror prox method for MI-ODM. Algorithm 1 summarizes the pseudo-code.

4.3 Discussion

Resorting to the monotone stochastic variational inequality technique [Juditsky *et al.*, 2011; Chen *et al.*, 2017], we can prove the time complexity is $O(1/\sqrt{\epsilon} + 1/\epsilon + 1/\epsilon^2)$, where the dominated term $1/\epsilon^2$ comes from the noise introduced by the stochastic optimization. On the other hand, the state-of-the-art cutting-plane based algorithms have the same time complexity $O(1/\epsilon^2)$ [Zhao *et al.*, 2008]. Further note that in each iteration, cutting-plane based algorithms need to find the most violated key instance assignment and retrain a SVM model, while our method just preforms some random samplings and two steps of gradient descent followed by a projection with closed-form solutions, therefore our method is much more efficient.

5 Empirical Studies

We empirically study our method on CBIR image data sets in Sec. 5.1 and benchmark data sets in Sec. 5.2, respectively.

5.1 Experiments on Image Data

The data set contains 500 images (bags) with resolution 160×160 from five categories: castle, firework, mountain, sunset and waterfall. Each instance is a region in the image with resolution 20×20. Some of the regions are manually labeled as key instances. Table 1 summarizes the statistics of these data sets.

We randomly sample 50 images from each category as training data and the rest are used as test data. The training/test split are repeated for 10 times. The average accuracies as well as the standard deviations are recorded.

The proposed method is compared to the following five large margin based methods: 1) Ins-KI-SVM [Li *et al.*, 2009]; 2) Bag-KI-SVM [Li *et al.*, 2009]; 3) MI-SVM [Andrews *et al.*, 2003]; 4) mi-SVM [Andrews *et al.*, 2003]; 5) MI-Kernel [Gärtner *et al.*, 2002]. The parameters C_1, C_2, λ_1, λ_2 are selected from $\{1, 10, 100, 1000\}$, and ν, θ are selected from $\{0.2, 0.4, 0.6, 0.8\}$. The RBF kernel is applied for all the methods and the width is selected from the set of $\{2^{-4}\delta, 2^{-2}\delta, 2^{0}\delta, 2^{2}\delta, 2^{4}\delta\}$, where δ is the reciprocal of di-

Algorithm 1 MI-ODM

Input: ODM parameters λ_1, λ_2, ν, θ, maximum iteration number T, stopping criteria ζ.
Initialize $\boldsymbol{u}_0 \leftarrow \boldsymbol{0}$, $\boldsymbol{v}_0 \leftarrow [1/|\mathcal{C}|, \ldots, 1/|\mathcal{C}|]$, $t \leftarrow 0$, $[\overline{\boldsymbol{u}}_0; \overline{\boldsymbol{v}}_0] \leftarrow [\boldsymbol{u}_0; \boldsymbol{v}_0]$, $[\hat{\boldsymbol{u}}_0; \hat{\boldsymbol{v}}_0] \leftarrow [\boldsymbol{u}_0; \boldsymbol{v}_0]$.
while $t < T$ **do**
  $\gamma_t \leftarrow 2/(t+1)$.
  $[\underline{\boldsymbol{u}}_t; \underline{\boldsymbol{v}}_t] \leftarrow (1-\gamma_t)[\overline{\boldsymbol{u}}_t; \overline{\boldsymbol{v}}_t] + \gamma_t[\boldsymbol{u}_t; \boldsymbol{v}_t]$.
  Select i_t from $\{1, 2, \ldots, |\mathcal{C}|\}$ according to $\underline{\boldsymbol{v}}_t$.
  $\partial_{\boldsymbol{u}}\widetilde{G} \leftarrow \partial_{\boldsymbol{u}} D(\underline{\boldsymbol{u}}_t, \boldsymbol{c}_{i_t})$.
  Uniformly select j_t from $\{1, 2, \ldots, |\mathcal{C}|\}$.
  $\partial_{\boldsymbol{v}}\widetilde{G} \leftarrow [0, \ldots, |\mathcal{C}| D(\underline{\boldsymbol{u}}_t, \boldsymbol{c}_{j_t}) \ldots, 0]$.
  $[\hat{\boldsymbol{u}}_t; \hat{\boldsymbol{v}}_t] \leftarrow [\boldsymbol{u}_t + \eta\partial_{\boldsymbol{u}}\widetilde{G}/a; \boldsymbol{v}_t \exp(-\eta\partial_{\boldsymbol{v}}\widetilde{G}/b)]$.
  $[\tilde{\boldsymbol{u}}_{t+1}; \tilde{\boldsymbol{v}}_{t+1}] \leftarrow [\max\{\hat{\boldsymbol{u}}_t, \boldsymbol{0}\}; \hat{\boldsymbol{v}}_t/\|\hat{\boldsymbol{v}}_t\|_1]$.
  Select i'_t from $\{1, 2, \ldots, |\mathcal{C}|\}$ according to $\tilde{\boldsymbol{v}}_{t+1}$.
  $\partial_{\boldsymbol{u}}\widetilde{G} \leftarrow \partial_{\boldsymbol{u}} D(\tilde{\boldsymbol{u}}_{t+1}, \boldsymbol{c}_{i'_t})$.
  Uniformly select j'_t from $\{1, 2, \ldots, |\mathcal{C}|\}$.
  $\partial_{\boldsymbol{v}}\widetilde{G} \leftarrow [0, \ldots, |\mathcal{C}| D(\tilde{\boldsymbol{u}}_{t+1}, \boldsymbol{c}_{j'_t}) \ldots, 0]$.
  $[\hat{\boldsymbol{u}}_{t+1}; \hat{\boldsymbol{v}}_{t+1}] \leftarrow [\boldsymbol{u}_t + \eta\partial_{\boldsymbol{u}}\widetilde{G}/a; \boldsymbol{v}_t \exp(-\eta\partial_{\boldsymbol{v}}\widetilde{G}/b)]$.
  $[\boldsymbol{u}_{t+1}; \boldsymbol{v}_{t+1}] \leftarrow [\max\{\hat{\boldsymbol{u}}_{t+1}, \boldsymbol{0}\}; \hat{\boldsymbol{v}}_{t+1}/\|\hat{\boldsymbol{v}}_{t+1}\|_1]$.
  $[\overline{\boldsymbol{u}}_{t+1}; \overline{\boldsymbol{v}}_{t+1}] \leftarrow (1-\gamma_t)[\overline{\boldsymbol{u}}_t; \overline{\boldsymbol{v}}_t] + \gamma_t[\boldsymbol{u}_{t+1}; \boldsymbol{v}_{t+1}]$.
  $t \leftarrow t + 1$.
  if duality gap smaller than the stopping criteria ζ **then**
    Break.
  end if
end while
Output: $\boldsymbol{u}$, $\boldsymbol{v}$.

categories	#images	#key-instance per image
castle	100	19.39
firework	100	27.23
mountain	100	24.93
sunset	100	2.32
waterfall	100	13.89

Table 1: Characteristics of experimental data sets.

	methods	castle	firework	mountain	sunset	waterfall
margin based	Ins-KI-SVM	64.74±6.64	83.70±15.43	76.78±5.46	66.85±6.03	63.41±10.56
	Bag-KI-SVM	60.63±7.53	54.00±22.13	72.70±7.66	47.78±13.25	45.04±21.53
	MI-SVM	56.63±5.06	58.04±20.31	67.63±8.43	33.30±2.67	33.30±8.98
	mi-SVM	51.44±4.93	40.74±4.24	67.37±4.48	32.19±1.66	22.04±4.97
	MI-Kernel	50.52±4.46	36.37±7.92	65.67±5.18	32.15±1.67	19.93±4.65
	MI-ODM	**76.80±4.99**	**84.12±7.42**	77.05±7.94	**67.15±2.48**	**65.91±7.15**
non-margin based	DD	35.89±15.23	38.67±30.67	68.11±7.54	57.00±18.40	37.78±29.61
	EM-DD	76.00±4.63	79.89±19.25	**77.22±13.29**	53.56±16.81	44.33±15.13
	C*k*NN-ROI	51.48±4.59	43.63±12.40	60.59±4.38	34.59±2.57	30.48±6.34

Table 2: Success rate (%) on identifying the key instances. The best performance on each data set is bolded.

	methods	Musk1	Musk2	Elephant	Fox	Tiger
margin based	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 11	84.3	60.3	84.2
	MI-ODM	**88.2**	**89.8**	**84.5**	**63.9**	**85.2**
non-margin based	DD	88.0	84.0	N/A	N/A	N/A
	EM-DD	84.8	84.9	78.3	56.1	72.1

Table 3: Accuracy (%) on the benchmark data sets. The best performance on each data set is bolded. DD could not return results on some data sets in 48 hours.

mension. Three classical non-large margin based methods, i.e., DD [Maron and Ratan, 1998], EM-DD [Zhang and Goldman, 2001] and C*k*NN-ROI [Zhou *et al.*, 2005], which can locate key instances, are also included as baselines. All the parameters are selected by 5-fold cross validation.

Following the setting in [Li *et al.*, 2009], we evaluate the success rate, i.e., the ratio of the number of successes divided by the total number of relevant images. Table 2 shows the results of al the compared methods. As can be seen, MI-ODM beats all the large margin based methods. As for the non-margin based methods, MI-ODM is also better than DD, C*k*NN-ROI, and highly comparable to EM-DD. Specifically, MI-ODM achieves the best performance on four categories, while EM-DD achieves the best performance on the remaining one (mountain).

5.2 Experiments on Benchmark Data

We have also evaluated our method on five benchmark data sets commonly used in the literature of MIL, i.e., Musk1, Musk2, Elephant, Fox and Tiger. Musk1 has 47 positive bags and 45 negative bags. Musk2 consists of 39 positive bags and 63 negative bags. The remaining three data sets all contains 100 positive bags and 100 negative bags. The detail of these data sets can be found in [Dietterich *et al.*, 1997; Andrews *et al.*, 2003].

The setting is the same with the previous experiment. We adopt the 5-fold cross validation to measure the performance. DD could not return results on some data sets in 48 hours because it lacks high efficient packages. As shown in Table 3, our method is always better or comparable, almost never worse than other baselines.

6 Conclusions

Recent studies on margin theory discloses the importance of margin distribution to generalization ability, which gives rise to a promising research direction, i.e., the optimal margin distribution learning. Based on this observation, we propose the MI-ODM, which can identify the key instance via explicitly optimizing the margin distribution. Extensive experimental results verify the superiority of the new learning paradigm. In the future, we will apply the variance reduction technique [Johnson and Zhang, 2013] to further accelerate our method and extend it to other learning settings, e.g., multi-instance multi-label learning [Zhou *et al.*, 2012].

References

[Amores, 2013] Jaume Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201(4):81–105, 2013.

[Andrews *et al.*, 2003] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pages 561–568, Cambridge, MA, 2003.

[Beck and Teboulle, 2003] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[Bi *et al.*, 2005] Jinbo Bi, Yixin Chen, and James Z. Wang. A sparse support vector machine approach to region-based image categorization. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1121–1128, San Diego, CA, 2005.

[Chen *et al.*, 2017] Yunmei Chen, Guanghui Lan, and Yuyuan Ouyang. Accelerated schemes for a class of variational inequalities. *Mathematical Programming*, 165(1):113–149, 2017.

[Cristianini and Shawe-Taylor, 2000] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Cambridge, UK, 2000.

[Dietterich *et al.*, 1997] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1–2):31–71, 1997.

[Foulds and Frank, 2010] James Richard Foulds and Eibe Frank. A review of multi-instance learning assumptions. *Knowledge Engineering Review*, 25(1):1–25, 2010.

[Gao and Zhou, 2013] Wei Gao and Zhi-Hua Zhou. On the doubt about margin explanation of boosting. *Artificial Intelligence*, 203:1–18, 2013.

[Gärtner *et al.*, 2002] Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alexander J. Smola. Multi-instance kernels. In *Proceedings of the 19th International Conference on Machine Learning*, pages 179–186, Sydney, Australia, 2002.

[Herrera *et al.*, 2016] Francisco Herrera, Sebastián Ventura, Rafael Bello, Chris Cornelis, Amelia Zafra, Dánel Sánchez-Tarragó, and Sarah Vluymans. *Multiple Instance Learning: Foundations and Algorithms*. Springer, Cham, Switzerland, 2016.

[Johnson and Zhang, 2013] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, Lake Tahoe, NV, 2013.

[Juditsky *et al.*, 2011] Anatoli Juditsky, Arkadii Nemirovski, and Claire Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 1(1):17–58, 2011.

[Li *et al.*, 2009] Yu-Feng Li, James T. Kwok, Ivor W. Tsang, and Zhi-Hua Zhou. A convex method for locating regions of interest with multi-instance learning. In *Proceedings of the 20th European Conference on Machine Learning*, pages 15–30, Bled, Slovenia, 2009.

[Maron and Ratan, 1998] Oded Maron and Aparna Lakshmi Ratan. Multiple-instance learning for natural scene classification. In *Proceedings of the 15th International Conference on Machine Learning*, pages 341–349, Madison, WI, 1998.

[Nemirovski, 2005] Arkadi Nemirovski. Prox-method with rate of convergence O(1/T) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2005.

[Nesterov, 2003] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, New York, NY, 2003.

[Sion, 1958] Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.

[Tan *et al.*, 2020] Zhi-Hao Tan, Peng Tan, Yuan Jiang, and Zhi-Hua Zhou. Multi-label optimal margin distribution machine. *Machine Learning*, 109(3):623–642, 2020.

[Xu *et al.*, 2004] Linli Xu, James Neufeld, Bryce Larson, and Dale Schuurmans. Maximum margin clustering. In *Advances in Neural Information Processing Systems*, pages 1537–1544, Vancouver, Canada, 2004.

[Zhang and Goldman, 2001] Qi Zhang and Sally A. Goldman. EM-DD: An improved multiple-instance learning technique. In *Advances in Neural Information Processing Systems*, pages 1073–1080, Cambridge, MA, 2001.

[Zhang and Zhou, 2018a] Teng Zhang and Zhi-Hua Zhou. Optimal margin distribution clustering. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 4474–4481, New Orleans, LA, 2018.

[Zhang and Zhou, 2018b] Teng Zhang and Zhi-Hua Zhou. Semi-supervised optimal margin distribution machines. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3104–3110, Stockholm, Sweden, 2018.

[Zhang and Zhou, 2019] Teng Zhang and Zhi-Hua Zhou. Optimal margin distribution machine. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1143–1156, 2019.

[Zhao *et al.*, 2008] Bin Zhao, Fei Wang, and Changshui Zhang. Efficient maximum margin clustering via cutting plane algorithm. In *Proceedings of the SIAM International Conference on Data Mining*, pages 751–762, Atlanta, GA, 2008.

[Zhou *et al.*, 2005] Zhi-Hua Zhou, Xiao-Bing Xue, and Yuan Jiang. Locating regions of interest in cbir with multi-instance learning techniques. In *Proceedings of the 18th Australian Joint Conference on Advances in Artificial Intelligence*, pages 92–101, Sydney, Australia, 2005.

[Zhou *et al.*, 2012] Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291–2320, 2012.

[Zhou, 2014] Zhi-Hua Zhou. Large margin distribution learning. In *Proceedings of the 6th IAPR International Workshop on Artificial Neural Networks in Pattern Recognition*, pages 1–11, Montreal, Canada, 2014.