Machine Learning (2020) 109:623–642
https://doi.org/10.1007/s10994-019-05837-8

Multi-label optimal margin distribution machine

Zhi-Hao Tan[1] · Peng Tan[1] · Yuan Jiang[1] · Zhi-Hua Zhou[1]

Received: 6 May 2019 / Revised: 30 July 2019 / Accepted: 6 September 2019 / Published online: 10 October 2019

Abstract
Multi-label support vector machine (Rank-SVM) is a classic and effective algorithm for multi-label classification. The pivotal idea is to maximize the minimum margin of label pairs, which is extended from SVM. However, recent studies disclosed that maximizing the minimum margin does not necessarily lead to better generalization performance, and instead, it is more crucial to optimize the margin distribution. Inspired by this idea, in this paper, we first introduce margin distribution to multi-label learning and propose multi-label Optimal margin Distribution Machine (mlODM), which optimizes the margin mean and variance of all label pairs efficiently. Extensive experiments in multiple multi-label evaluation metrics illustrate that mlODM outperforms SVM-style multi-label methods. Moreover, empirical study presents the best margin distribution and verifies the fast convergence of our method.

Keywords Optimal margin distribution machine · Multi-label learning · Support vector machine · Margin theory

1 Introduction

In contrast to traditional supervised learning, multi-label classification purports to build classification models for objects assigned with multiple labels simultaneously, which is a common learning paradigm in real-world tasks. In the past decades, it has attracted much attention (Zhang and Zhou 2014a). To name a few, in image classification, a scene image is usually annotated with several tags (Boutell et al. 2004); in text categorization, a docu-

Editors: Kee-Eung Kim and Jun Zhu.

✉ Yuan Jiang
jiangy@lamda.nju.edu.cn

Zhi-Hao Tan
tanzh@lamda.nju.edu.cn

Peng Tan
tanp@lamda.nju.edu.cn

Zhi-Hua Zhou
zhouzh@lamda.nju.edu.cn

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

ment may present multiple topics (McCallum 1999; Schapire and Singer 2000); in music information retrieval, a piece of music can convey various messages (Turnbull et al. 2008).

To solve the multi-label tasks, a variety of methods have been proposed (Zhang and Zhou 2014a, Zhang et al. 2018), among which Rank-SVM (Elisseeff and Weston 2002) is one of the most eminent methods. It extended the idea of *maximizing minimum margin* in support vector machine (SVM) (Cortes and Vapnik 1995) to multi-label classification and achieved impressive performance. Specifically, the central idea of SVM is to search a large margin separator, i.e., maximizing the smallest distance from the instances to the classification boundary in a RKHS (reproducing kernel Hilbert space). Rank-SVM modified the definition of margin for label pairs and adapted maximizing margin strategy to deal with multi-label data, where a set of classifiers are optimized simultaneously. Benefiting from kernel tricks and considering pairwise relations between labels, Rank-SVM could handle non-linear classification problems and achieve good generalization performance.

For maximizing minimum margin strategy of SVMs, the margin theory (Vapnik 1995) provided good support to the generalization performance. It is noteworthy that there is also a long history of utilizing margin theory to explain the good generalization of AdaBoost (Freund and Schapire 1997), due to its tending to be empirically resistant to over-fitting. Specifically, Schapire et al. (1998) first suggested margin theory to interpret the phenomenon that AdaBoost seems resistant to over-fitting; soon after, Breiman (1999) developed a boosting-style algorithm, named Arc-gv, which is able to maximize the minimum margin but with a poor generalization performance. Later, Reyzin and Schapire (2006) observed that although Arc-gv produced a larger minimum margin, its margin distribution is quite poor.

Recently, the margin theory for Boosting has finally been defended (Gao and Zhou 2013), and has disclosed that the *margin distribution* rather than a single margin is more crucial to the generalization performance. It suggests that there may still exist large space to further ameliorate for SVMs. Inspired by this finding, Zhang and Zhou (2014b, 2019) proposed a binary classification method to optimize margin distribution by characterizing it through the first- and second-order statistics, which achieves better experimental results than SVMs. Later, Zhang and Zhou (2017, 2019) extended the definition of margin for multi-class classification and proposed multi-class optimal margin distribution machine (mcODM), which always outperforms multi-class SVMs empirically. In addition to classic supervised learning tasks, there is also a series of work in various tasks verifying the better generalization performance of optimizing margin distribution. For example, Zhou and Zhou (2016) extended the idea to exploit unlabeled data and handle unequal misclassification cost; Zhang and Zhou (2018) proposed the margin distribution machine for clustering. Tan et al. (2019) accelerated the kernel methods and applied the idea to large-scale datasets.

Existing work has depicted that optimizing the margin distribution can obtain superior generalization performance in most cases, but it still remains open for multi-label classification because the margin distribution for multi-label classification is much more complicated and the tremendous number of variables makes the optimization more difficult. In this paper, we propose a method to first introduce the margin distribution to multi-label classification, named multi-label optimal margin distribution machine (mlODM). Specifically, we formulate the idea of optimizing the margin distribution in multi-label learning and solve it efficiently by dual block coordinate descent. Extensive experiments in multiple multi-label evaluation metrics illustrate that our method mlODM outperforms SVM-style multi-label methods. Moreover, empirical studies present the best margin distribution and verifies the fast convergence of our method.

The rest of paper is organized as follows. Some preliminaries are introduced in Sect. 2. In Sect. 3, we review Rank-SVM and reformulate it with our definition to display the key idea of maximizing the minimum margin more clearly. Section 4 presents the formulation of our proposed method mlODM. In Sect. 5, we use Block Coordinate Descent Algorithm to solve the dual of objective. Section 6 reports our experimental studies and empirical observations. The related work is introduced in Sect. 7. Finally, Sect. 8 concludes with future work.

2 Preliminaries

Suppose $\mathcal{X} = \mathbb{R}^d$ denotes the d-dimensional instance space, and $\mathcal{Y} = \left\{y_1, y_2, \ldots, y_q\right\}$ denotes the label space with q possible class labels. The task of multi-label learning is to learn a classifier $h : \mathcal{X} \to 2^{\mathcal{Y}}$ from the multi-label training set $\mathcal{S} = \{(\boldsymbol{x}_i, Y_i)\,|1 \leq i \leq m\}$. In most cases, instead of outputting a multi-label classifier, the learning system will produce a real-valued function of the form $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. For each multi-label example $(\boldsymbol{x}_i, Y_i)$, $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector $(x_{i1}, x_{i2}, \ldots, x_{id})^\top$ and $Y_i \subset \mathcal{Y}$ is the set of labels associated with $\boldsymbol{x}_i$. Besides, the complement of Y_i, i.e., $\bar{Y}_i = \mathcal{Y} \backslash Y_i$, is referred to as a set of irrelevant labels of $\boldsymbol{x}_i$.

Let $\phi : \mathcal{X} \mapsto \mathbb{H}$ be a feature mapping associated to some positive definite kernel κ. For multi-label classification setting, the hypothesis $\mathcal{W} = \left\{\boldsymbol{w}_j \mid 1 \leq j \leq q\right\}$ is defined based on q weight vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_q \in \mathbb{H}$, where each vector $\boldsymbol{w}_y$, $y \in \mathcal{Y}$ define a scoring function $x \mapsto \boldsymbol{w}_y^\top \phi(\boldsymbol{x})$ and the label of instance $\boldsymbol{x}$ is the ones resulting in large score. For systems that rank the value of $\boldsymbol{w}_y^\top \phi(\boldsymbol{x})$, the decision boundaries of $\boldsymbol{x}$ are defined by the hyperplanes $\boldsymbol{w}_k^\top \phi(\boldsymbol{x}) - \boldsymbol{w}_l^\top \phi(\boldsymbol{x}) = 0$ for each relevant–irrelevant label pair $(k, l) \in Y \times \bar{Y}$. Therefore, the *margin* of a labeled instance $(\boldsymbol{x}_i, Y_i)$ can be defined as:

$$\gamma_h(\boldsymbol{x}_i, y_k, y_l) = \boldsymbol{w}_k^\top \phi(\boldsymbol{x}_i) - \boldsymbol{w}_l^\top \phi(\boldsymbol{x}_i), \quad \forall (k, l) \in Y_i \times \bar{Y}_i, \tag{1}$$

which is the difference in the score of $\boldsymbol{x}_i$ on a label pair. In addition,we define the *ranking margin* as:

$$\min_{(k,l) \in Y_i \times \bar{Y}_i} \frac{1}{\|\boldsymbol{w}_k - \boldsymbol{w}_l\|_{\mathbb{H}}} \gamma_h(\boldsymbol{x}_i, y_k, y_l), \tag{2}$$

which is the normalized margin, also the minimum signed distance of $\boldsymbol{x}_i$ to the decision boundary using norm $\|\cdot\|_{\mathbb{H}}$. Thus the classifier h misclassifies $(\boldsymbol{x}_i, Y_i)$ if and only if it produces a negative margin for this instance, i.e., there exists at least a label pair $(k, l) \in Y_i \times \bar{Y}_i$ in the output such that $\gamma_{k,l}(\boldsymbol{x}, y_k, y_l) < 0$.

Based on the above definition of margin, the task of multi-label learning is tackled by considering pairwise relations between labels, which corresponds to the ranking between relevant label and irrelevant label. Therefore, the methods based on the ranking margin belong to *second-order* strategies (Zhang and Zhou 2014a), which could achieve better generalization performance than *first-order* approaches.

3 Review of Rank-SVM

Using the ranking margin Eq. (2), Elisseeff and Weston (2002) first extended the key idea of maximizing margin to multi-label classification and proposed Rank-SVM, which learns q base models to minimize the Ranking Loss while maximizing the ranking margin. The brief derivation process is reformulated as follows.

When the learning system is capable of properly ranking every relevant–irrelevant label pair for each training example, the learning system's margin on the whole training set S naturally follows

$$\min_{(\boldsymbol{x}_i, Y_i)\in S} \min_{(k,l)\in Y_i\times\bar{Y}_i} \frac{1}{\|\boldsymbol{w}_k - \boldsymbol{w}_l\|_{\mathbb{H}}} \gamma_h(\boldsymbol{x}_i, y_k, y_l) \tag{3}$$

In this ideal case, we can normalize the parameters to ensure that for $\forall\, (\boldsymbol{x}_i, Y_i) \in S$,

$$\gamma_h(\boldsymbol{x}_i, y_k, y_l) = \boldsymbol{w}_k^\top \phi(\boldsymbol{x}_i) - \boldsymbol{w}_l^\top \phi(\boldsymbol{x}_i) \geq 1 \tag{4}$$

and there exist instances satisfying the equation. Thereafter, the problem of maximizing the ranking margin in Eq. (3) can be expressed as:

$$\begin{aligned} &\max_{\mathcal{W}} \min_{(\boldsymbol{x}_i, Y_i)\in S} \min_{(k,l)\in Y_i\times\bar{Y}_i} \frac{1}{\|\boldsymbol{w}_k - \boldsymbol{w}_l\|_{\mathbb{H}}^2} \\ &\text{s.t. } \gamma_h(\boldsymbol{x}_i, y_k, y_l) \geq 1, \quad \forall (k,l) \in Y_i \times \bar{Y}_i,\ i = 1, \ldots, m. \end{aligned} \tag{5}$$

Suppose we have sufficient training examples such that two labels are always co-occurring, the objective in Eq. (5) becomes equivalent to $\max_{\mathcal{W}}\ \min_{k,l} \frac{1}{\|\boldsymbol{w}_k - \boldsymbol{w}_l\|_{\mathbb{H}}^2}$, and the optimization problem can be reformulated as:

$$\begin{aligned} &\min_{\mathcal{W}} \max_{k,l} \|\boldsymbol{w}_k - \boldsymbol{w}_l\|_{\mathbb{H}}^2 \\ &\text{s.t. } \gamma_h(\boldsymbol{x}_i, y_k, y_l) \geq 1, \quad \forall (k,l) \in Y_i \times \bar{Y}_i,\ i = 1, \ldots, m. \end{aligned} \tag{6}$$

To avoid the difficulty brought by the max operator, Rank-SVM chooses to approximate the maximum with the sum operator and obtains $\min_{\mathcal{W}} \sum_{k,l=1}^{q} \|\boldsymbol{w}_k - \boldsymbol{w}_l\|_{\mathbb{H}}^2$. Note that a shift in the optimization variables does not change the ranking, the constraint $\sum_{j=1}^{q} \boldsymbol{w}_j = 0$ is added. The previous problem Eq. (6) is equivalent to:

$$\begin{aligned} &\min_{\mathcal{W}} \sum_{k=1}^{q} \|\boldsymbol{w}_k\|_{\mathbb{H}}^2 \\ &\text{s.t. } \gamma_h(\boldsymbol{x}_i, y_k, y_l) \geq 1, \quad \forall (k,l) \in Y_i \times \bar{Y}_i,\ i = 1, \ldots, m. \end{aligned} \tag{7}$$

To generalize the method to real-world scenarios where constraints in Eq. (7) can not be fully satisfied, Rank-SVM introduces *slack variables* like binary SVM, and obtain the final optimization problem:

$$\begin{aligned} &\min_{\mathcal{W};\Xi} \sum_{k=1}^{q} \|\boldsymbol{w}_k\|_{\mathbb{H}}^2 + C \sum_{i=1}^{m} \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(k,l)\in Y_i\times\bar{Y}_i} \xi_{ikl} \\ &\text{s.t. } \gamma_h(\boldsymbol{x}_i, y_k, y_l) \geq 1 - \xi_{ikl}, \\ &\quad \xi_{ikl} \geq 0, \quad \forall (k,l) \in Y_i \times \bar{Y}_i,\ i = 1, \ldots, m \end{aligned} \tag{8}$$

where $\Xi = \left\{\xi_{ikl} \mid 1 \leq i \leq m, (k,l) \in Y_i \times \bar{Y}_i\right\}$ is the set of slack variables. In this way, Rank-SVM aims to minimize the margin while minimizing the Ranking Loss. Specifically, the first part in Eq. (8) corresponds to the ranking margin while the second part corresponds to the surrogate Ranking Loss in hinge form. These two parts are balanced by the trade-off parameter C.

4 Formulation of proposed mlODM

Gao and Zhou (2013) proved that, to characterize the margin distribution, it is important to consider both the margin mean and the margin variance. Inspired by this idea, Zhang and Zhou (2019) proposed optimal margin distribution machine (ODM) for binary classification, which maximizes the margin mean while minimizing the margin variance. In this section, we introduce optimizing the margin distribution into multi-label setting and propose the formulation of multi-label optimal margin distribution machine (mlODM), the key idea of which is to maximize the ranking margin mean and minimize the margin variance.

Like binary ODM, considering that all the data in the training set S can be well ranked, we can normalize the weight vectors $\boldsymbol{w}_j, j = 1, \ldots, q$ such that for every label pair $(k, l) \in Y_i \times \bar{Y}_i$, the mean of $\gamma_h(\boldsymbol{x}_i, y_k, y_l)$ is 1, i.e., $\bar{\gamma}_h(\boldsymbol{x}, y_k, y_l) = 1$. Therefore, the distance of the mean point for label pair $(k, l) \in Y_i \times \bar{Y}_i$ to the decision boundary using norm $\|\cdot\|_{\mathbb{H}}$ can be represented as $\frac{1}{\|\boldsymbol{w}_k - \boldsymbol{w}_l\|_{\mathbb{H}}}$. Thereafter, the minimum distance between mean points and decision boundaries in this case can be represented as:

$$\begin{aligned} &\min_{(k,l)\in Y_i\times\bar{Y}_i} \frac{1}{\|\boldsymbol{w}_k - \boldsymbol{w}_l\|_{\mathbb{H}}} \\ &\text{s.t. } \bar{\gamma}_h(\boldsymbol{x}, y_k, y_l) = 1, \quad \forall (k, l) \in Y_i \times \bar{Y}_i, \end{aligned} \tag{9}$$

which is the minimum margin mean. Corresponding to maximizing the minimum margin in Rank-SVM, we maximize the margin mean on the whole dataset and obtain

$$\begin{aligned} &\max_{\mathcal{W}} \min_{(\boldsymbol{x}_i,Y_i)\in S} \min_{(k,l)\in Y_i\times\bar{Y}_i} \frac{1}{\|\boldsymbol{w}_k - \boldsymbol{w}_l\|_{\mathbb{H}}^2} \\ &\text{s.t. } \bar{\gamma}_h(\boldsymbol{x}, y_k, y_l) = 1, \quad \forall (k, l) \in Y_i \times \bar{Y}_i. \end{aligned} \tag{10}$$

Then we use the same technique to simplify the objective. Specifically, we suppose the problem is not ill-conditioned, approximate the maximum operator with the sum operator and add the constraint $\sum_{j=1}^q \boldsymbol{w}_j = 0$. Thereafter, the objective of maximizing the margin mean can be reformulated as:

$$\begin{aligned} &\min_{\mathcal{W}} \sum_{k=1}^{q} \|\boldsymbol{w}_k\|_{\mathbb{H}}^2 \\ &\text{s.t. } \bar{\gamma}_h(\boldsymbol{x}, y_k, y_l) = 1, \quad \forall (k, l) \in Y_i \times \bar{Y}_i. \end{aligned} \tag{11}$$

After considering the margin mean, in order to optimize the margin distribution, we still need to minimize the margin variance. Like binary ODM, the variance can be formulated as slack variables. Considering the margin variance is calculated on every label pair, we use the framework of Ranking Loss to weighted average the variance. Then the objective can be represented as:

$$\begin{aligned} &\min_{\mathcal{W},\Xi,\Lambda} \sum_{k=1}^{q} \|\boldsymbol{w}_k\|_{\mathbb{H}}^2 + C \sum_{i=1}^{m} \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(k,l)\in Y_i\times\bar{Y}_i} \left(\xi_{ikl}^2 + \epsilon_{ikl}^2\right) \\ &\text{s.t. } \bar{\gamma}_h(\boldsymbol{x}, y_k, y_l) = 1, \\ &\quad \gamma_h(\boldsymbol{x}_i, y_k, y_l) \geq 1 - \xi_{ikl}, \\ &\quad \gamma_h(\boldsymbol{x}_i, y_k, y_l) \leq 1 + \epsilon_{ikl}, \quad \forall (k, l) \in Y_i \times \bar{Y}_i, \; i = 1, \ldots, m \end{aligned} \tag{12}$$

where C is the trade-off parameter to balance the margin mean and variance; $\varXi = \left\{\xi_{ikl} \mid 1 \leq i \leq m, (k,l) \in Y_i \times \bar{Y}_i\right\}$ and $\varLambda = \left\{\epsilon_{ikl} \mid 1 \leq i \leq m, (k,l) \in Y_i \times \bar{Y}_i\right\}$ are the set of slack variables. Because of setting margin mean as 1, the right part of the objective is the weighted average of margin variance. However, the above optimization problem is very difficult to solve due to the existence of the constraint of margin mean. Draw on the idea of insensitive margin loss in Support Vector Regression (Vapnik 1995) and in order to simplify the objective, we approximate the margin mean and variance by a θ-insensitive margin loss. The previous problem can be recast as:

$$\begin{aligned} &\min_{\mathcal{W},\varXi,\varLambda} \sum_{k=1}^{q} \|\boldsymbol{w}_k\|_{\mathbb{H}}^2 + C\sum_{i=1}^{m} \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(k,l)\in Y_i\times\bar{Y}_i} \left(\xi_{ikl}^2 + \epsilon_{ikl}^2\right) \\ &\text{s.t. } \gamma_h(\boldsymbol{x}_i, y_k, y_l) \geq 1 - \theta - \xi_{ikl}, \\ &\quad \gamma_h(\boldsymbol{x}_i, y_k, y_l) \leq 1 + \theta + \epsilon_{ikl}, \quad \forall (k,l) \in Y_i \times \bar{Y}_i,\ i = 1,\ldots,m \end{aligned} \tag{13}$$

where $\theta \in [0, 1]$ is a hyperparameter to control the degree of approximation. By the θ-insensitive margin loss, the margin mean is limited to the interval while the variance is only calculated by the outliers outside the interval. From another point of view, the first part of objective is the regularization term to limit the model complexity and minimize the structural risk; the second part is approximated weighted variance loss. Moreover, the parameter θ also control the number of support vector, i.e., the sparsity of solutions.

For each instance outside the interval, it is obvious that the instances corresponding to $\gamma_h(\boldsymbol{x}_i, y_k, y_l) < 1 - \theta$ are much easier to be misclassified than those falling on the other side. Thus like binary ODM, we set different weights for the loss of instances in different sides. This leads us to the final formulation of mlODM:

$$\begin{aligned} &\min_{\mathcal{W},\varXi,\varLambda} \sum_{k=1}^{q} \|\boldsymbol{w}_k\|_{\mathbb{H}}^2 + \sum_{i=1}^{m} \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(k,l)\in Y_i\times\bar{Y}_i} \left(\xi_{ikl}^2 + \mu\epsilon_{ikl}^2\right) \\ &\text{s.t. } \gamma_h(\boldsymbol{x}_i, y_k, y_l) \geq 1 - \theta - \xi_{ikl}, \\ &\quad \gamma_h(\boldsymbol{x}_i, y_k, y_l) \leq 1 + \theta + \epsilon_{ikl}, \quad \forall (k,l) \in Y_i \times \bar{Y}_i,\ i = 1,\ldots,m \end{aligned} \tag{14}$$

where $\mu \in (0, 1]$ is the weight parameter. The optimization problem of mlODM is more difficult than Rank-SVM because considering margin distribution is more complex than minimizing the hinge-form Ranking Loss.

5 Optimization

The mlODM problem Eq. (14) is a non-differentiable quadratic programming problem, we solve its dual form by Block Coordinate Descent (BCD) algorithm (Richtárik and Takáč 2014) in this paper. For the convenience of calculation, the origin problem can be represented as follows:

$$\begin{aligned} &\min_{\mathcal{W},\varXi,\varLambda} \frac{1}{2}\sum_{k=1}^{q} \|\boldsymbol{w}_k\|_{\mathbb{H}}^2 + \frac{C}{2}\sum_{i=1}^{m} \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(k,l)\in Y_i\times\bar{Y}_i} \left(\xi_{ikl}^2 + \mu\epsilon_{ikl}^2\right) \\ &\text{s.t. } \boldsymbol{w}_k^\top \phi(\boldsymbol{x}_i) - \boldsymbol{w}_l^\top \phi(\boldsymbol{x}_i) \geq 1 - \theta - \xi_{ikl}, \\ &\quad \boldsymbol{w}_k^\top \phi(\boldsymbol{x}_i) - \boldsymbol{w}_l^\top \phi(\boldsymbol{x}_i) \leq 1 + \theta + \epsilon_{ikl}, \quad \forall (k,l) \in Y_i \times \bar{Y}_i,\ i = 1,\ldots,m. \end{aligned} \tag{15}$$

5.1 Lagrangian dual problem

First we introduce the dual variables $\alpha_{ikl} \geq 0, (k,l) \in Y_i \times \bar{Y}_i$ related to the first $\sum_{i=1}^{m} |Y_i||\bar{Y}_i|$ constraints, and the variables $\beta_{ikl} \geq 0, (k,l) \in Y_i \times \bar{Y}_i$ related to the last $\sum_{i=1}^{m} |Y_i||\bar{Y}_i|$ constraints respectively. The Lagrangian function of Eq. (15) can be computed:

$$\begin{aligned}
&L(\boldsymbol{w}_k, \xi_{ikl}, \epsilon_{ikl}, \alpha_{ikl}, \beta_{ikl}) \\
&= \frac{1}{2}\sum_{k=1}^{q} \|\boldsymbol{w}_k\|_{\mathbb{H}}^2 + \frac{C}{2}\sum_{i=1}^{m} \frac{1}{|Y_i||\bar{Y}_i|} \sum_{(k,l)\in Y_i\times\bar{Y}_i} \left(\xi_{ikl}^2 + \mu\epsilon_{ikl}^2\right) \\
&\quad -\sum_{i=1}^{m} \sum_{(k,l)\in Y_i\times\bar{Y}_i} \alpha_{ikl}\left(\boldsymbol{w}_k^\top \phi(\boldsymbol{x}_i) - \boldsymbol{w}_l^\top \phi(\boldsymbol{x}_i) - 1 + \theta + \xi_{ikl}\right) \\
&\quad +\sum_{i=1}^{m} \sum_{(k,l)\in Y_i\times\bar{Y}_i} \beta_{ikl}\left(\boldsymbol{w}_k^\top \phi(\boldsymbol{x}_i) - \boldsymbol{w}_l^\top \phi(\boldsymbol{x}_i) - 1 - \theta - \epsilon_{ikl}\right)
\end{aligned}$$

By setting the partial derivations of variables $\boldsymbol{w}_k$ to zero, we can obtain:

$$\boldsymbol{w}_k = \sum_{i=1}^{m}\left(\sum_{(j,l)\in Y_i\times\bar{Y}_i} \left(\alpha_{ijl} - \beta_{ijl}\right)\cdot c_{ijl}^k\right)\phi(\boldsymbol{x}_i) \tag{16}$$

where c_{ijl}^k is defined as follows:

$$c_{ijl}^k = \begin{cases} 0 & j \neq k \text{ and } l \neq k \\ 1 & j = k \\ -1 & l = k. \end{cases}$$

Note that c_{ijl}^k depends on k. Then setting $\partial_{\xi_{ikl}} L = 0$ and $\partial_{\epsilon_{ikl}} L = 0$ at the optimum yields:

$$\xi_{ikl} = \frac{|Y_i||\bar{Y}_i|}{C}\alpha_{ikl}, \qquad \epsilon_{ikl} = \frac{|Y_i||\bar{Y}_i|}{\mu C}\beta_{ikl} \tag{17}$$

Substituting Eqs. (16) and (17) into Lagrange function, simplifying the problem by double counting and transforming the minimization to maximization, a the dual of Eq. (15) can then be expressed:

$$\begin{aligned}
\min_{\alpha_{ikl},\beta_{ikl}} \quad & \frac{1}{2}\sum_{k=1}^{q} \|\boldsymbol{w}_k\|_{\mathbb{H}}^2 + \sum_{i=1}^{m} \sum_{(k,l)\in Y_i\times\bar{Y}_i} [\alpha_{ikl}(\theta - 1) + \beta_{ikl}(\theta + 1)] \\
& + \sum_{i=1}^{m} \frac{|Y_i||\bar{Y}_i|}{2C} \sum_{(k,l)\in Y_i\times\bar{Y}_i} \left(\frac{1}{\mu}\beta_{ikl}^2 + \alpha_{ikl}^2\right) \\
\text{s.t.} \quad & \alpha_{ikl} \geq 0, \ \beta_{ikl} \geq 0, \quad \forall (k,l) \in Y_i \times \bar{Y}_i, \ i = 1, \ldots, m.
\end{aligned} \tag{18}$$

In order to use as simple notation as possible to make the objective concise, we express it with both dual variables and weight vectors $\boldsymbol{w}_k$. Now we transform the objective into block vector representation of dual variables for each instance. For the ith instance, we construct four column vectors $\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i, \boldsymbol{c}_i^k$ and $\boldsymbol{e}_i$, all of $|Y_i||\bar{Y}_i|$-dimension, which is the number of label pairs contained in Y_i. Specifically, for $1 \leq i \leq m$, the vectors are defined as follows:

$$\boldsymbol{\alpha}_i = \left[\alpha_{ikl} | (k,l) \in Y_i \times \bar{Y}_i\right]^\top,$$

$$\begin{aligned}\boldsymbol{\beta}_i &= \left[\beta_{ikl}|(k,l) \in Y_i \times \bar{Y}_i\right]^\top, \\ \boldsymbol{c}_i &= \left[c_{ijl}^k|(j,l) \in Y_i \times \bar{Y}_i\right]^\top, \\ \boldsymbol{e}_i &= [1,\ldots,1]^\top \end{aligned} \tag{19}$$

where $\boldsymbol{c}_i$ can only take the value $\{0, +1, -1\}$. Based on the above definition, the optimal solution of weight vectors can be rewritten as:

$$\boldsymbol{w}_k = \sum_{i=1}^m \left(\left(\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\right)^\top \boldsymbol{c}_i^k\right) \phi(\boldsymbol{x}_i) \tag{20}$$

Using Eqs. (19) and (20), the dual problem Eq. (18) can be finally represented as:

$$\begin{aligned}\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \ & \frac{1}{2}\sum_{k=1}^q \sum_{i=1}^m \sum_{h=1}^m \left[\left(\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\right)^\top \boldsymbol{c}_i^k\right]\left[\left(\boldsymbol{\alpha}_h - \boldsymbol{\beta}_h\right)^\top \boldsymbol{c}_h^k\right] \phi(\boldsymbol{x}_i)^\top \phi(\boldsymbol{x}_h) \\ & + \sum_{i=1}^m \frac{|Y_i||\bar{Y}_i|}{2C}\left(\frac{1}{\mu}\boldsymbol{\beta}_i^\top \boldsymbol{\beta}_i + \boldsymbol{\alpha}_i^\top \boldsymbol{\alpha}_i\right) + \sum_{i=1}^m \left[(\theta - 1)\boldsymbol{\alpha}_i^\top \boldsymbol{e}_i + (\theta + 1)\boldsymbol{\beta}_i^\top \boldsymbol{e}_i\right] \\ \text{s.t. } & \boldsymbol{\alpha}_i \geq 0,\ \boldsymbol{\beta}_i \geq 0,\ i = 1,\ldots,m. \end{aligned} \tag{21}$$

The optimization problem includes $2\sum_{i=1}^m |Y_i||\bar{Y}_i|$ variables, the order of which is $O\left(mq^2\right)$ in the worst case, so we need an efficient optimization method. Considering the variables can be partitioned into m disjoint sets, and the i-th set only involves $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$, so it's natural to use Block Coordinate Descent (BCD) method (Richtárik and Takáč 2014) to decompose the problem into m sub-problems.

We note that column vector $\boldsymbol{\zeta} = \left[\boldsymbol{\alpha}_1;\ldots;\boldsymbol{\alpha}_m;\boldsymbol{\beta}_1;\ldots;\boldsymbol{\beta}_m\right]$, and diagonal matrix I_i satisfies $I_i\boldsymbol{\zeta} = \boldsymbol{\alpha}_i,\ 1 \leq i \leq m$, and $I_i\boldsymbol{\zeta} = \boldsymbol{\beta}_i,\ m+1 \leq i \leq 2m$. Then the objective can be reformulated as

$$\min_{\boldsymbol{\zeta}} \ \boldsymbol{\zeta}^\top \boldsymbol{Q}\boldsymbol{\zeta} + \boldsymbol{\zeta}^\top \boldsymbol{u} + \Psi(\boldsymbol{\zeta}) \tag{22}$$

where $\boldsymbol{Q} = \sum_{k=1}^q \sum_{i,h=1}^m \left[(I_i - I_{i+m})\boldsymbol{c}_i^k\right]\left[(I_h - I_{h+m})\boldsymbol{c}_h^k\right] + \sum_{i=1}^m \frac{|Y_i||\bar{Y}_i|}{2C}\left(\frac{1}{\mu}I_{i+m}I_{i+m} + I_iI_i\right)$, and $\Psi(\boldsymbol{\zeta})$ equals to 0 when $\boldsymbol{\zeta} \geq 0$ and $+\infty$ otherwise. Notice that the first term of matrix $\boldsymbol{Q}$ is positive semi-definite and the second term is positive definite, it's easy to verify that the first term of optimization problem Eq. (22) is strongly convex and the problem satisfies the assumptions in Richtárik and Takáč (2014). Therefore, we can use Block Coordinate Descent algorithm to solve mlODM efficiently with linear convergence rate.

Algorithm 1 below shows the details of the optimization procedure of mlODM by BCD.

Algorithm 1 Dual Block Coordinate Descent for kernel mlODM

```
Input: training set $S$, hyperparameters $C, \theta, \mu$.
Initialize $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_m]$ and $\boldsymbol{\beta} = [\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_m]$ as zero vector.
while $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ not converge do
    randomly shuffle the training set $\{\pi(1),\ldots,\pi(m)\}$
    for $i = \pi(1)$ to $\pi(m)$ do
        solve the sub-problem 24 and obtain $\boldsymbol{\alpha}_i^{new}$, $\boldsymbol{\beta}_i^{new}$
        update $\boldsymbol{\alpha}_i$, $\boldsymbol{\beta}_i$
    end for
end while
Calculate the weight vectors $\boldsymbol{w}_k, k = 1,\ldots,q$ by 20
Output: $\boldsymbol{w}_k, k = 1,\ldots,q$.
```

Algorithm 2 Multiplicative Margin Maximization algorithm for sub-problem

```
Input: positive definite matrix $\boldsymbol{H} = \boldsymbol{H}_i$, row vector $\boldsymbol{v} = \boldsymbol{v}_i$.
Initialize $\boldsymbol{\eta} = \boldsymbol{\eta}_i^\top = [\eta_1, \ldots, \eta_{2m}]^\top$ as $[1, \ldots, 1]$.
Let
    $$H_{jk}^+ = \begin{cases} H_{jk} & \text{if } H_{jk} > 0, \\ 0 & \text{otherwise,} \end{cases} \text{ and } H_{jk}^- = \begin{cases} |H_{jk}| & \text{if } H_{jk} < 0, \\ 0 & \text{otherwise.} \end{cases}$$
while Fixed point does not occur do
    update each $\eta_j$ with
        $$\lambda_j \longleftarrow \frac{-v_j + \sqrt{v_j^2 + 4\left(\boldsymbol{H}^+\boldsymbol{\eta}\right)_j \left(\boldsymbol{H}^-\boldsymbol{\eta}\right)_j}}{2\left(\boldsymbol{H}^+\boldsymbol{\eta}\right)_j}$$
        $$\eta_j \longleftarrow \eta_j \cdot \lambda_j$$
end while
Output: $\boldsymbol{\eta}$.
```

5.2 Solving the sub-problem

For each sub-problem, we select $2|Y_i||\bar{Y}_i|$ variables $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ corresponding to a instance to minimize while keeping other variables constants, and repeat this procedure until convergence.

Note that all variables are fixed except $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$. After removing the constants, we obtain the sub-problem as follows:

$$\begin{aligned} \min_{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i} \quad & \frac{1}{2}\left(\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\right)^\top \boldsymbol{A}_i \left(\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\right) + M_i\left(\frac{1}{\mu}\boldsymbol{\beta}_i^\top\boldsymbol{\beta}_i + \boldsymbol{\alpha}_i^\top\boldsymbol{\alpha}_i\right) \\ & + \boldsymbol{b}_i\left(\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\right) + \theta\left(\boldsymbol{\alpha}_i + \boldsymbol{\beta}_i\right)^\top \boldsymbol{e}_i - \left(\boldsymbol{\alpha}_i - \boldsymbol{\beta}_i\right)^\top \boldsymbol{e}_i \\ \text{s.t.} \quad & \boldsymbol{\alpha}_i \geq \boldsymbol{0},\ \boldsymbol{\beta}_i \geq \boldsymbol{0} \end{aligned} \tag{23}$$

where $\boldsymbol{A}_i = \left(\sum_{k=1}^q \boldsymbol{c}_i^k \boldsymbol{c}_i^{k\top}\right)\kappa(\boldsymbol{x}_i, \boldsymbol{x}_i)$ is a matrix, $\boldsymbol{b}_i = \sum_{k=1}^q \sum_{j \neq i}\left(\boldsymbol{\alpha}_j - \boldsymbol{\beta}_j\right)^\top \boldsymbol{c}_j^k \boldsymbol{c}_i^{k\top}\kappa(\boldsymbol{x}_j, \boldsymbol{x}_i)$ is a row vector and $M_i = \frac{|Y_i||\bar{Y}_i|}{2C}$ is a constant. $\boldsymbol{\alpha}_i \geq \boldsymbol{0}$ represents that each element of $\boldsymbol{\alpha}_i$ is nonnegative, so as $\boldsymbol{\beta}_i$.

Let column vector $\boldsymbol{\eta}_i = \left[\boldsymbol{\alpha}_i; \boldsymbol{\beta}_i\right]$ for $i = 1, \ldots, m$ and $\boldsymbol{I}$ be an identity matrix with $|Y_i||\bar{Y}_i|$ dimension, let $\boldsymbol{I}_\mu$ be a diagonal matrix with $2|Y_i||\bar{Y}_i|$ dimension with the elements of the second half being $\frac{1}{\mu}$. the objective can be further represented as

$$\begin{aligned} \min_{\boldsymbol{\eta}_i} \quad & F(\boldsymbol{\eta}_i) \triangleq \frac{1}{2}\boldsymbol{\eta}_i^\top \boldsymbol{H}_i \boldsymbol{\eta}_i + \boldsymbol{v}_i \boldsymbol{\eta}_i \\ \text{s.t.} \quad & \boldsymbol{\eta}_i \geq \boldsymbol{0} \end{aligned} \tag{24}$$

where $\boldsymbol{H}_i = [\boldsymbol{I}, -\boldsymbol{I}]^\top \boldsymbol{A}_i [\boldsymbol{I}, -\boldsymbol{I}] + 2M_i \boldsymbol{I}_\mu$ is a matrix and $\boldsymbol{v}_i = \boldsymbol{b}_i[\boldsymbol{I}, -\boldsymbol{I}] + \theta \boldsymbol{e}_i^\top [\boldsymbol{I}, \boldsymbol{I}] - \boldsymbol{e}_i^\top [\boldsymbol{I}, -\boldsymbol{I}]$ is a row vector. It is easy to prove that the first part of $\boldsymbol{H}_i$ is positive semi-definite and the second part $2M_i\boldsymbol{I}_\mu$ is positive definite. Thus $\boldsymbol{H}_i$ is positive definite, and the problem is strictly convex.

Through the above derivation, the sub-problem is finally reformulated as a convex non-negative quadratic programming problem, which can be solved by QP solver efficiently. In order to avoid the drawback of having to choose a learning rate and control the precision, we

Table 1 Characteristics of datasets in our experiments

Dataset	#Train	#Test	#Feature	#Labels	LCard	LDen	Domain
Emotions	391	202	72	6	1.87	0.31	*Music*
Scene	1211	1196	294	6	1.07	0.18	*Image*
Yeast	1500	917	103	14	4.24	0.30	*Biology*
Birds	175	172	260	19	1.91	0.10	*Audio*
Genbase	463	199	1185	27	1.25	0.05	*Biology*
Medical	645	333	1449	45	1.25	0.03	*Text*
Enron	1123	579	1001	53	3.38	0.06	*Text*

choose Multiplicative Margin Maximization (M^3) method (Sha et al. 2002) to solve Eq. (24). Detailed algorithm is showed in Algorithm 2. It is worth mentioning that the M^3 algorithm achieve minimum value when the fixed point occurs, i.e., when one of two conditions holds for each element of optimization variables η_j: (1) $\eta_j^* > 0$ and $\lambda_j = 1$, or (2) $\eta_j^* = 0$. In experiments, each variable η_j should be initialized to 1, and the criterion of fixed points can be relaxed. In addition, we utilize a simple and effective heuristic shrinking strategy for further acceleration. Considering that $\nabla F(\boldsymbol{\eta}_i) = \mathbf{0}$ indicates that the corresponding block $\boldsymbol{\eta}_i$ has achieve optimum, we can move to next iteration without update $\boldsymbol{\eta}_i$ if this condition holds.

6 Empirical study

In this section, we empirically evaluate the effectiveness of our method on seven datasets. We first introduce the experimental settings in Sect. 6.1, which includes the information of datasets, the compared methods, the evaluation metrics used in experiments, the threshold calibration and the hyperparameters setting. In Sect. 6.2, we compare the performance in four metrics and verify the superiority of mlODM. We analyze the convergence of our method mlODM on six datasets in Sect. 6.3 and compare the margin distribution of each method by visualization in Sect. 6.4.

In general, we compare our method with three multi-label classification methods on seven classic datasets, and use four metrics to evaluate the performance of each method. Then we analyze the characteristics of our method empirically. The information of experiments is introduced below.

6.1 Experimental setup

These seven datasets include Emotions, Scene, Yeast, Birds, Genbase, Medical and Enron from MULAN (Tsoumakas et al. 2011b) multi-label learning library. These datasets cover five diverse domains: audio, music, image, biology and text. The information of all the datasets is detailed in Table 1. #Train, #Test and #Feature represent the number of training and test examples and number of features respectively. The number of labels is denoted by #Labels, and the label cardinality and density (%) by LCard and LDen respectively. All features are normalized into the interval [0, 1].

6.1.1 Compared methods

In experiments, we compare our proposed method mlODM with six multi-label algorithms as follows.

- *BP-MLL* (Zhang and Zhou 2006). BP-MLL is a neural network algorithm for multi-label classification, which employs a multi-label error function in Backpropagation algorithm.
- *Rank-SVM* (Elisseeff and Weston 2002). Rank-SVM is a famous and classic margin-based multi-label classification method, which aims to maximize the minimum margin of each label pair. The objective is optimized by Frank–Wolfe Algorithm (Frank and Wolfe 1956) with sub-problem being a Linear Programming problem.
- *ML-KNN* (Zhang and Zhou 2007). The basic idea of this method is to adapt k-nearest neighbor techniques to deal with multi-label data, where maximum a posteriori (MAP) rule is utilized to make prediction by reasoning with the labeling information embodied in the neighbors.
- *Rank-SVMz* (Xu 2012). By adding a zero label into Rank-SVM, Rank-SVMz has a special QP problem in which each class has an independent equality constraint, and does not need to learn the linear threshold function by regression.
- *Rank-CVM* (Xu 2013a). The key idea of Rank-CVM is to combine Rank-SVM with the binary core vector machine (CVM). The optimization is formulated as a QP problem with a unit simplex constraint like CVM.
- *Rank-LSVM* (Xu 2016). This method is proposed recently and generalizes binary Lagrangian support vector machine (LSVM) to multi-label classification, resulting into a strictly convex Quadratic Programming problem with non-negative constraints only.

The compared methods include three classic multi-label classification methods: BP-MLL, Rank-SVM and ML-KNN, which are coded in MATLAB from;[1] and three methods modified from Rank-SVM: Rank-SVMz, Rank-CVM and Rank-LSVM, all coded in C/C++ from package MLC-SVM.[2]

6.1.2 Evaluation metrics

In contrast to single-label classification, performance evaluation in multi-label classification is more complicated. A number of performance measures focusing on different aspects have been proposed (Schapire and Singer 2000; Tsoumakas et al. 2011a). Recently, Wu and Zhou (2017) provides a unified margin view of these measures, which suggests that it is more informative to evaluate using both measures optimized by label-wise effective predictors and measures optimized by instance-wise effective predictors. Inspired by this theoretical results, we select ranking loss, one-error and average precision for the first kind of measures and Hamming Loss for the second. We recall the definition of metrics as follows. The ↑ (↓) indicates that the larger (smaller) the value, the better the performance.

The ranking loss evaluates the fraction of reversely ordered label pairs, i.e., an irrelevant label is ranked higher than a relevant label.

$$rloss(\downarrow) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{|Y_i||\bar{Y}_i|}\left|\left\{(y_k, y_l) \mid f(\boldsymbol{x}_i, y_k) \geq f(\boldsymbol{x}_i, y_l),\ (y_k, y_l) \in Y_i \times \bar{Y}_i\right\}\right|$$

[1] http://cse.seu.edu.cn/people/zhangml/Resources.htm.

[2] http://computer.njnu.edu.cn/Lab/LABIC/LABIC_software.html.

The one-error evaluates the fraction of examples whose top-ranked label is not in the relevant label set.

$$one\text{-}error(\downarrow) = \frac{1}{m}\sum_{i=1}^{m}[\![[\text{argmax}_{y\in Y_i} f(\boldsymbol{x}_i, y)] \notin Y_i]\!]$$

where $[\![\cdot]\!]$ equals 1 if $\cdot$ is true and 0 otherwise.

The average precision evaluates the average fraction of relevant labels ranked higher than a particular label $y \in Y_i$.

$$averprec(\uparrow) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{|Y_i|}\sum_{y\in Y_i}\frac{\left|\left\{y' \mid rank_f(\boldsymbol{x}_i, y') \le rank_f(\boldsymbol{x}_i, y),\ y' \in Y_i\right\}\right|}{rank_f(\boldsymbol{x}_i, y)}$$

The Hamming loss evaluates how many times an instance-label pair is misclassified.

$$Hamming\ loss(\downarrow) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{q}\,|h(\boldsymbol{x}_i)\Delta Y_i|$$

where Δ stands for the symmetric difference between two sets. All methods will be evaluated in these four measures.

6.1.3 Settings of each method

For Rank-SVM, Rank-CVM, Rank-LSVM and our method mlODM, the threshold function is determined using linear regression technique in Elisseeff and Weston (2002). Specifically, we train a linear model to predict the set size. For this linear model, the learning system produces a q-dimensional vector $\left(f_1(\boldsymbol{x}_i), \ldots, f_q(\boldsymbol{x}_i)\right)$ as the training data, the target values are the optimal threshold values via minimizing the Hamming loss. Then a linear regression threshold function is trained as the label size predictor.

For Rank-SVM, Rank-CVM, Rank-SVMz, Rank-LSVM and mlODM, the RBF kernel will be used in all experiments. For the first four methods, the hyperparameters, i.e., the RBF kernel scale factor γ and the regularization parameter C, are optimally set as recommended in Xu (2012, 2016), which is tuned from $\left\{2^{-10}, 2^{-9}, \ldots, 2^{2}\right\}$ and $\left\{2^{-2}, 2^{-1}, \ \ldots, 2^{10}\right\}$ respectively. For our method mlODM, the C and γ are selected by 5-fold cross validation from the same range as Rank-SVM. In addition, the trade-off parameter μ and approximation parameter θ are selected from $\{0.1, 0.2, \ldots, 0.9\}$. For ML-KNN, the number of nearest neighbors is 10. For BP-MLL, as recommended, the learning rate is fixed at 0.05, the number of hidden neurons is set to be 20% of the number of input units, the number of training epochs is fixed to be 100 and the regularization constant is set to be 0.1. All randomized algorithms are repeated five times.

6.2 Results and discussion

Table 2 shows the results of ranking loss, Hamming loss, one-error and average precision respectively, where the best accuracy on each dataset in each metric is bolded. From the experimental results, our method mlODM outperforms other methods in all evaluation metrics on more than half of datasets, and obtains very competitive results on other datasets. Specifically, mlODM performs better than BP-MLL/ML-KNN/Rank-SVM/Rank-CVM/Rank-SVMz/Rank-LSVM on 24/28/19/20/25/18 over seven datasets in four metrics.

Table 2 Experimental results of seven methods on seven datasets in four measures

Loss	Dataset	BP-MLL	ML-KNN	Rank-SVM	Rank-CVM	Rank-SVMz	Rank-LSVM	mlODM
Ranking loss(↓)	*Emotions*	45.89	28.29	15.79	15.08	**14.64**	15.78	15.31
	Scene	39.16	9.31	13.70	7.30	7.38	6.81	**6.76**
	Yeast	17.47	17.15	**15.82**	15.97	16.60	15.97	15.83
	Birds	48.45	30.24	16.44	16.03	16.77	16.26	**15.42**
	Genbase	0.76	0.64	**0.12**	0.41	0.44	0.41	0.18
	Medical	5.23	5.85	2.48	2.69	2.96	2.65	**2.18**
	Enron	7.38	9.38	7.37	8.01	9.10	**7.10**	7.62
Hamming loss(↓)	*Emotions*	31.77	29.37	20.05	19.88	20.71	20.71	**19.50**
	Scene	29.17	9.89	14.56	**9.74**	10.66	9.84	9.78
	Yeast	20.84	19.80	**19.08**	19.62	19.29	19.27	**19.08**
	Birds	11.65	9.82	8.38	8.04	9.00	7.99	**7.80**
	Genbase	0.32	4.28	0.21	0.17	0.35	0.11	**0.06**
	Medical	2.66	1.87	1.50	1.35	1.35	**1.24**	1.44
	Enron	5.34	5.20	4.63	4.85	6.05	**4.61**	4.85
One error(↓)	*Emotions*	52.48	40.59	28.71	26.73	26.24	28.71	**25.74**
	Scene	82.69	24.25	29.43	20.82	20.65	**20.07**	20.15
	Yeast	23.77	23.45	23.12	22.79	23.34	23.88	**22.54**
	Birds	95.34	77.91	43.02	43.60	44.77	**42.44**	**42.44**
	Genbase	0.00	0.50	0.50	**0.00**	0.50	**0.00**	**0.00**
	Medical	53.18	35.04	**14.73**	15.50	18.92	15.04	16.07
	Enron	23.66	30.40	22.11	24.70	32.99	**21.59**	26.42
Average precision(↑)	*Emotions*	59.18	69.38	79.96	81.01	**81.70**	80.09	81.56
	Scene	46.72	85.12	80.69	87.39	87.35	87.90	**88.13**
	Yeast	75.05	75.85	76.98	77.00	76.76	76.63	**77.07**
	Birds	19.31	36.28	61.56	61.00	61.27	61.63	**62.04**
	Genbase	99.14	99.14	99.45	99.62	99.37	99.62	**99.64**
	Medical	62.03	72.56	88.30	87.82	86.34	**88.43**	87.64
	Enron	69.25	62.32	70.64	67.70	66.10	**71.17**	69.17
Counts	mlODM: win/tie/loss	24/1/3	28/0/0	19/1/8	20/2/6	25/0/3	18/2/8	

The best accuracy on each dataset in each measure is bolded

On the other hand, mlODM exceeds the performance that the best-tuned Rank-SVM and Rank-LSVM can achieve over many classic datasets, such as *emotions*, *scene* and *birds*, which verifies the better generalization performance of optimizing the margin distribution.

For the improved generalization performance, we can give an intuitive discussion of mlODM. Unlike taking only the points nearest to hyperplane into account in Rank-SVM, mlODM utilizes the information of data distribution by optimizing the margin distribution. At the same time, the approximation strategy in Sect. 4 makes efficiently solving possible. By introducing the information of data distribution, the method will be more robust and possess better generalization performance. To see this, we can assume the data is unevenly distributed, which is common in the real world, then SVM-style methods con-

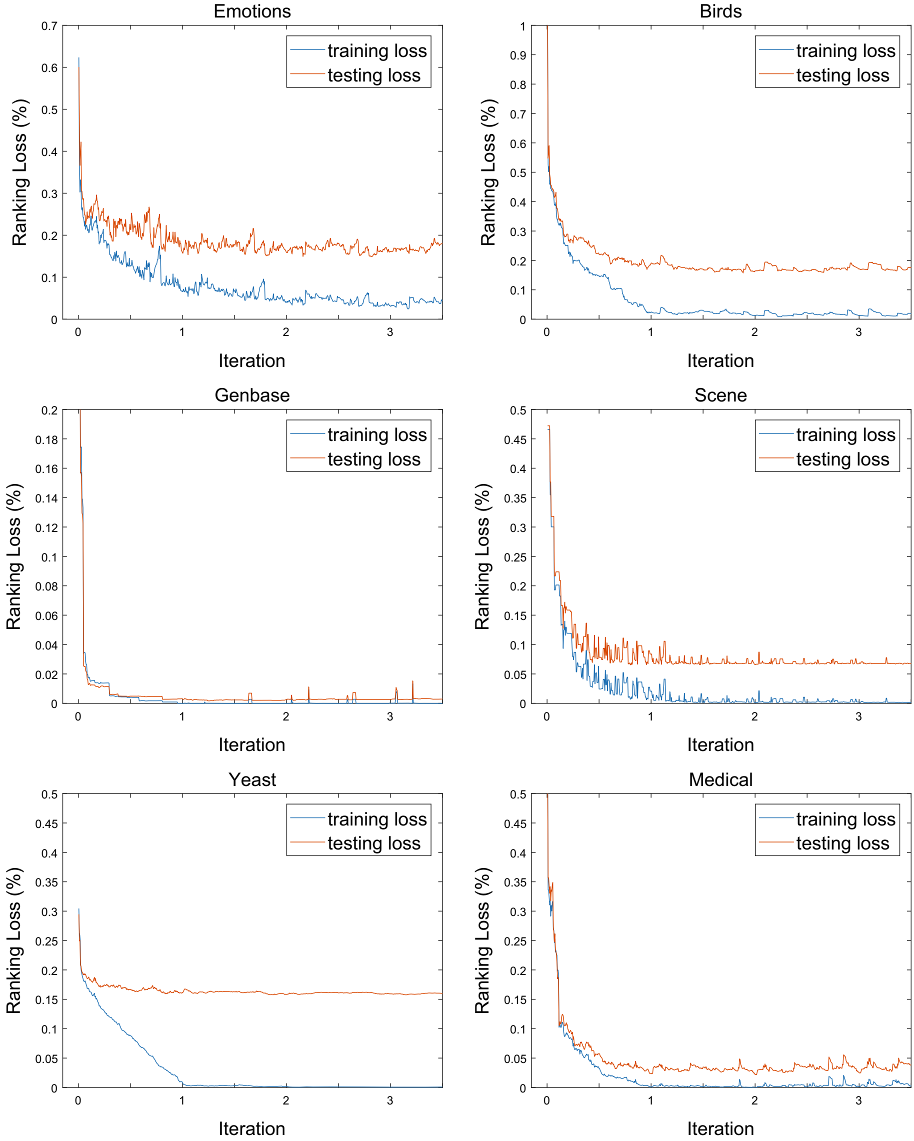

Fig. 1 Training process of mlODM on six datasets

siders only the points near decision boundary, which could be unrepresentative. However, ODM-style methods wish to separate the representative parts on both sides of the decision boundary. Thus it is reasonable that ODM-style methods have better generalization performance in most cases. But when the points nearest to the decision boundary characterize a good classifier, SVM-style methods can achieve similar generalization performance to mlODM.

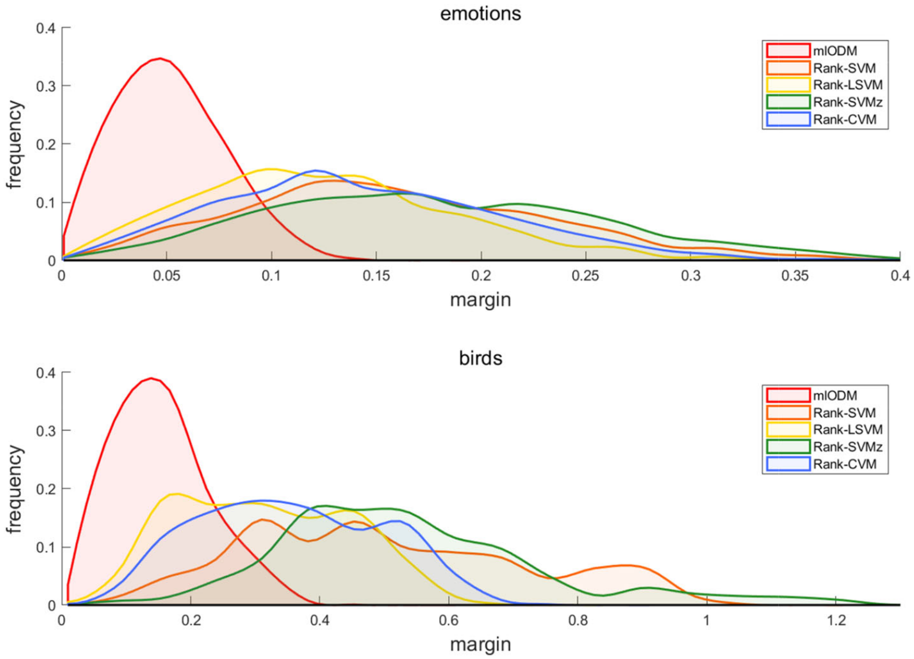

Fig. 2 Margin distribution of mlODM, Rank-SVM, Rank-CVM, Rank-SVMz and Rank-LSVM

6.3 Training process

We visualize the training process of mlODM on six datasets in this subsection, to verify the fast convergence rate as mentioned in Sect. 5. Figure 1 shows the training process on ranking loss over the training and testing data on six datasets. All coordinates are updated during an iteration. The figure illustrates that mlODM converges very fast in most datasets. Specifically, the testing loss of all training process converges within one iteration. According to the analysis in Sect. 6.2, this experiment indicates that although mlODM utilizes the information of data distribution, which seems more complicated than SVM-style methods, it still can be solved efficiently enough. The reason is that the dual problem of mlODM is still strictly convex and satisfies the assumptions that Richtárik and Takáč (2014) proposed, which results in the linear convergence rate of optimization.

6.4 Comparison of margin distribution

In this subsection, we empirically analyze the margin distribution of margin-based methods, i.e., mlODM, Rank-SVM, Rank-CVM, Rank-SVMz and Rank-LSVM, as shown in Figs. 2 and 3. It is obvious that mlODM obtains better margin distribution than other four methods, which means the distribution of margin is more concentrated. The figure also illustrates that SVM-style methods often have bad margin distribution such as *medical*, *birds* and *genbase*, the reason of which is the points nearest to the classification hyperplane can not always be representative. In general, without considering the distribution of data, the generalization performance of SVM-style methods is not always promising. In the experiments, the choice

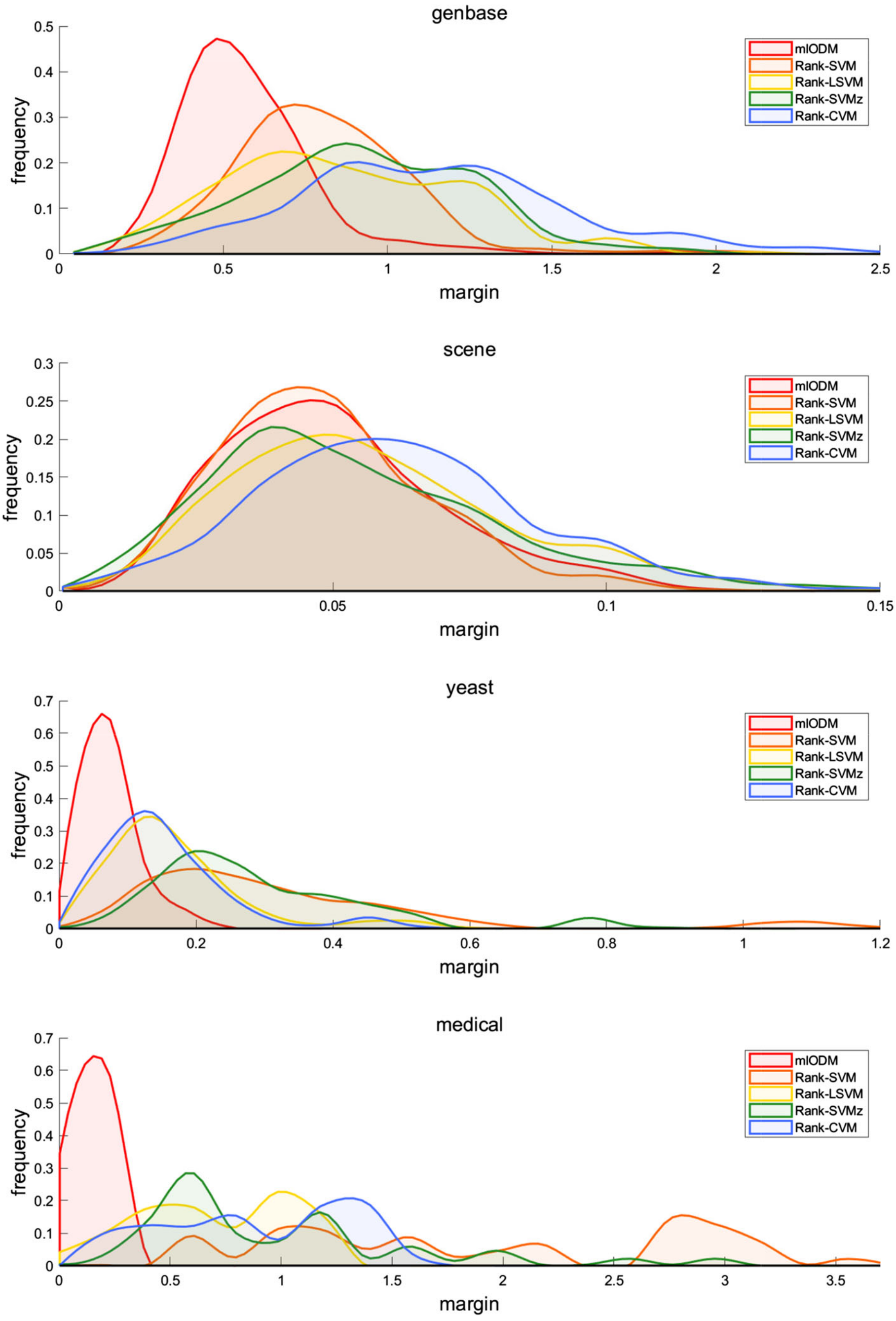

Fig. 3 Margin distribution of mlODM, Rank-SVM, Rank-CVM, Rank-SVMz and Rank-LSVM

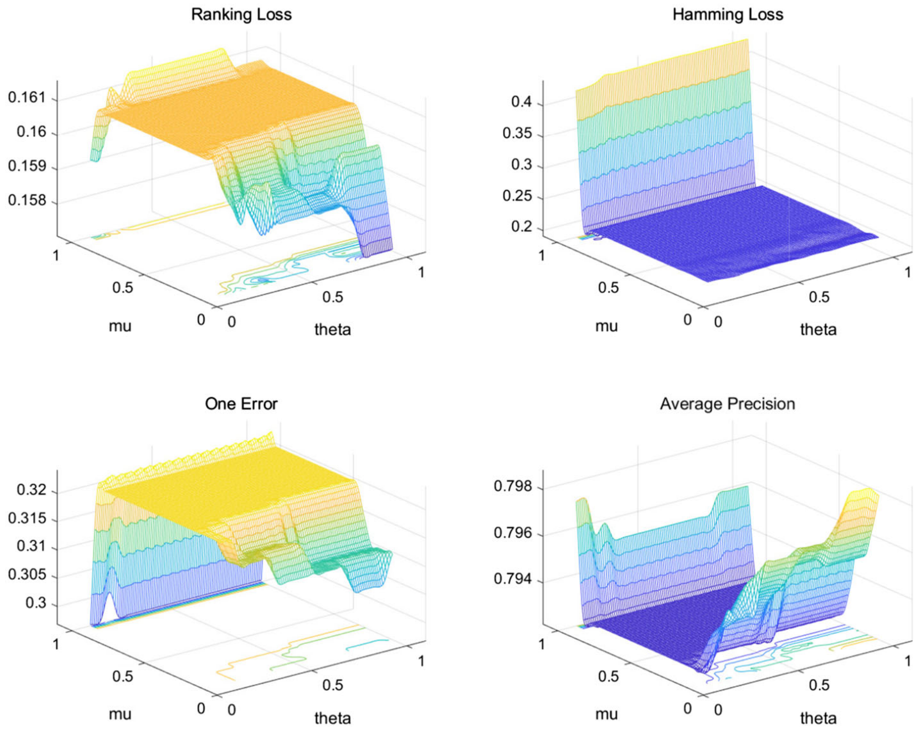

Fig. 4 Effect of hyperparameters of mlODM on four metrics over *Emotions*

of hyperparameters also has an effect on the margin distribution, so we utilize the uniform parameter settings, which is $C = 1$ and $\gamma = 2^{-1}$.

6.5 Effect of hyperparameters

In our proposed mlODM method, two hyperparameters, θ and μ, are introduced to improve sparsity and trade off the penalty on different sides respectively. Figure 4 presents the effect of hyperparameters of mlODM on four metrics over *Emotions*. The figure shows that both hyperparameters result in a smooth change of loss value, which makes it convenient to adjust hyperparameters and the method credible. Specifically, small μ and big θ is a good choice for *Emotions*.

7 Related work

This work is related to two branches of studies. The first one is SVM-style multi-label learning approaches. Support vector machine (SVM) (Cortes and Vapnik 1995) has been one of the most successful machine learning techniques in the past few decades, with kernel methods providing a powerful and unified framework for nonlinear problems (Schölkopf and Smola 2001). Elisseeff and Weston (2002) first applied this framework to multi-label learning and proposed Rank-SVM, which has been one of the most famous multi-label learning methods. Like binary SVM, the Rank-SVM can also be represented as minimizing the empirical ranking loss with a regularization term controlling the model complexity. Accordingly Tsochantaridis et al. (2005) extended the framework to a general form of structured output classification. In this general formulation, the ranking loss function can be replaced. For example, Guo and Schuurmans (2011) proposed *calibrated separation ranking loss* by using simpler dependence structure and obtain better generalization performance.

There are numerous work to improve Rank-SVM in efficiency or performance. Specifically, considering that the threshold selected may be not the optimal due to its separation from the training process, Jiang et al. (2008) proposed Calibrated-RankSVM. To accelerate the time-consuming training process, Xu (2012) proposed SVM-ML which consisted of adding a zero label to detect relevant labels and simplified the original form of Rank-SVM; Xu (2013b) use Random Block Coordinate Descent method to solve the dual problem instead of Frank–Wolfe algorithm. Both methods significantly reduced the computational cost and obtained competitive performance. In addition, there are also a number of variants and applications of Rank-SVM. For example, Xu (2016) generalized Lagrangian support vector machine (LSVM) to multi-label learning and proposed Rank-LSVM; Liu et al. (2015) proposed rank-wavelet SVM (Rank-WSVM) for the classification of power quality complex disturbances.

The second branch of studies is utilizing margin distribution in classification tasks. Although the above framework has been successful and the performance is promised by margin theory (Vapnik 1995), all of the above methods are based on large margin formulation. However, the studies in margin theory for Boosting (Schapire et al. 1998; Reyzin and Schapire 2006; Gao and Zhou 2013) have finally disclosed that maximizing the minimum margin does not necessarily lead to better generalization performance, and instead, the margin distribution has been proven to be more crucial. Later, inspired by this idea, Zhang and Zhou (2014b, 2019) proposed Large margin Distribution Machine (LDM) and its simplified version optimal margin distribution machine (ODM) for binary classification. Thereafter, varieties of methods based on margin distribution have been proposed. Zhou and Zhou (2016) and Zhang and Zhou (2017, 2018) generalized ODM to class imbalance learning, multi-class learning and unsupervised learning respectively. In weakly supervised learning (Zhou 2018), Zhang and Zhou (2018a) proposed the semi-supervised ODM(ssODM), which achieved significant improvement in performance compared to SVM-based methods. Lv et al. (2018) introduced margin distribution into neural networks and proposed the Optimal margin Distribution Network (mdNet), which outperforms the cross-entropy loss model.

However, for the more general learning paradigm in real-world tasks, i.e, the multi-label learning, whether optimizing the margin distribution is still effective is still unknown. By first introducing this idea into multi-label classification, this paper proposes multi-label optimal margin distribution machine (mlODM) and shows its superiority with extensive experiments.

8 Conclusion

In this paper, we propose a multi-label classification method named mlODM, which first extends the idea of optimizing the margin distribution to multi-label learning. Based on the approximation of margin mean and margin variance like binary ODM, and the simplification technique in Rank-SVM, we propose the formulation of mlODM in Sect. 4. Subsequently we use block coordinate descent method to solve the problem efficiently considering the structure of the optimization problem in Sect. 5. Empirically, extensive experiments compared to classic methods in different measures verify the superiority of our method. Finally, the visualization of margin distribution and convergence analyzes the characteristic of our method. In the future it will be interesting to solve the sub-problem in a more efficient way to accelerate the method and make theoretical analysis for the good performance of mlODM. Another interesting future issue is to incorporate the proposed method into the recently proposed *abductive learning* (Zhou, 2019), a new paradigm which leverages both machine learning and logical reasoning, to enable it handle multi-label concepts.

Acknowledgements This research was supported by the National Key R&D Program of China (2018YFB1004300), NSFC (61673201), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, *37*(9), 1757–1771.

Breiman, L. (1999). Prediction games and arcing algorithms. *Neural Computation*, *11*(7), 1493–1517.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Elisseeff, A., & Weston, J. (2002). A kernel method for multi-labelled classification. In T. G. Dietterich, S. Becker and Z. Ghahramani (Eds.), *Advances in neural information processing systems* (pp. 681–687). MIT Press.

Frank, M., & Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, *3*(1–2), 95–110.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Gao, W., & Zhou, Z. H. (2013). On the doubt about margin explanation of boosting. *Artificial Intelligence*, *203*, 1–18.

Guo, Y., & Schuurmans, D. (2011). Adaptive large margin training for multilabel classification. In: W. Burgard and D. Roth (Eds.), *25th AAAI conference on artificial intelligence*. San Francisco, CA: AAAI Press.

Jiang, A., Wang, C., & Zhu, Y. (2008). Calibrated Rank-SVM for multi-label image categorization. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1450–1455). IEEE.

Liu, Z., Cui, Y., & Li, W. (2015). A classification method for complex power quality disturbances using EEMD and rank wavelet SVM. *IEEE Transactions on Smart Grid*, *6*(4), 1678–1685.

Lv, S. H., Wang, L., & Zhou, Z. H. (2018). Optimal margin distribution network. arXiv preprint arXiv:1812.10761

McCallum, A. (1999). Multi-label text classification with a mixture model trained by EM. In *AAAI workshop on text learning* (pp. 1–7)

Reyzin, L., & Schapire, R. E. (2006). How boosting the margin can also boost classifier complexity. In *Proceedings of the 23rd international conference on machine learning* (pp. 753–760). ACM.

Richtárik, P., & Takáč, M. (2014). Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, *144*(1–2), 1–38.

Schapire, R. E., Freund, Y., Bartlett, P., Lee, W. S., et al. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, *26*(5), 1651–1686.

Schapire, R. E., & Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. *Machine Learning*, *39*(2–3), 135–168.

Schölkopf, B., & Smola, A. J. (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge: MIT Press.

Sha, F., Saul, L. K., & Lee, D. D. (2002). Multiplicative updates for nonnegative quadratic programming in support vector machines. In S. Becker, S. Thrun and K. Obermayer (Eds.,) *Advances in neural information processing systems* (pp. 1041–1048). MIT Press.

Tan, Z. H., Zhang, T., & Zhou, Z. H. (2019). Coreset stochastic variance-reduced gradient with application to optimal margin distribution machine. In *33rd AAAI conference on artificial intelligence*.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*(Sep), 1453–1484.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2011a). Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, *23*(7), 1079–1089.

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J., & Vlahavas, I. (2011b). Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, *12*, 2411–2414.

Turnbull, D., Barrington, L., Torres, D., & Lanckriet, G. (2008). Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing*, *16*(2), 467–476.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Berlin: Springer.

Wu, X. Z., & Zhou, Z. H. (2017). A unified view of multi-label performance measures. In *Proceedings of the 34th international conference on machine learning* (Vol. 70, pp. 3780–3788). JMLR.org.

Xu, J. (2012). An efficient multi-label support vector machine with a zero label. *Expert Systems with Applications*, *39*(5), 4796–4804.

Xu, J. (2013a). Fast multi-label core vector machine. *Pattern Recognition*, *46*(3), 885–898.

Xu, J. (2013b). A random block coordinate descent method for multi-label support vector machine. In *International conference on neural information processing* (pp. 281–290). Berlin: Springer.

Xu, J. (2016). Multi-label lagrangian support vector machine with random block coordinate descent method. *Information Sciences*, *329*, 184–205.

Zhang, M. L., Li, Y. K., Liu, X. Y., Xin, G. (2018). Binary relevance for multi-label learning: an overview. *Frontiers of Computer Science*, *12*(2), 191–202.

Zhou, Z. H. (2018). A brief introduction to weakly supervised learning. *National Science Review, 5*(1), 44–53.

Zhang, M. L., & Zhou, Z. H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, *18*(10), 1338–1351.

Zhang, M. L., & Zhou, Z. H. (2007). Ml-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, *40*(7), 2038–2048.

Zhang, M. L., & Zhou, Z. H. (2014a). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *26*(8), 1819–1837.

Zhang, T., & Zhou, Z. H. (2014b). Large margin distribution machine. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 313–322). ACM.

Zhang, T., & Zhou, Z. H. (2017). Multi-class optimal margin distribution machine. In *Proceedings of the 34th international conference on machine learning* (Vol. 70, pp. 4063–4071). JMLR.org.

Zhang, T., & Zhou, Z. H. (2018). Optimal margin distribution clustering. In *22nd AAAI conference on artificial intelligence*.

Zhang, T., & Zhou, Z. H. (2018a). Semi-supervised optimal margin distribution machines. In Jérôme Lang (ed.) Proceedings of the 27th international joint conference on artificial intelligence (pp. 3104–3110). Stockholm, Sweden: IJCAI.

Zhou, Z. H. (2019). Abductive learning: Towards bridging machine learning and logical reasoning. *Science China Information Sciences, 62*(7), 76101.

Zhang, T., & Zhou, Z. (2019). Optimal margin distribution machine. In *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2019.2897662.

Zhou, Y. H., & Zhou, Z. H. (2016). Large margin distribution learning with cost interval and unlabeled data. *IEEE Transactions on Knowledge and Data Engineering*, *28*(7), 1749–1763.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.