

Maximum Margin Multiple Instance Clustering with Applications to Image and Text Clustering

Dan Zhang, Fei Wang, Luo Si, and Tao Li

Abstract—In multiple instance learning problems, patterns are often given as *bags* and each bag consists of some *instances*. Most of existing research in the area focuses on multiple instance classification and multiple instance regression, while very limited work has been conducted for multiple instance clustering (MIC). This paper formulates a novel framework, maximum margin multiple instance clustering (M^3IC), for MIC. However, it is impractical to directly solve the optimization problem of M^3IC . Therefore, M^3IC is relaxed in this paper to enable an efficient optimization solution with a combination of the constrained concave-convex procedure and the cutting plane method. Furthermore, this paper presents some important properties of the proposed method and discusses the relationship between the proposed method and some other related ones. An extensive set of empirical results are shown to demonstrate the advantages of the proposed method against existing research for both effectiveness and efficiency.

Index Terms—Constrained concave-convex procedure, cutting plane, maximum margin, multiple instance clustering.

I. INTRODUCTION

MULTIPLE instance learning (MIL) can be viewed as a variation of the traditional learning methods for problems with incomplete knowledge on the examples (or *instances*). In the MIL setting, patterns are given as *bags*, and each bag consists of some instances. Most of the MIL research focuses on solving binary classification [1]–[3] and regression problems [4]–[6]. In a binary multiple instance classification problem, the labels are assigned to bags, rather than instances. A typical assumption is that a bag should be labeled as positive if at least one of its instances is positive, and negative if all of its instances are negative. Given a binary multiple instance classification problem, the main objective of MIL is to learn a classifier based on the labeled bags, and use this classifier to determine the labels for unlabeled bags. MIL has been widely employed in areas such as text mining [1], drug design [2], localized content-based image retrieval (LCBIR) [3], etc.

Manuscript received January 14, 2010; revised October 19, 2010 and January 4, 2011; accepted January 7, 2011. Date of publication March 28, 2011; date of current version May 4, 2011. This work was supported in part by the National Science Foundation of China under Grant CCF-0939370, Grant IIS-1017837, and Grant IIS-0746830.

D. Zhang and L. Si are with the Department of Computer Science, Purdue University, West Lafayette, IN 47906 USA (e-mail: danzhang2008@gmail.com; lsi@cs.purdue.edu).

F. Wang is with the Healthcare Transformation Group, IBM T. J. Watson Research Center, Hawthorne, NY 10532 USA (e-mail: fwang@us.ibm.com).

T. Li is with the School of Computer Science and Information Sciences, Florida International University, Miami, FL 33199 USA (e-mail: taoli@cs.fiu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNN.2011.2109011

As another branch of machine learning, clustering [7] is a research topic in both data mining and machine learning, and plays an important role in applications such as text mining, web analysis, marketing, etc. [8]. It aims at dividing the whole dataset into several clusters of similar objects, or the hidden patterns in an unsupervised way, where these different patterns may correspond to different underlying data concepts.

However, almost all of the current clustering methods are designed for the traditional learning problems, while in many cases they should be better formulated as MIL problems.

Similar to LCBIR [3], in image clustering there is natural ambiguity as to what portion of the image contains the clustering concept, where the concept can be a tiger, an elephant, etc., while most portion of the image may be irrelevant. In this case, we can treat each image as a bag, and each instance in this bag corresponds to a region in this image. It is clear that this application requires the solution of multiple instance clustering (MIC) to help users to partition these bags.

As a popular topic in text mining, almost all the previous text clustering methods [9] are designed for the traditional single instance learning problems, i.e., each document is treated as an instance. However, it is very likely that not all parts of a document are related to a specific topic. Text clustering can be better formulated as a multiple-instance problem, where different articles are represented as different collections of overlapping text passages [1]. Then, each document can be regarded as a bag, and the passages in it are instances in this bag. To cluster these bags, we need the solution of MIC.

Recently, very limited research has been conducted for MIC. In [10], the authors regard bags as atomic data items and use some forms of distance metric to measure the distances between bags. Then they adapt the K-medoids algorithm to cluster these bags based on the defined distances. Their method is efficient in some applications such as drug discovery. But, as claimed by [3], defining distances between bags in an unsupervised way may not be able to reflect their actual content differences. For example, two pictures may share identical background and only differ in that one contains an elephant and the other contains a fox. By using the minimal Hausdorff distance to measure distances between bags [11], the distance between these two pictures will be very low even though their actual contents (or concepts) may differ. The calculation of the distances between bags is quite time consuming, since it needs to calculate all the distances between instances in different bags.

This paper proposes a novel maximum margin multiple instance clustering (M^3IC) framework. The new formulation

aims at finding the desired hyperplanes that maximize the margin differences on at least one instance per bag in an unsupervised way. The initial formulation of M³IC is a non-convex optimization problem, which is impossible to solve directly. Therefore, we relax the original M³IC problem and propose a method, i.e., M³IC-MBM, which is a combination of the constrained concave-convex procedure (CCCP) and cutting plane methods, to solve the relaxed optimization task.

This paper substantially extends our preliminary work in [12]. Compared to the previous work, here we derive the MIC framework from a different viewpoint, making it clearer. Moreover, in this paper, we derive the dual form of the corresponding objective function and give proofs to some important theorems. Furthermore, to validate the performance of our method, more experiments are conducted on more datasets and the analysis is more precise.

II. RELATED WORKS

A. Multiple Instance Classification

The notion of MIL was first introduced by Dietterich *et al.* [2] to deal with the drug activity prediction. After that, many researchers have studied the binary multiple instance classification problem. These algorithms can be roughly divided into three groups: the group that is specifically designed to solve multiple-instance problems, the group that tries to modify single-instance learning for MIL by introducing MIL constraints, and the group that tries to convert MIL to a traditional single-instance problem and solve it using traditional learning methods.

For the first group, the first MIL method is APR [2], which represents positive instances by an axis-parallel rectangle in the feature space. Maron and Lozano-Pérez proposed a method called diverse density (DD) [13], [14], which intends to find a concept point in the feature space that resembles positive instance most, and classify instances according to the distances between the instances and this concept point. In [6] and [15], the authors accelerated the DD method by using expectation-maximization (EM) and proposed EM-DD.

As for the second group, Andrews *et al.* [1] proposed two MIL formulations based on support vector machine (SVM) [16], one (mi-SVM) for the instance-based classification and the other (MI-SVM) for bag-level classification. Since the MIL formulations are all nonconvex, Gehler and Chapelle applied deterministic annealing to solve them and obtained a better local solution [17]. Gärtner *et al.* [18] proposed a kernel function directly for bags. Later, Kwok and Cheung [19] extended their work by proposing a marginalized MI kernel and converting the MIL problem from an incomplete data problem to a complete data problem. In [20], the authors make some modifications on the loss functions of single-instance SVM and focus more on the positive bags with smaller sizes.

For the third group, DD-SVM [21] selects a set of prototypes from the local solutions of DD method and then a SVM is trained on the basis of the bag features summarized by these selected prototypes. In [22], bags are embedded into a feature space defined by instances, and a 1-norm SVM is applied to build the bag level classifiers.

A common strait of these classification methods is that they need to utilize the supervised information of all the labeled bags to determine which part/parts are the desired object/objects. However, in MIC, such label information is not available. So, it is hard to determine the desired concepts underlying different clusters, which is the main obstacle to solve the MIC problem.

B. MIC

Recently, very limited research [10] addresses the task of MIC. In [10], the authors defined the distances between bags in three different ways, i.e., the maximal, minimal, and the average Hausdorff distance [4], [10], [11]. These methods consider bags as atomic data items, and the K-medoids algorithm can be employed to cluster these data items based on the distances between them. Although these methods define the distances between bags in different ways, they are all unsupervised methods. Without identifying the desired objects/concepts in each bag, unsupervised distance definitions cannot reveal the true distances between bags. Different from these methods, the proposed method solves this problem more elegantly by incorporating the cluster assignments into the MIC formulation, the details of which will be explained in Section III.

C. Maximum Margin Clustering (MMC)

Maximum margin learning is a key technique for classification and dimensionality reduction [23]. Recently, the idea of maximum margin learning has also been applied to data clustering, which is usually referred to as MMC. MMC was first proposed in [24]. In [24], the authors borrowed the idea of a standard machine learning principle—maximum margin principle, and used it for clustering. More precisely, they try to assign instances to two classes $\{-1, +1\}$ so that the separation between the two classes can be as large as possible. The MMC method has a solid theoretical foundation and performs much better than the previous methods. But at first it could only deal with the two-class separation problem. In a later work [25], it was extended to the multiclass case. However, one of the potential problems in these methods is that they are too time consuming. In [26]–[29], the authors relaxed the original problem and solved it in an efficient way. Although greatly improved, MMC can only deal with the traditional learning problem. In this paper, similar to MMC, we extend the maximum margin principle to MIC and propose M³IC.

III. PROPOSED METHOD

A. Problem Statement

Suppose a set of n bags $\{\mathbf{B}_i, i = 1, 2, \dots, n\}$ is given, and the instances in the bag \mathbf{B}_i are denoted as $\{\mathbf{B}_{i1}, \mathbf{B}_{i2}, \dots, \mathbf{B}_{in_i}\}$, where n_i is the total number of instances in this bag. The goal of MIC is to partition this given dataset into k clusters such that the concepts in different clusters can be “distinct” from each other. A $1 \times n$ vector \mathbf{y} is used to denote the cluster assignment, with y_i being the cluster assignment for bag \mathbf{B}_i .

B. Motivation

In Section II-A, we reviewed binary multiple instance classification methods. Most of them are developed on the basis of the typical assumption that each positive bag contains at least one positive instance, while instances in negative bags are all negative. Given labels, the multiple instance classifier can be learnt by simply looking for the common concepts in positive bags, which are also far away from all of the negative instances. Since any instance in a positive bag can be either positive or negative, looking for the real positive instance is very difficult and cannot be formulated as a convex optimization problem.

MIC is a much harder problem than the previous binary multiple instance classification problems. In MIC, no label information is given beforehand, and therefore it is even harder to find the different concepts representing different clusters. Another problem is that almost all the traditional multiple instance learning methods can only classify the unlabeled bags as either positive or negative ones. This is because almost all the current MIL methods were designed to solve application problems where only the relevances of objects need to be determined, such as in LCBIR and drug discovery. But in MIC, the whole dataset needs to be categorized into several, probably more than two, clusters.

In MIC, we need to both identify the most relevant instances and cluster these instances into several different groups. Both these two objectives are very difficult, let alone couple them together. Fortunately, the idea of MMC [24], [25], [27], [28], which was introduced in Section II-C, gives us some inspirations to solve this problem. The clustering principle we investigate here is to find a desired labeling so that if one were to subsequently run a multiple instance SVM [1], the margin obtained would be maximal over all possible labels.

C. Formulation

In this subsection, the M³IC problem will be formulated. For each class $p \in \{1, \dots, k\}$, we define a weight vector \mathbf{w}_p . We wish to solve for a labeling \mathbf{y} that leads to the maximum (soft) margin. Straightforwardly, one could attempt to tackle the following optimization problem directly:

$$\begin{aligned} \min_{\mathbf{w}_1, \dots, \mathbf{w}_k, y_1, \dots, y_n, \xi_i \geq 0} \quad & \frac{1}{2} \sum_{p=1}^k \|\mathbf{w}_p\|^2 + \frac{C}{n} \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & \forall i \in \{1, \dots, n\}, \quad \forall p_i \in \{1, 2, \dots, k\} \setminus y_i \\ & \bigcup_{j=1}^{n_i} I(\mathbf{w}_{y_i}^T \mathbf{B}_{ij} - \mathbf{w}_{p_i}^T \mathbf{B}_{ij} \geq 1 - \xi_i) = 1 \end{aligned} \quad (1)$$

where $\sum_{i=1}^n \xi_i$ is divided by n to make this formulation scalable to the dataset size and “ \setminus ” means ruling out. $I(\text{expression})$ is an indication function. It is true when the “*expression*” holds. “ \bigcup ” is an union operator. \mathbf{B}_{ij} is defined in Section III-A. It can be seen that the large margin constraint is imposed on at least one instance per bag. The reason for this is similar to the assumption in the binary multiple instance classification problems, i.e., there should be at least one common concept in each bag. But there are also some differences. In binary

multiple instance classification problems, negative bags are often referred to as irrelevant information, which may not represent any interesting group (for example, in LCBIR, negative bags are always some arbitrary background pictures). However, in MIC, we need to partition the whole dataset into k groups, and therefore the common concept constraint should be imposed on all the clusters within the whole dataset.

The form of problem (1) is very complicated. However, as shown in Theorem 1, it can be transformed to a much more simplified form, in which the number of variables that need to be optimized could be reduced as well.

Theorem 1: Problem (1) is equivalent to

$$\begin{aligned} \min_{\mathbf{w}_1, \dots, \mathbf{w}_k, \xi_i \geq 0} \quad & \frac{1}{2} \sum_{p=1}^k \|\mathbf{w}_p\|^2 + \frac{C}{n} \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & i = 1, \dots, n, \quad \max_{j \in \mathbf{B}_i} \left(\mathbf{w}_{u_{ij}^*}^T \mathbf{B}_{ij} - \mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij} \right) \geq 1 - \xi_i \end{aligned} \quad (2)$$

where $u_{ij}^* = \arg \max_p (\mathbf{w}_p^T \mathbf{B}_{ij})$, $v_{ij}^* = \arg \max_{p \setminus u_{ij}^*} (\mathbf{w}_p^T \mathbf{B}_{ij})$,¹ and the optimal value for y_i , which is \hat{y}_i , equals $\arg \max_{y_i \in \{1, 2, \dots, k\}} \mathbf{w}_{y_i}^T \mathbf{B}_{ij^*}$, where $j^* = \arg \max_{j \in \mathbf{B}_i} [\mathbf{w}_{u_{ij}^*}^T \mathbf{B}_{ij} - \mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij}]$.

Proof: It is clear that $\bigcup_{j=1}^{n_i} I(\mathbf{w}_{y_i}^T \mathbf{B}_{ij} - \mathbf{w}_{p_i}^T \mathbf{B}_{ij} \geq 1 - \xi_i) = 1$ is equivalent to $I(\max_{j \in \mathbf{B}_i} (\mathbf{w}_{y_i}^T \mathbf{B}_{ij} - \mathbf{w}_{p_i}^T \mathbf{B}_{ij}) \geq 1 - \xi_i) = 1$. So, the constraints in problem (1) can be transformed to

$$\begin{aligned} \forall i \in \{1, \dots, n\}, \quad \forall p_i \in \{1, 2, \dots, k\} \setminus y_i \\ \max_{j \in \mathbf{B}_i} \left(\mathbf{w}_{y_i}^T \mathbf{B}_{ij} - \mathbf{w}_{p_i}^T \mathbf{B}_{ij} \right) \geq 1 - \xi_i. \end{aligned} \quad (3)$$

Since the second part of the problem 1 is $C/n \sum_{i=1}^n \xi_i$, it is evident that the optimal value for ξ_i can be expressed as

$$\hat{\xi}_i = \min_{y_i \in \{1, 2, \dots, k\}} \max_{p_i \in \{1, 2, \dots, k\} \setminus y_i} \max_{j \in \mathbf{B}_i} \left\{ 1 - \left(\mathbf{w}_{y_i}^T \mathbf{B}_{ij} - \mathbf{w}_{p_i}^T \mathbf{B}_{ij} \right), 0 \right\}.$$

For any $j \in \mathbf{B}_i$, if $\mathbf{w}_{y_i}^T \mathbf{B}_{ij} < \mathbf{w}_{p_i}^T \mathbf{B}_{ij}$, then the right-hand side of this equation will be larger than 1. So, \hat{y}_i , which is the optimal solution of y_i , can only take values from u_{ij}^* s that achieves the largest output within the instances in bag \mathbf{B}_i , i.e., $\hat{y}_i \in \{u_{ij}^*, j = 1, \dots, n_i\}$ and $u_{ij}^* = \arg \max_{y_{ij} \in \{1, 2, \dots, k\}} \mathbf{w}_{y_{ij}}^T \mathbf{B}_{ij}$. Similarly, the optimal value \hat{p}_i would also take values from v_{ij}^* s that satisfy $v_{ij}^* = \arg \max_{p_{ij} \setminus u_{ij}^*} \mathbf{w}_{p_{ij}}^T \mathbf{B}_{ij}$. Since the optimization of $[1 - (\mathbf{w}_{y_i}^T \mathbf{B}_{ij} - \mathbf{w}_{p_i}^T \mathbf{B}_{ij})]$ is also taken w.r.t. all the instances in \mathbf{B}_i , it is clear that $\hat{y}_i = \arg \max_{y_i \in \{1, 2, \dots, k\}} \mathbf{w}_{y_i}^T \mathbf{B}_{ij^*}$, where $j^* = \arg \max_{j \in \mathbf{B}_i} [\mathbf{w}_{u_{ij}^*}^T \mathbf{B}_{ij} - \mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij}]$. Integrating j^* and subjecting the solution of \hat{y}_i to the constraint (3), this constraint turns to

$$i = 1, \dots, n, \quad \max_{j \in \mathbf{B}_i} \left(\mathbf{w}_{u_{ij}^*}^T \mathbf{B}_{ij} - \mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij} \right) \geq 1 - \xi_i. \quad \blacksquare$$

By employing Theorem 1, compared to the original optimization problem (1) the number of variables that need to be optimized is reduced by n . Besides the direct derivative from problem (1), this transformed problem (2) can also be perceived from the large margin perspective. In particular, the *Bag*

¹This equation can also be written as $v_{ij}^* = \arg \max_{p \neq u_{ij}^*} (\mathbf{w}_p^T \mathbf{B}_{ij})$.

Margin (BM) for a bag \mathbf{B}_i is defined as $\max_{j \in \mathbf{B}_i} (\mathbf{w}_{u_{ij}^*}^T \mathbf{B}_{ij} - \mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij})$. Problem (2) can be interpreted as follows: instead of labeling all the samples by running an SVM implicitly over all possible labels as that in MMC [24], in M³IC we try to find a labeling on bags that results in classifiers that maximize margins on bags, and, for each bag, its BM is determined by its most “discriminative” instance.

Beside the large margin constraint, one has to enforce the class balance constraint to avoid the trivially “optimal” solution, where all of the bags are assigned to one cluster and achieve an unbounded margin, and to minimize the impact of some outliers, which may form some very small clusters [24]. Then, the formulation of M³IC can be formally defined as

$$\begin{aligned} \min_{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k, \zeta_i \geq 0} \quad & \frac{1}{2} \sum_{p=1}^k \|\mathbf{w}_p\|^2 + \frac{C}{n} \sum_{i=1}^n \zeta_i \\ \text{s.t. } \quad & i = 1, \dots, n, \quad \max_{j \in \mathbf{B}_i} (\mathbf{w}_{u_{ij}^*}^T \mathbf{B}_{ij} - \mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij}) \geq 1 - \zeta_i \\ & \forall p, q \in \{1, 2, \dots, k\} \\ & -l \leq \sum_{i=1}^n \sum_{j \in \mathbf{B}_i} I_{ij} \mathbf{w}_p^T \mathbf{B}_{ij} - \sum_{i=1}^n \sum_{j \in \mathbf{B}_i} I_{ij} \mathbf{w}_q^T \mathbf{B}_{ij} \leq l. \end{aligned} \quad (4)$$

Here, I_{ij^*} equals 1 if $j^* = \arg \max_{j \in \mathbf{B}_i} (\mathbf{w}_{u_{ij}^*}^T \mathbf{B}_{ij} - \mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij})$, and otherwise 0. l is a parameter that controls the cluster balance as in [24]. It is clear that, in this formulation, these two sets of constraints are imposed only on the instances that determine the bag margins of their corresponding bags. Once these “witness” instances have been identified, the other instances become irrelevant. If we can obtain results from problem (4), the cluster assignment of a specific bag \mathbf{B}_i can be determined by $y_i = \arg \max_p \sum_{j \in \mathbf{B}_i} I_{ij} \mathbf{w}_p^T \mathbf{B}_{ij}$.

However, it is impossible to get the exact solution of problem (4) efficiently. In the following sections, we will state the reason, and suggest a method to approximate it.

D. M³IC-MBM

For the first set of constraints in problem (4), i.e., $\max_{j \in \mathbf{B}_i} (\mathbf{w}_{u_{ij}^*}^T \mathbf{B}_{ij} - \mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij}) \geq 1 - \zeta_i$, the convexity of $\mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij}$ cannot be determined and the “max” function also makes this constraint nonconvex. For the second set of constraints, the indication function I_{ij} makes these constraints nonconvex. Therefore, it is impractical to search for the global optimal solution of the optimization problem (4).

1) *Relaxation*: The nonconvexity of problem (4) is mainly caused by the two sets of constraints. So, we consider relaxing these two sets of constraints, so that a solution that is as close to that of the original problem as possible can be approximated.

First of all, the first set of constraints is relaxed. To achieve this goal, a new notion, i.e., modified bag margin (MBM) is introduced. For the bag \mathbf{B}_i , MBM is defined as follows:

$$\max_{j \in \mathbf{B}_i} \left(\max_u \mathbf{w}_u^T \mathbf{B}_{ij} - \text{mean}_{v \setminus u_{ij}^*} (\mathbf{w}_v^T \mathbf{B}_{ij}) \right) \quad (5)$$

where the “mean” function calculates the average value of the input function with respect to the subscript variable.

As another way of defining bag margins, MBM can be considered as a relaxation of BM, which would slightly increase the feasible domain of problem (4) but avoid dealing with the intractable second maximal function v_{ij}^* .

Replacing BM with MBM, the first set of constraints in problem (4) becomes $\max_{j \in \mathbf{B}_i} (\max_u \mathbf{w}_u^T \mathbf{B}_{ij} - \text{mean}_{v \setminus u_{ij}^*} (\mathbf{w}_v^T \mathbf{B}_{ij})) \geq 1 - \zeta_i$. This is also equivalent to $(k/k - 1) \max_{j \in \mathbf{B}_i} (\max_u \mathbf{w}_u^T \mathbf{B}_{ij} - \text{mean}_v (\mathbf{w}_v^T \mathbf{B}_{ij})) \geq 1 - \zeta_i$.

For the second constraint in problem (4), the nonconvexity is mainly brought in by the indication function I_{ij} . So, we relax it and rewrite this constraint as follows:

$$\begin{aligned} \forall p, q \in \{1, 2, \dots, k\} \\ -l \leq \sum_{i=1}^n \sum_{j \in \mathbf{B}_i} \frac{1}{n_i} \mathbf{w}_p^T \mathbf{B}_{ij} - \sum_{i=1}^n \sum_{j \in \mathbf{B}_i} \frac{1}{n_i} \mathbf{w}_q^T \mathbf{B}_{ij} \leq l. \end{aligned} \quad (6)$$

Now, the two constraints in the original problem have already been relaxed. Furthermore, for the convenience of computation, without loss of generality, we introduce two *concatenated* vectors as

$$\tilde{\mathbf{w}} = [\mathbf{w}_1^T, \dots, \mathbf{w}_p^T, \dots, \mathbf{w}_k^T]^T, \quad \mathbf{B}_{ij(p)} = [\mathbf{0}, \dots, \mathbf{B}_{ij}^T, \dots, \mathbf{0}]^T \quad (7)$$

where $\mathbf{0}$ is a $1 \times d$ zero vector, and d is the dimension of \mathbf{B}_{ij} . In $\mathbf{B}_{ij(p)}$, only the $(p-1)d$ to pd -th elements are nonzero and equals \mathbf{B}_{ij} and it is clear that $\tilde{\mathbf{w}}^T \mathbf{B}_{ij(p)} = \mathbf{w}_p^T \mathbf{B}_{ij}$.

After the relaxation of the two constraints in (5) and (6) and the introduction of the two *concatenated* vectors in (7), the original problem (4) can be transformed to

$$\begin{aligned} \min_{\tilde{\mathbf{w}}, \zeta_i \geq 0} \quad & \frac{1}{2} \|\tilde{\mathbf{w}}\|^2 + \frac{C}{n} \sum_i \zeta_i \\ \text{s.t. } \quad & i = 1, \dots, n, \\ & \frac{k}{k-1} \max_{j \in \mathbf{B}_i} \left(\max_u \tilde{\mathbf{w}}^T \mathbf{B}_{ij(u)} - \text{mean}_v (\tilde{\mathbf{w}}^T \mathbf{B}_{ij(v)}) \right) \\ & \geq 1 - \zeta_i \quad \forall p, q \in \{1, 2, \dots, k\}, -l \\ & \leq \sum_{i=1}^n \sum_{j \in \mathbf{B}_i} \frac{1}{n_i} \tilde{\mathbf{w}}^T (\mathbf{B}_{ij(p)} - \mathbf{B}_{ij(q)}) \leq l. \end{aligned} \quad (8)$$

In the following sections, we will propose a method, i.e., M³IC with MBM (M³IC-MBM), which is characterized by an outer CCCP iteration and an inner cutting plane iteration, to solve this relaxed problem.

2) *CCCP Decomposition*: Although some relaxations and simplifications have been made on the original problem, its first set of constraints is still nonconvex. Fortunately, it is the difference between two convex functions. The CCCP is just devised to solve this kind of optimization problems [30]. Next, we will show how to use CCCP to solve this problem.

To simplify the formulation, let $f(\tilde{\mathbf{w}}, i)$ be $(k/k - 1) \max_{j \in \mathbf{B}_i} g(\tilde{\mathbf{w}}, i, j)$ and $g(\tilde{\mathbf{w}}, i, j)$ be $\max_u \tilde{\mathbf{w}}^T \mathbf{B}_{ij(u)} - \text{mean}_v (\tilde{\mathbf{w}}^T \mathbf{B}_{ij(v)})$. The first set of constraints in problem (8) turns to $f(\tilde{\mathbf{w}}, i) \geq 1 - \zeta_i$. It is obvious that this set of constraints is, although not convex, the difference of two convex functions, which could be solved by using CCCP.

Given an initial point $\tilde{\mathbf{w}}^{(0)}$, CCCP computes $\tilde{\mathbf{w}}^{(t+1)}$ from $\tilde{\mathbf{w}}^{(t)}$ iteratively by replacing $f(\tilde{\mathbf{w}}, i)$ with its first-order Taylor expansions at $\tilde{\mathbf{w}}^{(t)}$ and solves the resulting quadratic programming problem until convergence.

Therefore, to use CCCP, we should first calculate the first-order Taylor expansion of $f(\tilde{\mathbf{w}}, i)$ at $\tilde{\mathbf{w}}^{(t)}$. However, $f(\tilde{\mathbf{w}}, i)$ is a nonsmooth function w.r.t. $\tilde{\mathbf{w}}$. So, we replace its gradient with the corresponding subgradient³ as follows:

$$\begin{aligned} \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} &= \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial g(\tilde{\mathbf{w}}, i, j)} \times \frac{\partial g(\tilde{\mathbf{w}}, i, j)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} \\ &= \sum_{j \in \mathbf{B}_i} \left(z_{ij}^{(t)} \times \frac{k}{k-1} \left(\sum_{r=1}^k \gamma_{ijr}^{(t)} \mathbf{B}_{ij(r)} - \frac{1}{k} \sum_{p=1}^k \mathbf{B}_{ij(p)} \right) \right) \end{aligned} \quad (9)$$

$$\text{where } z_{ij}^{(t)} = \begin{cases} 1, & \text{if } j = \arg \max_{j \in \mathbf{B}_i} g(\tilde{\mathbf{w}}^{(t)}, i, j) \\ 0, & \text{otherwise} \end{cases}, \text{ and } \gamma_{ijr}^{(t)} = \begin{cases} 1, & \text{if } r = \arg \max_{r \in \{1, 2, \dots, k\}} (\tilde{\mathbf{w}}^{(t)})^T \mathbf{B}_{ij(r)} \\ 0, & \text{otherwise.} \end{cases}$$

Then, we decompose $f(\tilde{\mathbf{w}}, i)$ at $\tilde{\mathbf{w}}^{(t)}$ as

$$\begin{aligned} f(\tilde{\mathbf{w}}, i) &= f(\tilde{\mathbf{w}}^{(t)}, i) + (\tilde{\mathbf{w}} - \tilde{\mathbf{w}}^{(t)})^T \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} \\ &= \tilde{\mathbf{w}}^T \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} + \frac{k}{k-1} \\ &\max_{j \in \mathbf{B}_i} \left(\max_u \left((\tilde{\mathbf{w}}^{(t)})^T \mathbf{B}_{ij(u)} - \text{mean}_v \left((\tilde{\mathbf{w}}^{(t)})^T \mathbf{B}_{ij(v)} \right) \right) - (\tilde{\mathbf{w}}^{(t)})^T \right. \\ &\quad \left. \times \sum_{j \in \mathbf{B}_i} \left(z_{ij}^{(t)} \times \frac{k}{k-1} \left(\sum_{r=1}^k \gamma_{ijr}^{(t)} \mathbf{B}_{ij(r)} - \frac{1}{k} \sum_{p=1}^k \mathbf{B}_{ij(p)} \right) \right) \right) \\ &= \tilde{\mathbf{w}}^T \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}}. \end{aligned} \quad (10)$$

So, for the t th CCCP iteration, by replacing $f(\tilde{\mathbf{w}}, i)$ in problem (8) with (10), this problem is transformed to

$$\begin{aligned} \min_{\tilde{\mathbf{w}}, \xi_i \geq 0} & \frac{1}{2} \|\tilde{\mathbf{w}}\|^2 + \frac{C}{n} \sum_i \xi_i \\ \text{s.t. } & i = 1, \dots, n, \tilde{\mathbf{w}}^T \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} \geq 1 - \xi_i \\ & \forall p, q \in \{1, 2, \dots, k\}, -l \\ & \leq \sum_{i=1}^n \sum_{j \in \mathbf{B}_i} \frac{1}{n_i} \tilde{\mathbf{w}}^T (\mathbf{B}_{ij(p)} - \mathbf{B}_{ij(q)}) \leq l. \end{aligned} \quad (11)$$

3) *Cutting Plane*: Directly solving the problem (11) as a quadratic programming (QP) problem is time consuming when the number of bags is large. So, we seek a method that would greatly accelerate the optimization procedure. This method was originally proposed in [31], and was shown to be effective and efficient for solving similar tasks. There are n slack variables ξ_i in problem (11). As in [31], to solve this

²We use the superscript t to denote that the result is obtained from the t th CCCP iteration, i.e., $\tilde{\mathbf{w}}^{(t)}$ is optimized from the t -th CCCP iteration step.

³For nonsmooth functions, at nonsmooth points, we cannot find its gradient as differentiable functions. At these point, the subgradient takes values from an interval that satisfies some specific constraints [32]. In practical use, we can simply choose one value from this interval.

problem efficiently, the 1-slack form of problem (11) is first derived. More precisely, we introduce a single slack variable ξ and rewrite the problem (11) as

$$\begin{aligned} \min_{\tilde{\mathbf{w}}, \xi \geq 0} & \frac{1}{2} \|\tilde{\mathbf{w}}\|^2 + C\xi \\ \text{s.t. } & \forall \mathbf{c} \in \{0, 1\}^n, \frac{1}{n} \tilde{\mathbf{w}}^T \sum_{i=1}^n \mathbf{c}_i \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} \\ & \geq \frac{1}{n} \sum_{i=1}^n \mathbf{c}_i - \xi \quad \forall p, q \in \{1, 2, \dots, k\}, \\ & -l \leq \sum_{i=1}^n \sum_{j \in \mathbf{B}_i} \frac{1}{n_i} \tilde{\mathbf{w}}^T (\mathbf{B}_{ij(p)} - \mathbf{B}_{ij(q)}) \leq l. \end{aligned} \quad (12)$$

It can be proved that the solution to problem (12) is identical to that of problem (11) with $\xi = (1/n) \sum_{i=1}^n \xi_i$ (similar to [31]).

Now the problem turns to how to solve (12) efficiently, which is convex, but has an exponential number of constraints because of the large number of feasible \mathbf{c} . To solve this problem, we employ an adaption of the cutting plane algorithm [33], whose main objective is to find a small subset of constraints Ω^t from the whole set of constrains $\{0, 1\}^n$ in (12) that guarantees an accurate solution. By using this method, a nested sequence of tighter relaxations can be constructed. More precisely, the first constraint is replaced by

$$\forall \mathbf{c} \in \Omega^t, \frac{1}{n} \tilde{\mathbf{w}}^T \sum_{i=1}^n \mathbf{c}_i \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} \geq \frac{1}{n} \sum_{i=1}^n \mathbf{c}_i - \xi. \quad (13)$$

A subset of constraints Ω^t with polynomial size can be generally found. With the constraints in Ω^t , the solution of the relaxed problem satisfies all the constraints from (12) up to a precision ϵ_2 , i.e., $\forall \mathbf{c} \in \{0, 1\}^n$

$$\frac{1}{n} \tilde{\mathbf{w}}^T \sum_{i=1}^n \mathbf{c}_i \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} \geq \frac{1}{n} \sum_{i=1}^n \mathbf{c}_i - (\xi + \epsilon_2). \quad (14)$$

The remaining exponential number of constraints will not be violated up to the precision ϵ_2 . Therefore, they do not need to be explicitly added to Ω^t . The algorithm constructs Ω^t in (13) iteratively. It starts with an empty set Ω^{t_0} as follows:

$$\begin{aligned} \min_{\tilde{\mathbf{w}}} & \frac{1}{2} \|\tilde{\mathbf{w}}\|^2 \\ \text{s.t. } & i = 1, \dots, n, \quad \forall p, q \in \{1, 2, \dots, k\} \\ & -l \leq \sum_{i=1}^n \sum_{j \in \mathbf{B}_i} \frac{1}{n_i} \tilde{\mathbf{w}}^T (\mathbf{B}_{ij(p)} - \mathbf{B}_{ij(q)}) \leq l. \end{aligned} \quad (15)$$

After solving this problem and getting the corresponding solution $\tilde{\mathbf{w}}^{(t_0)}$, the most violated constraint is computed as

$$\mathbf{c}_i^{t_0} = \begin{cases} 1, & \text{if } (\tilde{\mathbf{w}}^{(t_0)})^T \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t_0)}} \leq 1 \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

⁴Here, we denote t_i as the i th iteration of the cutting plane algorithm for solving the problem from the t th iteration of CCCP. And therefore, $\Omega^{t_0} = \emptyset$, $|\Omega^{t_s}| = s$, where $|\cdot|$ denote the number of vectors in Ω^{t_s} .

Then, this new constraint will be added to Ω^t and the optimization problem becomes

$$\begin{aligned} \min_{\tilde{\mathbf{w}}, \zeta \geq 0} \quad & \frac{1}{2} \|\tilde{\mathbf{w}}\|^2 + C\zeta \\ \text{s.t.} \quad & \forall \mathbf{c} \in \Omega^t, \frac{1}{n} \tilde{\mathbf{w}}^T \sum_{i=1}^n \mathbf{c}_i \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} \\ & \geq \frac{1}{n} \sum_{i=1}^n \mathbf{c}_i - \zeta \quad \forall p, q \in \{1, 2, \dots, k\}, -l \\ & \leq \sum_{i=1}^n \sum_{j \in \mathbf{B}_i} \frac{1}{n_i} \tilde{\mathbf{w}}^T (\mathbf{B}_{ij(p)} - \mathbf{B}_{ij(q)}) \leq l. \end{aligned} \quad (17)$$

For the current cutting plane step, in Ω^t there is only one element, which is obtained from (16). From this updated optimization problem, the optimal solution of $\tilde{\mathbf{w}}^{(t_1)}$ can be computed. Then, the most violated constraint vector $\mathbf{c}_i^{t_1}$ would be computed as in (16), where the only difference is that the weight vector $\tilde{\mathbf{w}}^{(t_0)}$ is replaced by $\tilde{\mathbf{w}}^{(t_1)}$. This procedure is repeated until all the constraints satisfy the requirements in (14). In this way, the successive strengthening approximation series of the problem (12) is constructed by the expanding number of cutting planes that cuts off the current optimal solution from the feasible set [33].

4) *Dual Form*: Problem (17) is a QP problem. In many cases, QP problems are solved from their dual forms. So, in this section we derive the dual form of problem (17) with t_1 replaced by t_s .

First of all, the following simplifications are made, we define a set of examples $\mathbf{X}^{t_s} = \{\mathbf{x}_1^{t_s}, \dots, \mathbf{x}_s^{t_s}, \mathbf{x}_{s+1}^{t_s}, \dots, \mathbf{x}_{s+2k^2}^{t_s}\}$. Here, $[\mathbf{x}_1^{t_s}, \dots, \mathbf{x}_s^{t_s}]$ are defined as $\mathbf{x}_j^{t_s} = (1/n) \sum_{i=1}^n \mathbf{c}_i^{t_s j} (\partial f(\tilde{\mathbf{w}}, i) / \partial \tilde{\mathbf{w}}) \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}}$, $j = 1, \dots, s$, $[\mathbf{x}_{s+1}^{t_s}, \dots, \mathbf{x}_{s+2k^2}^{t_s}]$ are defined as

$$\begin{aligned} \forall p, q \in \{1, 2, \dots, k\}, \\ \mathbf{x}_{(s+(p-1) \times k + q)}^{t_s} = - \sum_{i=1}^n \sum_{j \in \mathbf{B}_i} \frac{1}{ln_i} (\mathbf{B}_{ij(p)} - \mathbf{B}_{ij(q)}). \end{aligned} \quad (18)$$

$[\mathbf{x}_{s+k^2+1}^{t_s}, \dots, \mathbf{x}_{s+2k^2}^{t_s}]$ are defined as

$$\begin{aligned} \forall p, q \in \{1, 2, \dots, k\}, \\ \mathbf{x}_{(s+k^2+(p-1) \times k + q)}^{t_s} = \sum_{i=1}^n \sum_{j \in \mathbf{B}_i} \frac{1}{ln_i} (\mathbf{B}_{ij(p)} - \mathbf{B}_{ij(q)}). \end{aligned} \quad (19)$$

Then problem (17) can be transformed to

$$\begin{aligned} \min_{\tilde{\mathbf{w}}, \zeta \geq 0} \quad & \frac{1}{2} \|\tilde{\mathbf{w}}\|^2 + C\zeta \\ \text{s.t.} \quad & \forall j \in \{1, \dots, s\}, \quad \tilde{\mathbf{w}}^T \mathbf{x}_j^{t_s} \geq \frac{1}{n} (\mathbf{c}^{t_s j})^T \mathbf{1} - \zeta \\ & \forall j \in \{s+1, \dots, s+2k^2\}, \quad \tilde{\mathbf{w}}^T \mathbf{x}_j^{t_s} \geq -1. \end{aligned} \quad (20)$$

Proposition 1: The objective function of the dual problem of (20) is given by $D^{t_s}(\alpha^{t_s}) = -(1/2)(\alpha^{t_s})^T \mathbf{K}^{t_s} \alpha^{t_s} + (1/n) \sum_{i=1}^s \alpha_i^{t_s} (\mathbf{c}^{t_s i})^T \mathbf{1} - \sum_{i=s+1}^{s+2k^2} \alpha_i^{t_s}$, where $\mathbf{K}_{i,j}^{t_s} = \langle \mathbf{x}_i^{t_s}, \mathbf{x}_j^{t_s} \rangle$ and α^{t_s} is an s -dimensional vector. The corresponding dual optimization problem can then be formulated as

$$\max_{\alpha^{t_s} \geq 0} D^{t_s}(\alpha^{t_s}), \text{ s.t. } \sum_{i=1}^s \alpha_i^{t_s} \leq C.$$

TABLE I

Algorithm: M³IC-MBM	
Input:	1. bags $\{\mathbf{B}_1, \dots, \mathbf{B}_n\}$; 2. parameters: regularization constant C , CCCP solution precision ϵ_1 , cutting plane solution precision ϵ_2 , cluster number k , cluster size balance l
Output:	The cluster assignment γ
CCCP Iterations:	
1.	Construct $\tilde{\mathbf{B}} = \{\mathbf{B}_{ij(r)}\}$
2.	Initialize $\tilde{\mathbf{w}}^0, t=0, \Delta J = 10^{-3}, J^{-1} = 10^{-3}$
3.	while $\Delta J / J^{t-1} > \epsilon_1$ do
4.	Derive problem (17). Set the constraint set $\Omega = \phi, \forall 1 \leq i \leq n, \mathbf{c}_{j(i)}=0, s = -1$
Cutting Plane Iterations:	
5.	while H^{t_s} is true do
6.	$s = s + 1$
7.	Get $(\tilde{\mathbf{w}}^{(t_s)}, \zeta^{(t_s)})$ by solving (17) under Ω^{t_s}
8.	Compute the most violated bags, i.e., $\mathbf{c}_i^{t_s}$, by
	$\mathbf{c}_i^{t_s} = \begin{cases} 1, & \text{if } (\tilde{\mathbf{w}}^{(t_s)})^T \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big _{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} \leq 1 \\ 0, & \text{otherwise} \end{cases}$
	and update the constraint set Ω^{t_s} by $\Omega^{t_s+1} = \Omega^{t_s} \cup \mathbf{c}_i^{t_s}$.
9.	end while
10.	$t = t + 1$
11.	$\tilde{\mathbf{w}}^{(t)} = \tilde{\mathbf{w}}^{(t-1)s}$
12.	$\Delta J = J^{t-1} - J^t$
13.	end while
14. Cluster Assignment:	
	For bag \mathbf{B}_i , $y_i = \arg \max_p (\tilde{\mathbf{w}}^{(t)})^T \mathbf{B}_{ij^*(p)}$, where $j^* = \arg \max_{j \in \mathbf{B}_i} (\max_u (\tilde{\mathbf{w}}^{(t)})^T \mathbf{B}_{ij(u)} - \text{mean}_v ((\tilde{\mathbf{w}}^{(t)})^T \mathbf{B}_{ij(v)}))$.

Proof: Refer to Appendix A. ■

5) *Algorithm*: The proposed method is characterized by an outer iteration, i.e., CCCP iteration, and an inner iteration, i.e., cutting plane iteration. H^{t_s} is used to denote the constraint $(1/n)(\tilde{\mathbf{w}}^{(t_s)})^T \sum_{i=1}^n \mathbf{c}_i^{t_s} (\partial f(\tilde{\mathbf{w}}, i) / \partial \tilde{\mathbf{w}}) \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} \leq (1/n) \sum_{i=1}^n \mathbf{c}_i^{t_s} - (\zeta^{(t_s)} + \epsilon_2)$ and $J^t = (1/2) \|\tilde{\mathbf{w}}^{(t)}\|^2 + C\zeta^{(t)}$. The whole method is summarized in Table I.

E. Discussion

1) *Properties*: The outer iteration of our method is CCCP. It has already been shown that CCCP decreases the objective function monotonically and converges to a local minimum solution [30]. As for the inner iteration—the cutting plane iteration, we have the following three theorems. Theorem 3 justifies Step 8 of the proposed algorithm in Table I. Theorem 4 gives the calculation complexity for each cutting plane iteration. Theorem 5 gives an upper bound on the convergence rate.

Theorem 2: The most violated constraint for problem (17), with t_1 being replaced by t_s , is as follows:

$$\mathbf{c}_i^{t_s} = \begin{cases} 1, & \text{if } (\tilde{\mathbf{w}}^{(t_s)})^T \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}^{(t)}} \leq 1 \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

TABLE II
DETAILED DESCRIPTION OF THE DATASETS

Dataset	Categories	Features	Bags	Instances	Min. # Bags/Class	Max. # Bags/Class	Ave. # Bags/Class
Corel	3	230	300	1953	100	100	100
SIVAL1	5	30	300	9300	60	60	60
SIVAL2	5	30	300	9300	60	60	60
SIVAL3	5	30	300	9300	60	60	60
SIVAL4	5	30	300	9300	60	60	60
SIVAL5	5	30	300	9300	60	60	60
SIVAL6	5	30	300	9300	60	60	60
Natural Scene1	3	15	949	8541	268	341	316.3
Natural Scene2	3	15	935	8415	216	378	311.7
Natural Scene3	3	15	824	7416	216	340	274.7
Natural Scene4	5	15	1543	13887	216	378	308.6
Reuters1	3	243	1316	3726	81	798	438.7
Reuters2	4	243	385	1779	73	109	96.3
Reuters3	3	243	627	2171	81	437	209
Reuters4	3	243	640	2393	94	437	213.3
Reuters5	4	243	1061	3140	73	109	89.3
Reuters6	2	243	1235	3409	437	798	617.5

Proof: The most violated constraints are the ones that give the largest ζ . The maximal of ζ can be given by

$$\begin{aligned}
\zeta^* &= \max_{\mathbf{c}_i^{t_s} \in \{0,1\}} \left(\frac{1}{n} \sum_{i=1}^n \mathbf{c}_i^{t_s} - \frac{1}{n} (\tilde{\mathbf{w}}^{t_s})^T \sum_{i=1}^n \mathbf{c}_i^{t_s} \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}(t)} \right) \\
&= \frac{1}{n} \max_{\mathbf{c}_i^{t_s} \in \{0,1\}} \left(\sum_{i=1}^n \mathbf{c}_i^{t_s} (1 - (\tilde{\mathbf{w}}^{t_s})^T \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}(t)}) \right) \\
&= \frac{1}{n} \sum_{i=1}^n \max_{\mathbf{c}_i^{t_s} \in \{0,1\}} \left(\mathbf{c}_i^{t_s} (1 - (\tilde{\mathbf{w}}^{t_s})^T \frac{\partial f(\tilde{\mathbf{w}}, i)}{\partial \tilde{\mathbf{w}}} \Big|_{\tilde{\mathbf{w}}=\tilde{\mathbf{w}}(t)}) \right). \quad (22)
\end{aligned}$$

So, $\mathbf{c}_i^{t_s}$ should be calculated using (21). ■

Theorem 3: Each iteration from Step 5 to Step 9 in Table I takes time $O(en)$ for a constant constraint set Ω^{t_s} , where e is the average number of nonzero features of \mathbf{B}_{ij} and $e = d$ for nonsparse data.

Proof: Each inner product in Step 5 and Step 8 takes $O(ek)$ time. For each line, n inner products need to be calculated. So, in total it takes $O(ekn)$ time. As for Step 7, we can solve it through the dual form, setting up the dual form of problem (17) is dominated by computing $(s + 2k^2)^2$ inner products, and each inner product takes around $O(ekn)$ calculations after $\mathbf{x}_j^{t_s}$, $j \in \{s+1, \dots, s+2k^2\}$ are precomputed. In total, the cost time is $O((s + 2k^2)^2 ekn)$. Solving the dual requires time $O((s + 2k^2)^2)$. Therefore, each CCCP iteration takes time $O(en)$ for a constant constraint set Ω^{t_s} . ■

Theorem 4: The cutting plane iteration in Table I terminates after at most $\{(2/\epsilon_2), (8CR^2/\epsilon_2^2)\}$ steps, where $R^2 = (k/k - 1) \times \max_{ij} \|\mathbf{B}_{ij}\|^2$.

Proof: Refer to Appendix B. ■

Although the convergence of M³IC-MBM can be guaranteed, its outer iteration—CCCP iteration, only converges to a local minimal solution. Therefore, in this paper, during each trial the M³IC-MBM algorithm is run several times, and the solution with the minimal J^t value is chosen as the output. We

will show, in the experiments, that M³IC-MBM is very fast and that with only a few repetition times we can get good results.

2) *Relationship with Prior Work:* In [27], [28], and [34], the authors accelerate the MMC and semisupervised SVM for the traditional single instance learning. They first divide the original problem into a series of nonconvex subproblems by using cutting plane, then solve each nonconvex subproblem using CCCP iteratively. These methods have shown state-of-the-art performances, both in accuracy and efficiency, and they look similar to M³IC-MBM in this paper. But the main common problem in their methods is that the cutting plane approach is designed to solve convex problems rather than nonconvex problems. Since they try to solve a nonconvex problem by using cutting plane, the convergence and optimality of their methods may not be guaranteed. In both [27] and [28], the authors try to prove the convergence rate from the dual form. However, the constraints of their objective problem are not convex. In this case, the optimal solution of the original problem does not equal to that of the dual problem [35]. So, the proof of the convergence of the dual problem cannot be used to justify the convergence of the primal problem.

Different from their method, in M³IC-MBM we first apply the CCCP to decompose the original nonconvex problem into a series of convex ones, and then use the cutting plane method to solve each of them. In this way, the final solution can be guaranteed to converge to a local optimal value as proved in Appendix B. Therefore, M³IC-MBM is theoretically much more elegant than the previous related methods.

IV. EXPERIMENTS

A. Real-World Datasets

The used datasets are described below and in Table II.

1) *Corel:* Pictures from three categories of the Corel dataset, namely elephant, fox, and tiger, are merged together.

More specifically, we merge the positive bags from the benchmark datasets—elephant, fox, and tiger [1]. The reason why the negative bags in these datasets are not used is that the main objective of clustering task is to discover the hidden concepts/patterns in a dataset. But, in these datasets, the negative bags are just some background pictures and contain no common hidden concept/pattern.

2) *SIVAL*: There are in total 25 categories in the SIVAL dataset [3]. For each category, there are 60 images. We randomly partition these 25 categories into six groups, with each group containing 5 categories.

3) *Natural Scene*: This dataset was originally used to test the performances of multiple instance multiple label algorithms,⁵ where each bag can be associated with more than one labels. There are in total 2000 pictures in this dataset, associated with five labels. We choose, from this dataset, the 1543 pictures that are associated with only one label. We randomly partition these pictures into four groups.

4) *Reuters*: This dataset is from Reuters-21578 collection.⁶ and was also originally used for the multiple instance multiple label problem. In this dataset, there are in total 2000 bags and they are associated with seven labels. We use the same strategy as we employed on the Natural Scene dataset, and delete the bags that are associated with more than one label. There are in total such 1701 bags left. We further partition them into six sub-datasets.

B. Experimental Setups

We have conducted comprehensive performance evaluations by testing our method and comparing it with the only existing MIC method—BAMIC [10], and two other single instance learning methods, i.e., K -means [36] and CPM3C [27].

For BAMIC, we used the three bag distance measurement methods as in [10], i.e., minimal Hausdorff distance, maximal Hausdorff distance, and average Hausdorff distance. We name the BAMIC methods with these three bag distance measurements as BAMIC1, BAMIC2, and BAMIC3, respectively. BAMIC is in fact based on K -medoids, but does not provide a way to avoid the local minimal problem. So, for each dataset, we run each of these BAMIC algorithms 10 times independently, and report *only the best performance* of these 10 independent runs.

For M³IC-MBM, we set $\epsilon_1 = 0.01$ and $\epsilon_2 = 0.01$. The class imbalance parameter l is set by grid search from the grid $[0, 0.001, 0.01, 0.1, 1 : 1 : 5, 10]$ and the parameter C is searched from the exponential grid $2^{[-4:1:4]}$. $\tilde{\mathbf{w}}^0$ is randomly initialized. To avoid the local minimal problem that we mentioned in Section III-E, for each experiment we run the M³IC algorithm five times independently and report the final result *with the minimal J^l* in Table I.

For K -means and CPM3C, they are not designed to solve MIC, but can be adapted to solve it. More specifically, we first cluster all of the instances in bags by using these two methods

separately. (For K -means, the result is summarized over 50 independent runs.) Then, for each bag, the cluster assignment is determined by the cluster assignment that appears most frequently on the instances of this bag. These two methods are used as the traditional single instance learning opponents.

CPM3C is related to the proposed method. But as mentioned in Section III-E.2, one of the significant differences between CPM3C and the proposed method is that the outer iteration of CPM3C is the cutting plane and its inner iteration is the CCCP iteration. However, when using the cutting plane method to solve a nonconvex problem, the convergence property may not hold. So, in order to avoid the infinite cutting plane loop, the maximum number of cutting plane loops is set to 20. The set of parameters for CPM3C are exactly the same as that of the proposed method. Therefore, in this paper, we employed the same way to tune the parameters of CPM3C as the one used in the proposed method.

C. Evaluation

In experiments, the number of clusters k is set to the true number of classes of these datasets. The following evaluation criteria are employed:

1) *Clustering Accuracy (Acc)*: The Acc [24], [27], [28] is used to evaluate the final clustering performance. Specifically, we first take a set of labeled bags, remove the labels of these bags and run the clustering algorithms, and then relabel these bags using the clustering assignments returned by the algorithms. Finally, we measure the percentage of correct classifications by comparing the true labels and the labels assigned by the clustering algorithms as $Acc = (1/n) \sum_{i=1}^n \delta(y_i, map(c_i))$, where $map(\cdot)$ is a function that maps each cluster index to a class label, which can be found by the Hungarian algorithm [37]. c_i and y_i are the cluster index of \mathbf{x}_i and the true class label. $\delta(a, b)$ is a function which is equal to 1 when a equals b , and 0 otherwise. It is clear that Acc discovers the one-to-one relationship between clusters, and measures the extent to which cluster assignments are associated with the corresponding true categories. The greater the clustering accuracy, the better the performance of the clustering algorithm.

2) *Normalized Mutual Information (NMI)*: Another evaluation metric we employed here is the NMI [38], which was originally a symmetric measure to quantify the statistical information shared between two distributions. More precisely, for two random variables X and Y , NMI is defined as $NMI(X, Y) = (I(X, Y) / \sqrt{H(X)H(Y)})$. Here, $I(X, Y)$ is mutual information between X and Y , while $H(X)$ and $H(Y)$ are entropies of X and Y . If there is a one-to-one correspondence between the distribution of X and that of Y , NMI equals one. Given two clustering results, we can consider the clustering assignments as distributions of random variables, and NMI can be estimated as $NMI = \frac{\sum_{p=1}^k \sum_{q=1}^k n_{p,q} \log(\frac{n_{p,q}}{n_p \hat{n}_q})}{\sqrt{(\sum_{p=1}^k n_p \log \frac{n_p}{n})(\sum_{q=1}^k \hat{n}_q \log \frac{\hat{n}_q}{n})}}$, where n_p refers to the number of data contained in the p th cluster, \hat{n}_q is the number of data belonging to the q th cluster, and $n_{p,q}$ denotes the number of data that are in the intersection between the p th and the q th clusters.

⁵It can be Available at <http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/annex/miml-image-data.htm>.

⁶It can be Available at <http://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/annex/miml-text-data.htm>.

TABLE III
CLUSTERING ACCURACY (%) COMPARISONS (THE BEST RESULTS IN EACH DATASET ARE HIGHLIGHTED IN BOLDFACE)

	M ³ IC-MBM	BAMIC1	BAMIC2	BAMIC3	<i>K</i> -means	CPM3C
Core1	54.0	40.3	47.3	36.7	52.0	49.7
SIVAL1	47.0	26.3	35.3	38.0	26.0	23.3
SIVAL2	42.0	29.0	31.7	39.3	29.0	28.3
SIVAL3	41.0	30.0	35.7	38.7	27.0	29.7
SIVAL4	39.0	26.0	32.7	30.0	25.3	26.0
SIVAL5	40.7	25.7	36.3	34.3	26.0	28.0
SIVAL6	44.7	25.7	31.7	24.7	24.0	23.3
Natural Scene1	56.6	39.7	33.5	45.0	40.2	39.4
Natural Scene2	63.2	38.1	52.5	44.9	50.5	49.4
Natural Scene3	60.4	55.1	46.2	42.1	40.4	45.6
Natural Scene4	41.7	33.4	32.4	38.9	29.3	37.8
Reuters1	77.0	66.8	49.5	62.7	60.7	75.1
Reuters2	69.9	35.1	63.6	40.8	28.3	38.4
Reuters3	61.1	55.7	38.6	58.9	69.7	69.7
Reuters4	61.9	36.4	44.5	49.4	68.2	68.2
Reuters5	54.2	42.3	32.8	47.6	30.5	31.1
Reuters6	89.7	67.9	70.3	68.9	65.7	85.9

TABLE IV
NMI COMPARISONS (THE BEST RESULTS IN EACH DATASET ARE HIGHLIGHTED IN BOLDFACE)

	M ³ IC-MBM	BAMIC1	BAMIC2	BAMIC3	<i>K</i> -means	CPM3C
Core1	0.159	0.027	0.092	0.024	0.154	0.012
SIVAL1	0.178	0.015	0.107	0.048	0.030	0.024
SIVAL2	0.166	0.023	0.075	0.048	0.087	0.107
SIVAL3	0.266	0.015	0.105	0.088	0.097	0.092
SIVAL4	0.137	0.006	0.079	0.027	0.023	0.028
SIVAL5	0.162	0.011	0.037	0.023	0.023	0.038
SIVAL6	0.120	0.018	0.119	0.025	0.018	0.076
Natural Scene1	0.151	0.020	0.018	0.009	0.025	0.024
Natural Scene2	0.156	0.046	0.063	0.092	0.065	0.067
Natural Scene3	0.239	0.023	0.033	0.191	0.094	0.092
Natural Scene4	0.136	0.059	0.073	0.135	0.083	0.114
Reuters1	0.302	0.115	0.226	0.248	0.239	0.385
Reuters2	0.369	0.050	0.100	0.217	0.046	0.052
Reuters3	0.199	0.011	0.116	0.189	0.000	0.000
Reuters4	0.296	0.078	0.211	0.115	0.000	0.000
Reuters5	0.301	0.173	0.192	0.198	0.228	0.245
Reuters6	0.530	0.279	0.229	0.288	0.259	0.499

3) *CPU Running Time*: All of our experiments are conducted on MATLAB r2008a on a 2.5-GHz Intel CoreTM2 Duo PC running Windows Vista with 2.0 GB main memory. The average CPU times of different algorithms are also used as evaluation.

D. Clustering Results

From Tables III and IV, it is easy to tell that, in most cases, M³IC-MBM works much better than the other methods. From Table V, it is clear that our method runs much faster than BAMIC and CPM3C. Within most of these 17 datasets, the running time of M³IC-MBM is comparable to that of *K*-means. However, it is also clear that, among the datasets where *K*-means runs faster, the clustering performance of

K-means is much worse than M³IC-MBM. So, it is safe to say, in these datasets, the proposed method is both effective and efficient than the comparison methods. From Reuters3 and Reuters4 results reported in Table III, it seems that *K*-means performs better than the proposed method. But as we can see from Table IV, the corresponding NMI values of *K*-means are all 0, which means *K*-means gives the same cluster assignments to all of the bags. In traditional single instance clustering problems, *K*-means will not give the same cluster assignments to all of the instances. But in our experiments of multiple instance clustering, the cluster assignments of bags depend on that of the associated instances. In this case, it is possible that all of the bags are grouped into the same cluster by *K*-means, since it is performed on all of the instances, rather than directly on bags. The high accuracies of *K*-means on these two datasets

TABLE V
CPU RUNNING TIME (IN SECONDS, THE BEST RESULTS IN EACH DATASET ARE HIGHLIGHTED IN BOLDFACE)

	M ³ IC-MBM	BAMIC1	BAMIC2	BAMIC3	<i>K</i> -means	CPM3C
Corel	1.2	267.8	257.1	261.6	6.1	137.6
SIVAL1	1.8	95.5	96.1	92.5	7.0	87.6
SIVAL2	3.1	95.1	98.4	95.8	5.6	108.3
SIVAL3	2.7	93.3	100.4	95.7	5.0	93.1
SIVAL4	2.9	107.7	100.5	94.8	7.7	86.9
SIVAL5	3.2	95.1	117.2	106.0	7.1	95.7
SIVAL6	6.7	69.7	70.5	71.2	8.3	69.6
Natural Scene1	11.4	68.3	69.2	68.3	2.2	49.0
Natural Scene2	17.1	63.9	66.4	66.6	2.6	138.2
Natural Scene3	13.2	49.3	51.2	51.5	2.0	52.8
Natural Scene4	30.6	290.0	316.2	320.6	4.1	80.3
Reuters1	13.9	42.0	44.3	42.8	17.0	198.8
Reuters2	3.5	7.4	7.4	7.4	3.9	419.9
Reuters3	3.3	8.8	10.7	9.2	6.8	218.4
Reuters4	3.4	10.2	10.9	11.0	6.1	211.2
Reuters5	13.8	19.6	21.8	21.4	8.1	148.5
Reuters6	8.2	65.8	74.1	68.5	7.2	54.2

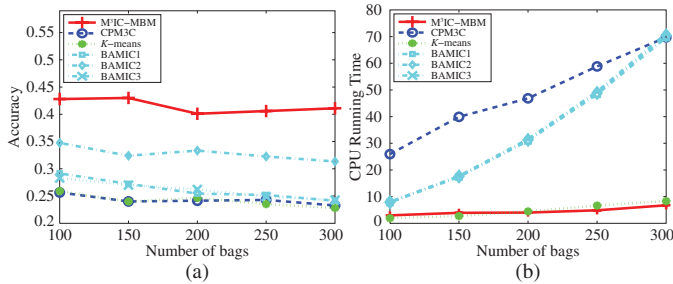


Fig. 1. Comparison of clustering accuracy and the CPU running time (in seconds) with varying dataset sizes on SIVAL. (a) Clustering accuracy. (b) CPU time.

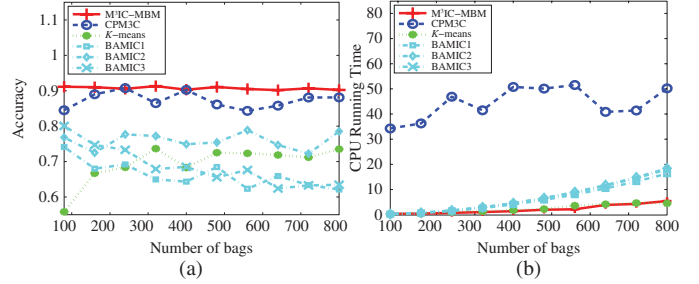


Fig. 2. Comparison of clustering accuracy and the CPU running time (in seconds) with varying dataset sizes on Reuters. (a) Clustering accuracy. (b) CPU time.

are in fact brought in by the extremely imbalanced nature of different categories in the corresponding datasets.

The reason for the good performance of M³IC-MBM is that it can incorporate the nature of multiple instance setting into the clustering problem more naturally. So, it gives a better clustering performance than the other methods. And in our algorithm, the outer iteration—CCCP iteration, as well as the inner iteration—cutting plane iteration, converges very fast. So, the proposed method is very efficient.

For BAMIC, it is an adaption of *K*-medoids. It tries to integrate the MIL into the clustering problem. The motivation is good. But this method tries to define the distances between bags in an unsupervised way, which brings the ambiguous problem into the clustering. The most important “concept” in each bag is not identified. So, in these datasets, its performances are not quite promising. And since it needs to calculate the distances between instances in different bags many times, its speed will be badly affected.

As for *K*-means, it is relatively fast in these datasets because it is an iterative method, and during each iteration it only needs to calculate the distances between the instances and the clustering centers. But it is not designed

for MLC, and therefore cannot take the multiple instance assumptions into account. This is the main reason why *K*-means performs worse than M³IC-MBM.

CPM3C is both an effective and efficient method in some single instance learning datasets, as shown in [27]. However, it is not suitable to be adapted to solve the MIC problems. Its potential problem, i.e., the convergence problem, is very evident in these real-world multiple instance datasets, which validates our comments in Section III-E.2 very well. And its performance in clustering accuracies and NMI are not quite promising in these datasets either.

E. Properties

1) *Scalability*: First of all, the performance of the proposed method with different dataset sizes is studied here. For each trial, the same number of bags from each category is randomly chosen, and then the corresponding performances of different algorithms are measured. This procedure is repeated 100 times and the average results are reported in Figs. 1–3.

From the scalability comparison results, it can be seen that, with different dataset sizes, the clustering accuracies are relatively stable, and are much better than those of the

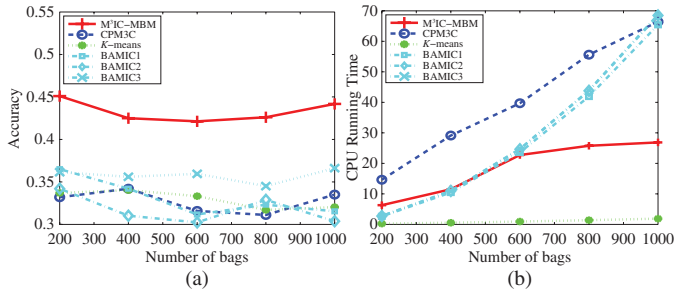


Fig. 3. Comparison of clustering accuracy and the CPU running time (in seconds) with varying dataset sizes on Natural Scene. (a) Clustering accuracy. (b) CPU time.

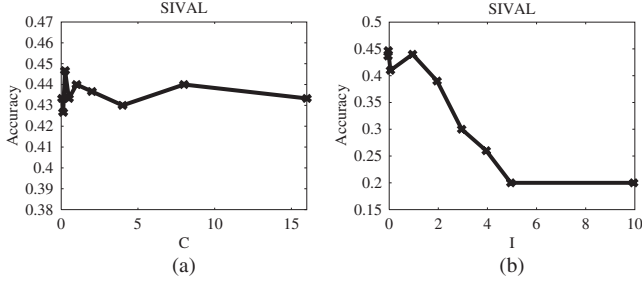


Fig. 4. (Left) Sensitivity of the parameter C on SIVAL. (Right) Sensitivity of the parameter l on SIVAL6. (a) Sensitivity on C . (b) Sensitivity on l .

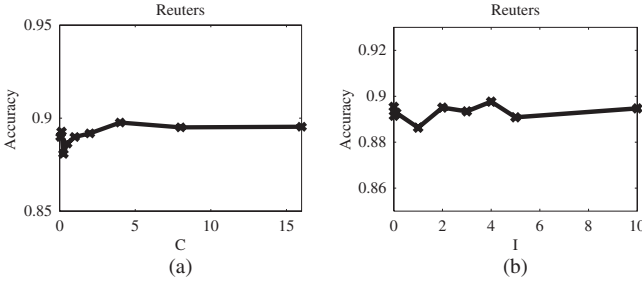


Fig. 5. (Left) Sensitivity of the parameter C on Natural Scene. (Right) Sensitivity of the parameter l on Reuters6. (a) Sensitivity on C . (b) Sensitivity on l .

comparison methods. As for the CPU running time, with the increase of the bags, the CPU running time of BAMIC increases almost exponentially with the increase of bags, while that of M^3IC -MBM and K -means are relatively stable. And in these datasets, in most cases, the CPU running time of K -means is shorter than that of M^3IC -MBM. But it is evident that the clustering accuracy performance of K -means is much worse than that of M^3IC -MBM. For CPM3C, it is clear that its clustering accuracy is good compared to the other methods except M^3IC -MBM. However, its CPU running time is too long, which may hinder its practical use in this kind of applications.

2) *Sensitivity*: When conducting experiments on M^3IC -MBM, C and l are set by grid search. In this section, we will study the impact of these two parameters on the proposed method.

For each dataset, each time, we will only fix one parameter while tuning the other one. In this way, the parameter sensitivity experiments of C and l are reported in Figs. 4–6. It is clear that, with the increase of the parameter C , the performance

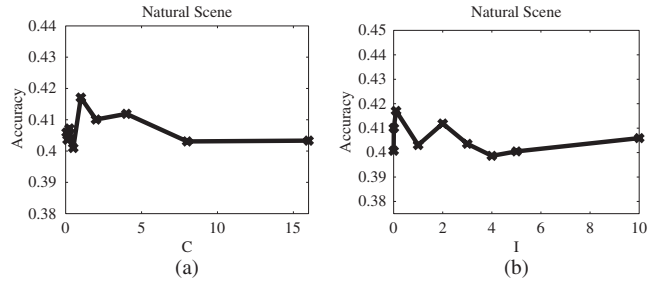


Fig. 6. (Left) Sensitivity of the parameter C on Natural Scene. (Right) Sensitivity of the parameter l on Natural Scene4. (a) Sensitivity on C . (b) Sensitivity on l .

of M^3IC -MBM is relatively stable. As for the experimental results on l , in SIVAL the performance of the proposed method deteriorates very quickly after l becomes larger than 2, while in Reuters and Natural Scene, its impacts are relatively small. So, in practice, we suggest using a relatively small l .

V. CONCLUSION

In this paper, we formulated a novel M^3IC problem for the MIC problem, which can be directly used to solve applications such as image and text clustering. In order to avoid solving a nonconvex problem directly, we relaxed the original problem. Then, a combination of CCCP and the cutting plane method— M^3IC -MBM, was used to solve the relaxed problem. After that, some important properties of the proposed method were also demonstrated and proved. In the experimental part, we compared the proposed method with the existing method—BAMIC, and two other single-instance clustering methods on several real-world image and text datasets. The comparison results are very promising.

ACKNOWLEDGMENT

The authors would like to thank the reviewers and editors for their constructive suggestions.

APPENDIX A

PROOF OF THE DUAL PROBLEM

Proposition 2: The dual problem of (20) is given by

$$\begin{aligned} \max_{\alpha^{ts} \geq 0} & -\frac{1}{2} \alpha^{ts T} \mathbf{K}^{ts} \alpha^{ts} + \frac{1}{n} \sum_{i=1}^s \alpha_i^{ts} (\mathbf{c}^i)^T \mathbf{1} - \sum_{i=2^n+1}^{s+2K^2} \alpha_i^{ts} \\ \text{s.t.} & \sum_{i=1}^s \alpha_i^{ts} \leq C \end{aligned}$$

where $\mathbf{K}_{i,j}^{ts} = \langle \mathbf{x}_i^{ts}, \mathbf{x}_j^{ts} \rangle$, and α^{ts} is a s -dimensional vector.

Proof: The Lagrangian form of problem (20) is

$$\begin{aligned} L(\tilde{\mathbf{w}}, \zeta, \eta, \alpha^{ts}) &= \frac{1}{2} \|\tilde{\mathbf{w}}\|^2 + C\zeta \\ &+ \sum_{j=1}^s \alpha_j^{ts} \left(-\zeta - \tilde{\mathbf{w}} \mathbf{x}_j^{ts} + \frac{1}{n} (\mathbf{c}^j)^T \mathbf{1} \right) \\ &+ \sum_{j=s+1}^{s+2K^2} \alpha_j^{ts} (-1 - \tilde{\mathbf{w}} \mathbf{x}_j^{ts}) - \eta \zeta \end{aligned}$$

where $\alpha^{t_s} \geq 0$ and $\eta > 0$. Differentiating this Lagrangian form with respect to $\tilde{\mathbf{w}}$ and ζ , we can get

$$\frac{\partial L(\tilde{\mathbf{w}}, \zeta, \eta, \alpha^{t_s})}{\partial \tilde{\mathbf{w}}} = \tilde{\mathbf{w}} - \sum_{i=1}^{s+2k^2} \alpha_i^{t_s} \mathbf{x}_i^{t_s},$$

$$\frac{\partial L(\tilde{\mathbf{w}}, \zeta, \eta, \alpha^{t_s})}{\partial \zeta} = C - \sum_{i=1}^s \alpha_i^{t_s} - \eta.$$

Setting these derivatives to zeros, the optimal solution for $\tilde{\mathbf{w}}$ and ζ are required as follows:

$$\tilde{\mathbf{w}} = \sum_{i=1}^{s+2k^2} \alpha_i^{t_s} \mathbf{x}_i^{t_s}, \quad \sum_{i=1}^s \alpha_i^{t_s} = C - \eta. \quad (23)$$

Plugging (23) into the Lagrangian, we obtain the dual problem

$$\max_{\alpha^{t_s} \geq 0} -\frac{1}{2} \alpha^{t_s T} \mathbf{K}^{t_s} \alpha^{t_s} + \frac{1}{n} \sum_{i=1}^s \alpha_i^{t_s} (\mathbf{c}^{t_i})^T \mathbf{1} - \sum_{i=s+1}^{s+2k^2} \alpha_i^{t_s}$$

s.t. $\sum_{i=1}^s \alpha_i^{t_s} \leq C.$ (24)

APPENDIX B

PROOF OF THE CONVERGENCE RATE

Theorem 5: The cutting plane iteration in Table I terminates after at most $\{(2/\epsilon_2), (8CR^2/\epsilon_2^2)\}$ steps, where $R^2 = (k/k-1) \times \max_{ij} \|\mathbf{B}_{ij}\|^2$.

Proof: The objective function value of problem (20) is upper bounded by C , since $\mathbf{w} = 0$ and $\zeta = 1$ is a feasible point. Next, we will show, during each cutting plane iteration, the objective function will increase by at least a constant value.

Let \mathbf{c}^{s+1} be the newly added constraint. Let $\hat{\alpha}^{t_s}$ be the optimal solution of the dual problem before this constraint is added. Suppose $\tilde{\alpha}^{t_s} = [\hat{\alpha}_1^{t_s}, \dots, \hat{\alpha}_s^{t_s}, 0, \hat{\alpha}_{s+1}^{t_s}, \dots, \hat{\alpha}_{s+2k^2}^{t_s}]$. To lower bound the progress made by each cutting plane iteration, we consider the increase in a specific direction of the dual function by line search, which can be described as follows:

$$\max_{0 \leq \beta \leq C} D^{t_s+1}(\tilde{\alpha}^{t_s} + \beta \eta) - D^{t_s+1}(\tilde{\alpha}^{t_s}) \quad (25)$$

where $D^{t_s+1}(\tilde{\alpha}) = -(1/2) \tilde{\alpha}^T \mathbf{K}^{t_s+1} \tilde{\alpha} + (1/n) \sum_{i=1}^{s+1} \tilde{\alpha}_i (\mathbf{c}^{t_i})^T \mathbf{1} - \sum_{i=s+1}^{s+1+2k^2} \tilde{\alpha}_i$, and the direction η is specified as

$$\eta_j = \begin{cases} -\frac{1}{C} \tilde{\alpha}_j^{t_s}, & j = 1, \dots, s \\ 1, & j = s+1 \\ 0, & j = s+2, \dots, s+1+2k^2. \end{cases} \quad (26)$$

Using this way to construct η , $\tilde{\alpha}^{t_s} + \beta \eta$ is guaranteed to lie in the feasible region of the dual. Then, Theorem 6, which has been proved in [39], is employed. In our formulation, the following equations hold:

$$\frac{\partial D^{t_s+1}(\tilde{\alpha}^{t_s})}{\partial \tilde{\alpha}_j^{t_s}} = \frac{1}{n} (\mathbf{c}^{t_j})^T \mathbf{1} - (\tilde{\mathbf{w}}^{t_s})^T \mathbf{x}_j^{t_s+1} = \zeta, \quad \text{if } \tilde{\alpha}_j^{t_s} \neq 0$$

$$j = s+2, \dots, s+1+2k^2$$

$$\frac{\partial D^{t_s+1}(\tilde{\alpha}^{t_s})}{\partial \tilde{\alpha}_j^{t_s}} = -1 - (\tilde{\mathbf{w}}^{t_s})^T \mathbf{x}_j^{t_s+1} = 0, \quad \text{if } \tilde{\alpha}_j^{t_s} \neq 0.$$

These two equations can be directly derived from the KKT conditions for solving a convex optimization problem [35].

For the newly added constraint

$$\frac{\partial \tilde{D}^{t_s+1}(\tilde{\alpha}^{t_s})}{\partial \tilde{\alpha}_{s+1}^{t_s}} = \frac{1}{n} (\mathbf{c}^{t_j})^T \mathbf{1} - (\tilde{\mathbf{w}}^{t_s})^T \mathbf{x}_j^{t_s+1} \geq \zeta + \epsilon_2. \quad (27)$$

Then, we can get

$$\nabla D^{t_s+1}(\tilde{\alpha}^{t_s})^T \eta \geq \zeta + \epsilon_2 - \sum_{j=1}^s \frac{\tilde{\alpha}_j^{t_s}}{C} \zeta \geq \epsilon_2 \geq 0 \quad (28)$$

$$\eta^T \mathbf{K}^{t_s+1} \eta = \left(\mathbf{x}_{s+1}^{t_s+1} \right)^2 - \frac{2}{C} \sum_{i=1}^s \tilde{\alpha}_i^{t_s} \mathbf{K}^{t_s+1} \left(\mathbf{x}_i^{t_s+1}, \mathbf{x}_{s+1}^{t_s+1} \right)$$

$$+ \frac{1}{C^2} \sum_{i=1}^s \sum_{j=1}^s \tilde{\alpha}_i^{t_s} \tilde{\alpha}_j^{t_s} \mathbf{K}^{t_s+1} \left(\mathbf{x}_i^{t_s+1}, \mathbf{x}_j^{t_s+1} \right)$$

$$\leq R^2 + 2R^2 + \frac{1}{C^2} C^2 R^2 = 4R^2. \quad (29)$$

By applying [28, Cor. 12], the minimum increase per cutting plane iteration is $\max_{0 \leq \beta \leq C} D^{t_s+1}(\tilde{\alpha}^{t_s} + \beta \eta) - D^{t_s+1}(\tilde{\alpha}^{t_s}) \geq \min\{C\epsilon_2/2, \epsilon_2^2/8R^2\}$. It is clear that this increase is independent of s and t . So, the cutting plane algorithm will terminate after at most $\max\{2/\epsilon_2, 8CR^2/\epsilon_2^2\}$ iterations. ■

REFERENCES

- [1] S. Andrews, I. Tsochantaris, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003, pp. 561–568.
- [2] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, nos. 1–2, pp. 31–71, Jan. 1998.
- [3] R. Rahmani and S. A. Goldman, "MISSL: Multiple-instance semi-supervised learning," in *Proc. Int. Conf. Mach. Learn.*, vol. 10. 2006, pp. 705–712.
- [4] R. Amar, D. R. Dooley, S. A. Goldman, and Q. Zhang, "Multiple-instance learning of real-valued data," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 3–10.
- [5] S. Ray and D. Page, "Multiple instance regression," in *Proc. Int. Conf. Mach. Learn.*, 2001, pp. 425–432.
- [6] Q. Zhang and S. A. Goldman, "EM-DD: An improved multiple-instance learning technique," in *Proc. Adv. Neural Inf. Process. Syst.*, 1998, pp. 1073–1080.
- [7] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ: Prentice-Hall, 1988.
- [8] J. Han and M. Kamber, *Data Mining*. San Mateo, CA: Morgan Kaufmann, 2001.
- [9] C. Manning and P. Raghavan, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
- [10] M. L. Zhang and Z. H. Zhou, "Multi-instance clustering with applications to multi-instance prediction," *Appl. Intell.*, vol. 31, no. 1, pp. 47–68, Aug. 2009.
- [11] J. Wang and J. D. Zucker, "Solving the multiple-instance problem: A lazy learning approach," in *Proc. Int. Conf. Mach. Learn.*, 2000, pp. 1119–1126.
- [12] D. Zhang, F. Wang, L. Si, and T. Li, "M³IC: Maximum margin multiple instance clustering," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1339–1344.
- [13] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 10. 1998, pp. 570–576.

- [14] O. Maron and A. L. Ratan, "Multiple-instance learning for natural scene classification," in *Proc. Int. Conf. Mach. Learn.*, 1998, pp. 341–349.
- [15] R. Rahmani, S. A. Goldman, H. Zhang, J. Krettek, and J. E. Fritts, "Localized content based image retrieval," in *Proc. Int. Workshop Multimedia Inf. Ret.*, 2005, pp. 227–236.
- [16] B. Scholkopf and A. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2002.
- [17] P. Gehler and O. Chapelle, "Deterministic annealing for multiple-instance learning," in *Proc. 11th Int. Conf. Artif. Intell. Stat.*, 2007, pp. 123–130.
- [18] T. Gartner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels," in *Proc. Int. Conf. Mach. Learn.*, 2002, pp. 179–186.
- [19] J. T. Kwok and P.-M. Cheung, "Marginalized multi-instance kernels," in *Proc. 19th Int. Joint Conf. Artif. Intell.*, 2007, pp. 901–906.
- [20] C. B. Razvan and R. J. Mooney, "Multiple instance learning for sparse positive bags," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 112–119.
- [21] Y. Chen and J. Z. Wang, "Image categorization by learning and reasoning with regions," *J. Mach. Learn. Res.*, vol. 5, pp. 913–939, Aug. 2004.
- [22] Y. Chen, J. Bi, and J. Z. Wang, "MILES: Multiple-instance learning via embedded instance selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 1931–1947, Dec. 2006.
- [23] Q. Tao, D. Chu, and J. Wang, "Recursive support vector machines for dimensionality reduction," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 189–193, Jan. 2008.
- [24] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1537–1544.
- [25] L. Xu and D. Schuurmans, "Unsupervised and semi-supervised multi-class support vector machines," in *Proc. Nat. Conf. Artif. Intell.*, 2005, pp. 904–910.
- [26] K. Zhang, I. W. Tsang, and J. T. Kwok, "Maximum margin clustering made practical," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 583–596, Apr. 2009.
- [27] B. Zhao, F. Wang, and C. Zhang, "Efficient maximum margin clustering via cutting plane algorithm," in *Proc. SIAM Int. Conf. Data Mining*, 2008, pp. 751–762.
- [28] B. Zhao, F. Wang, and C. Zhang, "Efficient multiclass maximum margin clustering," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 751–762.
- [29] F. Wang, B. Zhao, and C. Zhang, "Linear time maximum margin clustering," *IEEE Trans. Neural Netw.*, vol. 21, no. 2, pp. 319–332, Feb. 2010.
- [30] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann, "Kernel methods for missing variables," in *Proc. Int. Workshop Artif. Intell. Stat.*, 2005, pp. 325–332.
- [31] T. Joachims, "Training linear SVMs in linear time," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 217–226.
- [32] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*. New York: Springer-Verlag, 2006.
- [33] J. E. Kelley, "The cutting-plane method for solving convex programs," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 4, pp. 703–712, Dec. 1960.
- [34] B. Zhao, F. Wang, and C. Zhang, "Cuts3vm: A fast semi-supervised SVM algorithm," in *Proc. 14th Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 830–838.
- [35] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [36] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2001.
- [37] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. New York: Dover, 1998.
- [38] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Dec. 2002.
- [39] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, pp. 1453–1484, Sep. 2005.



Dan Zhang received the Bachelor's degree from the Department of Computer Science, Purdue University, West Lafayette, IN, and the Master's degree from the Department of Automation, Tsinghua University, Beijing, China, in 2008.

He is now with the Department of Computer Science, Purdue University. His current research interests include information retrieval, machine learning, data mining, and pattern recognition.



Fei Wang received the Ph.D. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2008.

He spent one year in the School of Computing and Information Science, Florida International University, Miami, as a Post-Doctoral Research Associate, and one year in the Department of Statistical Science, Cornell University, Ithaca, NY, as a Post-Doctoral Research Associate. He is currently with the Healthcare Transformation Group, IBM T. J. Watson Research Center, Hawthorne, NY. He has

published over 70 papers in his fields of expertise. His current research interests include machine learning, data mining, pattern recognition, and medical informatics.



Luo Si received the Ph.D. degree from Carnegie Mellon University, Pittsburgh, PA, in 2006.

He is an Assistant Professor in the Computer Science and Statistics Department (by courtesy), Purdue University, West Lafayette, IN. He has published more than 60 research papers in his fields of expertise. His current research interests include information retrieval, machine learning techniques and applications, and text mining techniques for different applications.

Dr. Si is an Associate Editor of the Association for Computing Machinery *Transactions on Information Systems*, and a member of the Editorial Board of *Information Processing and Management*. He is a recipient of the National Science Foundation Career Award in 2008.



Tao Li received the Ph.D. degree in computer science from the University of Rochester, Rochester, NY, in 2004.

He is an Associate Professor in the School of Computer Science, Florida International University, Miami. His current research interests include machine learning, data mining, and pattern recognition.