M^{3}IC: Maximum Margin Multiple Instance Clustering*

Dan Zhang[1], **Fei Wang**[2], **Luo Si**[3], **Tao Li**[4]
[1,3] Department of Computer Science,
Purdue University, West Lafayette, IN, 47906
[2,4] School of Computing & Information Sciences,
Florida International University, Miami, FL, 33199
[1,3]{zhang168, lsi} @cs.purdue.edu, [2,4]{feiwang, taoli}@cs.fiu.edu

Abstract

Clustering, classification, and regression, are three major research topics in machine learning. So far, much work has been conducted in solving multiple instance classification and multiple instance regression problems, where supervised training patterns are given as *bags* and each bag consists of some *instances*. But the research on unsupervised multiple instance clustering is still limited . This paper formulates a novel *Maximum Margin Multiple Instance Clustering* (*M^{3}IC*) problem for the multiple instance clustering task. To avoid solving a non-convex optimization problem directly, M^{3}IC is further relaxed, which enables an efficient optimization solution with a combination of *Constrained Concave-Convex Procedure* (*CCCP*) and the *Cutting Plane* method. Furthermore, this paper analyzes some important properties of the proposed method and the relationship between the proposed method and some other related ones. An extensive set of empirical results demonstrate the advantages of the proposed method against existing research for both effectiveness and efficiency.

1 Introduction

Multiple instance learning (*MIL*) can be viewed as a variation of the 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. In a binary multiple instance classification problem, the labels are assigned to bags, rather than instances. A typical assumption for this kind of problem 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. Then, using MIL methods, we can devise a classifier based on the labeled bags and predict the labels for the unlabeled ones. So far, MIL has been widely used in areas such as text mining [Andrews *et al.*, 2003], drug design [Dietterich *et al.*, 1998], Localized Content Based Image Retrieval (LCBIR) [Rahmani and Goldman, 2006], etc.

As another branch of machine learning, clustering [Jain and Dubes, 1988] is one of the most fundamental research topics in both data mining and machine learning. It aims to divide data into groups of similar objects, i.e., clusters. From a machine learning perspective, what clustering does is to learn the hidden patterns of the dataset in an unsupervised way, and these patterns are usually referred to as data concepts. From a practical perspective, clustering plays an outstanding role in data mining applications such as information retrieval, text mining, Web analysis, marketing, computational biology, and many others [Han and Kamber, 2001].

However, so far, almost all of the clustering methods are designed to solve traditional single instance learning problems, while in many cases clustering can be better formulated as MIL problems. For example, in image clustering, there is natural ambiguity that as to what portion of each image contains the common 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, where each instance in this bag corresponds to a region in this image. Then, this application requires the solution of *Multiple Instance Clustering* (*MIC*) to help users to partition these bags. Besides Image Clustering, MIC methods can also be applied to many other applications such as drug molecules clustering [Zhang and Zhou, in press] and text clustering, etc.

Recently, very limited research addresses the task of MIC. In [Zhang and Zhou, in press], the authors regard bags as atomic data items and use some distance metric to measure the distances between bags. Then they adapt the k-medoids algorithm to cluster bags. Their method is efficient in some applications. But, as claimed by [Rahmani and Goldman, 2006], defining distances between bags in an unsupervised way may not reflect their actual content differences. For example, two pictures may share identical background and only differ in that one contains a tiger and the other contains a fox. By using the minimal Hausdorff distance to measure distances between bags [Wang and Zucker, 2000], the distance between these two pictures will be very low even though their actual contents (or concepts) may differ. And the calculation of the distances between bags is quite time consuming, since it always needs to calculate all the distances between instances in different bags.

*The work of Dan Zhang and Luo Si is supported by NSF research grant IIS-0746830 and the work of Fei Wang and Tao Li is partially supported by NSF grants IIS-0546280, DMS-0844513 and CCF-0830659.

In this paper, we solve the MIC problem in a different way. We first formulate a novel *Maximum Margin Multiple Instance Clustering* (*M^{3}IC*) problem based on *Maximum Margin Clustering* (*MMC*) [Xu *et al.*, 2005]. The new formulation aims at finding desired hyperplanes that maximize the margin differences on at least one instance per bag in a unsupervised way. But the formulation of M^{3}IC is a non-convex optimization problem, and we can not solve it directly. Therefore, we relax the original M^{3}IC problem and propose a method – M^{3}IC-MBM, which is a combination of *Constrained Concave-Convex Procedure* (*CCCP*) and *Cutting Plane* methods, to solve the relaxed optimization task.

The rest of the paper is organized as follows: Section 2, introduces the MIC problem. Section 3 formulates the novel M^{3}IC problem and propose an efficient method to solve it. Section 4 presents the experimental results. Section 5 concludes and points some future research directions.

2 Problem Statement

Suppose we are given a set of n bags, $\{\mathbf{B}_i, i = 1, 2, \cdots, n\}$. The instances in the bag $\mathbf{B}_i$ are denoted as $\{\mathbf{B}_{i1}, \mathbf{B}_{i2}, ..., \mathbf{B}_{in_i}\}$, where n_i is the total number of instances in this bag. The goal is to partition this given dataset into k clusters, such that the concepts in different clusters can be "distinct" from each other. We use a $1 \times n$ vector $\mathbf{f}$ to denote the cluster assignment array, with $\mathbf{f}_i$ being the cluster assignment for bag $\mathbf{B}_i$.

3 The Proposed Method

3.1 Formulation

First of all, we formulate the M^{3}IC problem. For each class $p \in \{1, \cdots, k\}$, we define a weight vector $\mathbf{w}_p$. Instead of labeling all the samples by running an SVM implicitly over all possible labels as that in MMC [Xu *et al.*, 2005], in M^{3}IC, we try to find a labeling on bags that results in several large margin classifiers that maximize margins on bags, and it is natural to define the *Bag Margin* (*BM*) for a bag $\mathbf{B}_i$ as:

$$\max_{j \in \mathbf{B}_i} (\mathbf{w}_{u_{ij}^*}^T \mathbf{B}_{ij} - \mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij}) \tag{1}$$

where, $u_{ij}^* = \arg\max_p (\mathbf{w}_p^T \mathbf{B}_{ij})$, and $v_{ij}^* = \arg\max_{p \setminus u_{ij}^*} (\mathbf{w}_p^T \mathbf{B}_{ij})$[1]. It is obvious that BM is determined by the most "discriminative " instance. With this definition, M^{3}IC can then be formulated as:

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

[1]Throughout this paper, "\" means ruling out. So, this definition can also be written as: $v_{ij}^* = \arg\max_{p \neq u_{ij}^*} (\mathbf{w}_p^T \mathbf{B}_{ij})$

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 to avoid the trivially "optimal" solutions [Xu *et al.*, 2005]. It is clear that, in this formulation, these two 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 (2), the cluster assignment of a specific bag $\mathbf{B}_i$ can be determined by $\mathbf{f}_i = \arg\max_p \sum_{j \in \mathbf{B}_i} I_{ij} \mathbf{w}_p^T \mathbf{B}_{ij}$.

3.2 M^{3}IC-MBM

However, the optimization problem (2) is difficult to solve. For the first constraint, i.e., $\max_{j \in \mathbf{B}_i} (\mathbf{w}_{u_{ij}^*}^T \mathbf{B}_{ij} - \mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij}) \geq 1 - \xi_i$, the convexity of $\mathbf{w}_{v_{ij}^*}^T \mathbf{B}_{ij}$ is unknown, which makes the form of the constraint too complicated. For the second constraint, the indication function I_{ij} also makes this constraint non-convex. In this section, we first relax these two constraints, and then propose an efficient method – *M^{3}IC-MBM* to solve the resulting optimization problem.

Relaxation

To relax the first constraint, we consider introducing the notion of *Modified Bag Margin* (*MBM*). For a bag $\mathbf{B}_i$, *MBM* is defined as:

$$\max_{j \in \mathbf{B}_i} (\max_u \mathbf{w}_u^T \mathbf{B}_{ij} - mean_{v \setminus u_{ij}^*} (\mathbf{w}_v^T \mathbf{B}_{ij})) \tag{3}$$

Here, the "$mean$" function calculates the average value of the input function with respect to the subscript variable. Replacing BM with MBM, the first constraint in problem (2) turns to: $\max_{j \in \mathbf{B}_i} (\max_u \mathbf{w}_u^T \mathbf{B}_{ij} - mean_{v \setminus u_{ij}^*} (\mathbf{w}_v^T \mathbf{B}_{ij})) \geq 1 - \xi_i$. This is equivalent to $\frac{k}{k-1} \max_{j \in B_i} (\max_u \mathbf{w}_u^T \mathbf{B}_{ij} - mean_v (\mathbf{w}_v^T \mathbf{B}_{ij})) \geq 1 - \xi_i$.

For the second constraint in problem (2), we relax the indication function I_{ij}, and rewrite this constraint as follows:

$$\begin{aligned} & \forall p, q \in \{1, 2, \cdots, k\} \\ -l \leq \sum_{i=1}^{n} \sum_{j \in \mathbf{B}_i} & \frac{1}{n_i} \mathbf{w}_p^T \mathbf{B}_{ij} - \sum_{i=1}^{n} \sum_{j \in \mathbf{B}_i} \frac{1}{n_i} \mathbf{w}_q^T \mathbf{B}_{ij} \leq l \end{aligned} \tag{4}$$

Without loss of generality, we introduce two *concatenated* vectors as:

$$\begin{aligned} \widetilde{\mathbf{w}} &= [\mathbf{w}_1^T, \mathbf{w}_2^T, \cdots, \mathbf{w}_p^T, \cdots, \mathbf{w}_k^T]^T \\ \mathbf{B}_{ij(p)} &= [\mathbf{0}, \mathbf{0}, \cdots, \mathbf{B}_{ij}^T, \cdots, \mathbf{0}]^T \end{aligned} \tag{5}$$

Here, $\mathbf{0}$ is a $1 \times d$ zero vector, where 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 $\mathbf{B}_{ij}$. Then, we have $\widetilde{\mathbf{w}}^T \mathbf{B}_{ij(p)} = \mathbf{w}_p^T \mathbf{B}_{ij}$.

With the relaxation of the two constraints in Eq. (3), Eq. (4), and the introduction of the two *concatenated* vectors in

Eq.(5), problem (2) can be transformed to:

$$\begin{aligned}\min_{\widetilde{\mathbf{w}},\xi_i\geq 0}\ & \frac{1}{2}\|\widetilde{\mathbf{w}}\|^2+\frac{C}{n}\sum_i\xi_i \\ s.t.\ & i=1,\dots,n, \\ & \frac{k}{k-1}\max_{j\in\mathbf{B}_i}(\max_u \widetilde{\mathbf{w}}^T\mathbf{B}_{ij(u)}-mean_v(\widetilde{\mathbf{w}}^T\mathbf{B}_{ij(v)})) \\ & \geq 1-\xi_i \\ & \forall p,q\in\{1,2,\cdots,k\} \\ & -l\leq\sum_{i=1}^{n}\sum_{j\in\mathbf{B}_i}\frac{1}{n_i}\widetilde{\mathbf{w}}^T(\mathbf{B}_{ij(p)}-\mathbf{B}_{ij(q)})\leq l\end{aligned} \tag{6}$$

In the following sections, we will propose a method, ***M^{3}IC with Modified Bag Margin*** (***M^{3}IC-MBM***), to solve this relaxed problem (6).

CCCP Decomposition

Although the objective function and the second constraint in problem (6) are smooth and convex, the first constraint is not. Fortunately, the constrained concave-convex procedure (CCCP) is just designed to solve the optimization problems with a concave convex objective function with concave convex constraints [Smola *et al.*, 2005]. Next, we will show how to use CCCP to solve the problem (6).

To simplify the notation, let $f(\widetilde{\mathbf{w}},i)$ be $\frac{k}{k-1}\max_{j\in\mathbf{B}_i} g(\widetilde{\mathbf{w}},i,j)$ and $g(\widetilde{\mathbf{w}},i,j)$ be $\max_u \widetilde{\mathbf{w}}^T\mathbf{B}_{ij(u)}-mean_v(\widetilde{\mathbf{w}}^T\mathbf{B}_{ij(v)})$. Then, the first constraint in problem (6) becomes: $f(\widetilde{\mathbf{w}},i)\geq 1-\xi_i$. It is obvious that this constraint is, although not convex, the difference of two convex functions.

Hence, we can solve problem (6) with CCCP. Given an initial point $\widetilde{\mathbf{w}}^{(0)}$, CCCP iteratively computes $\widetilde{\mathbf{w}}^{(t+1)}$ from $\widetilde{\mathbf{w}}^{(t)}$ [2] by replacing $f(\widetilde{\mathbf{w}},i)$ with its first order Taylor expansions at $\widetilde{\mathbf{w}}^{(t)}$, and solving the resulting quadratic programming problem, until convergence.

Therefore, in order to use CCCP, we should first calculate the gradient and the first-order Taylor expansion of $f(\widetilde{\mathbf{w}},i)$ at $\widetilde{\mathbf{w}}^{(t)}$. But $f(\widetilde{\mathbf{w}},i)$ is a non-smooth functions w.r.t. $\widetilde{\mathbf{w}}$. So, we replace its gradient with its subgradient as follows:

$$\begin{aligned}&\frac{\partial f(\widetilde{\mathbf{w}},i)}{\partial\widetilde{\mathbf{w}}}\mid_{\widetilde{\mathbf{w}}=\widetilde{\mathbf{w}}^{(t)}} \\ =&\frac{\partial f(\widetilde{\mathbf{w}},i)}{\partial g(\widetilde{\mathbf{w}},i,j)}\times\frac{\partial g(\widetilde{\mathbf{w}},i,j)}{\partial\widetilde{\mathbf{w}}}\mid_{\widetilde{\mathbf{w}}=\widetilde{\mathbf{w}}^{(t)}} \\ =&\sum_{j\in\mathbf{B}_i}\left(z_{ij}^{(t)}\times\frac{k}{k-1}(\sum_{r=1}^{k}\gamma_{ijr}^{(t)}\mathbf{B}_{ij(r)}-1/k\sum_{p=1}^{k}\mathbf{B}_{ij(p)})\right)\end{aligned} \tag{7}$$

Here,

$$z_{ij}^{(t)}=\begin{cases}1, & \text{if } j=\arg\max_{j\in\mathbf{B}_i} g(\widetilde{\mathbf{w}}^{(t)},i,j) \\ 0, & \text{otherwise}\end{cases} \tag{8}$$

[2] We use the superscript t to denote that the result is obtained from the t-th CCCP iteration. For example, $\widetilde{\mathbf{w}}^{(t)}$ is the optimized weight vector from the t-th CCCP iteration step.

and

$$\gamma_{ijr}^{(t)}=\begin{cases}1, & \text{if } r=\arg\max_{r\in\{1,2,\cdots,k\}}(\widetilde{\mathbf{w}}^{(t)})^T\mathbf{B}_{ij(r)} \\ 0, & \text{otherwise}\end{cases} \tag{9}$$

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

$$\begin{aligned}&f(\widetilde{\mathbf{w}},i) \\ =&f(\widetilde{\mathbf{w}}^{(t)},i)+(\widetilde{\mathbf{w}}-\widetilde{\mathbf{w}}^{(t)})^T\frac{\partial f(\widetilde{\mathbf{w}},i)}{\partial\widetilde{\mathbf{w}}}\mid_{\widetilde{\mathbf{w}}=\widetilde{\mathbf{w}}^{(t)}} \\ =&\widetilde{\mathbf{w}}^T\frac{\partial f(\widetilde{\mathbf{w}},i)}{\partial\widetilde{\mathbf{w}}}\mid_{\widetilde{\mathbf{w}}=\widetilde{\mathbf{w}}^{(t)}}+ \\ &\frac{k}{k-1}\max_{j\in\mathbf{B}_i}\left(\max_u\left((\widetilde{\mathbf{w}}^{(t)})^T\mathbf{B}_{ij(u)}\right)-mean_v\left((\widetilde{\mathbf{w}}^{(t)})^T\mathbf{B}_{ij(v)}\right)\right) \\ &-(\widetilde{\mathbf{w}}^{(t)})^T\times \\ &\sum_{j\in\mathbf{B}_i}\left(z_{ij}^{(t)}\times\frac{k}{k-1}(\sum_{r=1}^{k}\gamma_{ijr}^{(t)}\mathbf{B}_{ij(r)}-1/k\sum_{p=1}^{k}\mathbf{B}_{ij(p)})\right) \\ =&\widetilde{\mathbf{w}}^T\frac{\partial f(\widetilde{\mathbf{w}},i)}{\partial\widetilde{\mathbf{w}}}\mid_{\widetilde{\mathbf{w}}=\widetilde{\mathbf{w}}^{(t)}}\end{aligned} \tag{10}$$

Thus, for the t-th CCCP iteration, by replacing $f(\widetilde{\mathbf{w}},i)$ with Eq. (10) in problem (6), we obtain the following optimization problem:

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

Cutting Plane

It is true that, for each CCCP iteration step, we can solve problem (11) directly as a quadratic programming problem. But instead of directly solving this optimization problem, we employ the Cutting Plane method, which has shown its effectiveness and efficiency in solving similar tasks recently [Joachims, 2006]. In problem (11), we have n slack variables ξ_i. To solve it efficiently, we first derive the 1-slack form of problem (11) as in [Joachims, 2006]. More specifically, we introduce a signle slack variable $\xi\geq 0$ and rewrite the problem (11) as:

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

Algorithm: M^{3}IC-MBM

Input:

1. bags $\{\mathbf{B}_1, ..., \mathbf{B}_n\}$
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 $\mathbf{f}$

CCCP Iterations:

1. Construct $\widetilde{\mathcal{B}} = \{\mathbf{B}_{ij(r)}\}$
2. Initialize $\widetilde{\mathbf{w}}^0$,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 $\Omega = \phi$,$\forall 1 \leq i \leq n$, c_i=0, $s = -1$

Cutting Plane Iterations:

5. **while** H^{t_s} is true **do**
6. $s = s + 1$
7. Get $(\widetilde{\mathbf{w}}^{(t_s)}, \xi^{(t_s)})$ by solving (17) under Ω
8. Compute the most violated bags, i.e., $c_i^{t_s}$, by

$$c_i^{t_s} = \begin{cases} 1, & if \ \ (\widetilde{\mathbf{w}}^{(t_s)})^T \dfrac{\partial f(\widetilde{\mathbf{w}}, i)}{\partial \widetilde{\mathbf{w}}} \mid_{\widetilde{\mathbf{w}}=\widetilde{\mathbf{w}}^{(t)}} \leq 1 \\ 0, & otherwise \end{cases}$$

and update the constraint set Ω by $\Omega = \Omega \bigcup \mathbf{c}^{t_s}$.

9. **end while**
10. $t = t + 1$
11. $\widetilde{\mathbf{w}}^{(t)} = \widetilde{\mathbf{w}}^{(t-1)_s}$
12. $\Delta J = J^{t-1} - J^t$
13. **end while**
14. **Cluster Assignment**:

For bag $\mathbf{B}_i$, $\mathbf{f}_i = \arg\max_p (\widetilde{\mathbf{w}}^{(t)})^T \mathbf{B}_{ij^*(p)}$, where $j^* = \arg\max_{j \in \mathbf{B}_i} (\max_u (\widetilde{\mathbf{w}}^{(t)})^T \mathbf{B}_{ij(u)} - mean_v ((\widetilde{\mathbf{w}}^{(t)})^T \mathbf{B}_{ij(v)}))$

Table 1: **Algorithm**: M^{3}IC-MBM

It can be proved that the solution to problem (12) is identical to problem (11) with $\xi = \frac{1}{n} \sum_{i=1}^n \xi_i$ (similar to [Joachims, 2006]).

Now the problem becomes how to solve problem (12) efficiently, which is convex and has 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 [Kelley, 1960], which is intended to find a small subset of constraints Ω from the whole set of constrains $\{0,1\}^n$ in problem (12) that guarantees a sufficiently accurate solution. Using this algorithm, we can construct a nested sequence of tighter relaxations. Specifically, in this algorithm, the first constraint is replaced by:

$$\begin{aligned} & i = 1, \ldots, n, \forall \mathbf{c} \in \Omega \\ & \frac{1}{n} \widetilde{\mathbf{w}}^T \sum_{i=1}^n c_i \frac{\partial f(\widetilde{\mathbf{w}}, i)}{\partial \widetilde{\mathbf{w}}} \mid_{\widetilde{\mathbf{w}}=\widetilde{\mathbf{w}}^{(t)}} \geq \frac{1}{n} \sum_{i=1}^n c_i - \xi \end{aligned} \quad (13)$$

Similar to [Joachims, 2006], we can generally find a polynomially sized subset of constraints Ω , with which the solution of the relaxed problem satisfies all the constraints from problem (12) up to a precision ϵ_2, i.e., $\forall \mathbf{c} \in \{0,1\}^n$:

$$\frac{1}{n} \widetilde{\mathbf{w}}^T \sum_{i=1}^n c_i \frac{\partial f(\widetilde{\mathbf{w}}, i)}{\partial \widetilde{\mathbf{w}}} \mid_{\widetilde{\mathbf{w}}=\widetilde{\mathbf{w}}^{(t)}} \geq \frac{1}{n} \sum_{i=1}^n c_i - (\xi + \epsilon_2) \quad (14)$$

This means, the remaining exponential number of constraints will not be violated up to the precision ϵ_2. Therefore, we don't need to explicitly add them to Ω.

The algorithm iteratively constructs Ω in Eq.(13). The algorithm starts with an empty set of constraints Ω. Specifically, it starts with the following problem:

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

After getting the solution $\widetilde{\mathbf{w}}^{(t_0)}$[3] of the above problem, the most violated constraint can be computed as:

$$c_i^{t_0} = \begin{cases} 1, & if \ \ (\widetilde{\mathbf{w}}^{(t_0)})^T \dfrac{\partial f(\widetilde{\mathbf{w}}, i)}{\partial \widetilde{\mathbf{w}}} \mid_{\widetilde{\mathbf{w}}=\widetilde{\mathbf{w}}^{(t)}} \leq 1 \\ 0, & otherwise \end{cases} \quad (16)$$

Then, this constraint will be added to Ω and the optimization problem turns to:

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

Please note that, for the current cutting plane step, in Ω, there is only one n-dimensional vector, which is obtained from Eq.(16). From this updated optimization problem, we can get the solution $\widetilde{\mathbf{w}}^{(t_1)}$. Then, the most violated constraint $c_i^{t_1}$ can be computed similarly as in Eq.(16). The only difference is that the weight vector $\widetilde{\mathbf{w}}^{(t_0)}$ is replaced by $\widetilde{\mathbf{w}}^{(t_1)}$. This procedure is repeated until all the constraints satisfy the requirement in Eq.(14). In this way, a successive strengthening approximation series of the problem (12) can be constructed by the expanding number of cutting planes that cut off the current optimal solution from the feasible set [Kelley, 1960].

The Whole Method

Our method is characterized by an outer iteration, i.e., CCCP iteration and an inner iteration, i.e., Cutting Plane iteration. We use H^{t_s} to denote the constraint $\frac{1}{n}(\widetilde{\mathbf{w}}^{(t_s)})^T \sum_{i=1}^n c_i^{t_s} \frac{\partial f(\widetilde{\mathbf{w}}, i)}{\partial \widetilde{\mathbf{w}}} \mid_{\widetilde{\mathbf{w}}=\widetilde{\mathbf{w}}^{(t)}} \geq \frac{1}{n} \sum_{i=1}^n c_i^{t_s} - (\xi^{(t_s)} + \epsilon_2)$ and $J^t = \frac{1}{2} \|\widetilde{\mathbf{w}}^{(t)}\|^2 + C\xi^{(t)}$. Then, the whole method is summarized in Table 1.

[3]Here, 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.

3.3 Discussion

Convergence and Local Minimal

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 [Smola *et al.*, 2005]. As for the inner iteration – the Cutting Plane iteration, we have the following two theorems:

Theorem 1: Each iteration from step 5 to step 9 in Table 1 takes time $O(ekn)$ for a constant constraint set Ω, where e is the average number of nonzero features of $\mathbf{B}_{ij}$ and $e = d$ for non-sparse data.

Theorem 2: The Cutting Plane iteration in Table 1 terminates after at most $\{\frac{2}{\epsilon_2}, \frac{8CR^2}{\epsilon_2^2}\}$ steps, where $R^2 = \frac{k}{k-1}\max_{ij}\|\mathbf{B}_{ij}\|^2$

The proofs of these two theorems are similar to the proofs in [Joachims, 2006], and therefore are omitted here.

Although the convergence of M^{3}IC-MBM can be guaranteed, it is true that its outer iteration – CCCP iteration only converges to a local minimum solution. Therefore, we would expect a way to get a better solution. In this paper, we run the M^{3}IC-MBM algorithm several times, and choose the solution with the minimal J^t value. We will show, in the experiments, M^{3}IC-MBM is pretty fast and with only a few repetition times, we can get good results.

Relationship

In [Zhao *et al.*, 2008a], [Zhao *et al.*, 2008b] and [Zhao *et al.*, 2008c], the authors accelerate the MMC and Semi-Supervised SVM for the traditional single instance learning problems. They first divide the original problem into a series of non-convex sub-problems by using Cutting Plane, then solve each non-convex sub-problem using CCCP iteratively. These methods have shown state-of-the-art performances, both in accuracy and efficiency, and they look similar to M^{3}IC-MBM in this paper. But the main common problem in their methods is that the Cutting Plane approach is designed to solve convex nonsmooth problems, rather than non-convex problems. Since, they try to solve a non-convex problem by using cutting plane, the convergence and optimality of their methods may not be guaranteed. Different from their method, in M^{3}IC-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. Therefore, M^{3}IC-MBM is theoretically more elegant than the previous related methods.

Dataset	Categories	Features	Bags	Instances
Corel	3	230	300	1953
SIVAL1	5	30	300	9300
SIVAL2	5	30	300	9300
SIVAL3	5	30	300	9300
SIVAL4	5	30	300	9300
SIVAL5	5	30	300	9300

Table 2: The detailed description of the datasets

4 Experiments

In this section, we will present a set of experiments to validate the effectiveness and the efficiency of the proposed method. All the experiments are performed with MATLAB r2008a on a 2.5GHZ Intel Core^{TM}2 Duo PC running Windows Vista with 2.0GB main memory.

4.1 Datasets

Currently, there is no benchmark dataset for MIC algorithms. Fortunately, we can utilize several available datasets for multiple instance classifications, and make them eligible for the MIC tasks. Although MUSK datasets, i.e., MUSK1 and MUSK2 [Dietterich *et al.*, 1998], are two most popular datasets, we can not use them here, because there is only one potential concept – musk in these two datasets, while we need at least two concepts to measure the clustering performances.

Corel We merge pictures from three categories of the Corel dataset, namely elephant, fox, and tiger. More specifically, we merge the positive bags from the benchmark datasets – elephant, fox, and tiger [Andrews *et al.*, 2003]. 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 may contain no common hidden concept/pattern. Then, the detailed description of this combined dataset is summarized in Table 2.

SIVAL There are in total 25 categories in the SIVAL dataset [Rahmani and Goldman, 2006]. For each category, there are 60 images. We randomly partition these 25 categories into 5 groups, with each group containing 5 categories. We name the five groups as SIVAL1,SIVAL2,SIVAL3,SIVAL4, and SIVAL5. The descriptions of these datasets are summarized in Table 2.

4.2 Experimental Setups and Comparisons

We have conducted comprehensive performance evaluations by testing our method and comparing it with BAMIC [Zhang and Zhou, in press].

For BAMIC, we used the three bag distance measurement methods as in [Zhang and Zhou, in press], 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. 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 M^{3}IC-MBM, we set $\epsilon_1 = 0.01$, $\epsilon_2 = 0.01$, 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 search from the exponential grid $2^{[-4:1:4]}$. $\widetilde{\mathbf{w}}^0$ is randomly initialized. To avoid the local minimal problem that we have mentioned in Section 3.3, for each experiment, we run the M^{3}IC algorithm 5 times independently and report the final result *with the minimal* J^t in Table 1.

In experiments, we set the number of clusters k to the true number of classes for all clustering algorithms. Then, we use the clustering accuracy to evaluate the final clustering performance as in [Valizadegan and Jin, 2006][Xu *et al.*,

2005][Zhao *et al.*, 2008b][Zhao *et al.*, 2008c]. Specifically, we first take a set of labeled bags, remove the labels of these bags and run the clustering algorithms, then we 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 given by the clustering algorithms. The average CPU-times for each independent run of these algorithms are also reported here.

4.3 Clustering Results

The clustering accuracies for different algorithms are reported in Table 3, while the average CPU time of all the independent runs for these algorithms is reported in Table 4.

From Table 3, it is easy to tell that M^{3}IC-MBM works much better than the BAMIC method. From Table 4, it is clear that our method runs much faster than BAMIC. This is because, in our algorithm, the outer iteration–CCCP iteration, as well as the inner iteration–Cutting Plane iteation, converges very fast. But for BAMIC, since it needs to calculate the distances between instances in different bags many times, its speed will be badly affected.

	M^{3}IC-MBM	BAMIC1	BAMIC2	BAMIC3
Corel	**54.0**	40.3	47.3	36.7
SIVAL1	**47.0**	26.3	35.3	38.0
SIVAL2	**42.0**	29.0	31.7	39.3
SIVAL3	**41.0**	30.0	35.7	38.7
SIVAL4	**39.0**	26.0	32.7	30.0
SIVAL5	**40.7**	25.7	36.3	34.3

Table 3: Clustering accuracy (%) comparisons

	M^{3}IC-MBM	BAMIC1	BAMIC2	BAMIC3
Corel	**1.2**	267.8	257.1	261.6
SIVAL1	**1.8**	95.5	96.1	92.5
SIVAL2	**3.1**	95.1	98.4	95.8
SIVAL3	**2.7**	93.3	100.4	95.7
SIVAL4	**2.9**	107.7	100.5	94.8
SIVAL5	**3.2**	95.1	117.2	106.0

Table 4: CPU Running Time (in seconds)

5 Conclusions

In this paper, we formulate a novel M^{3}IC problem for the multiple instance clustering task. In order to avoid solving a non-convex problem directly, we relax the original problem. Then, a combination of *Constrained Concave-Convex Procedure* (*CCCP*) and the *Cutting Plane* method – M^{3}IC-MBM is proposed to solve the relaxed problem. After that, we demonstrate some important properties of the proposed method. In the experiment part, we compare our method with the existed method–BAMIC on several real-world datasets. However, it is true that under some special cases, a bag may belong to more than one clusters. But in our algorithm, we can only assign a bag to one cluster. In the future, we will consider how to deal with this problem.

References

[Andrews *et al.*, 2003] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pages 561–568, Cambridge, MA: MIT Press., 2003.

[Dietterich *et al.*, 1998] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. In *Artificial Inteligence*, pages 1–8, 1998.

[Han and Kamber, 2001] Jiawei Han and Micheline Kamber. *Data Mining*. Morgan Kaufmann Publishers, 2001.

[Jain and Dubes, 1988] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[Joachims, 2006] T. Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM New York, NY, USA, 2006.

[Kelley, 1960] JE Kelley. The cutting plane method for solving convex programs. *Journal of the SIAM*, 8(4):703–712, 1960.

[Rahmani and Goldman, 2006] R. Rahmani and S. A. Goldman. Missl: Multiple-instance semi-supervised learning. In *International Conference on Machine Learning*, volume 10, pages 705–712, Pittsburgh, PA., 2006.

[Smola *et al.*, 2005] A.J. Smola, SVN Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

[Valizadegan and Jin, 2006] H. Valizadegan and R. Jin. Generalized Maximum Margin Clustering and Unsupervised Kernel Learning. *Advances in Neural Information Processing Systems*, 2006.

[Wang and Zucker, 2000] J. Wang and J.D. Zucker. Solving the Multiple-Instance Problem: A Lazy Learning Approach. In *International Conference on Machine Learning*, 2000.

[Xu *et al.*, 2005] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. *Advances in Neural Information Processing Systems*, 17:1537–1544, 2005.

[Zhang and Zhou, in press] M.-L. Zhang and Z.-H. Zhou. Multi-instance clustering with applications to multi-instance prediction. *Applied Intelligence*, in press.

[Zhao *et al.*, 2008a] B. Zhao, F. Wang, and C. Zhang. Cuts3vm: a fast semi-supervised svm algorithm. 2008.

[Zhao *et al.*, 2008b] B. Zhao, F. Wang, and C. Zhang. Efficient maximum margin clustering via cutting plane algorithm. *In The 8th SIAM International Conference on Data Mining*, pages 751–762, 2008.

[Zhao *et al.*, 2008c] B. Zhao, F. Wang, and C. Zhang. Efficient multiclass maximum margin clustering. *The 25th International Conference on Machine Learning*, pages 751–762, 2008.