Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 2186 – 2201
© TÜBİTAK
doi:10.3906/elk-2009-144

A linear programming approach to multiple instance learning

Emel Şeyma KÜÇÜKAŞCI[1,*], Mustafa Gökçe BAYDOĞAN[2], Z. Caner TAŞKIN[2]
[1]Department of Industrial Engineering, Faculty of Engineering, İstanbul Commerce University, İstanbul, Turkey
[2]Department of Industrial Engineering, Faculty of Engineering, Boğaziçi University, İstanbul, Turkey

Received: 21.09.2020 • **Accepted/Published Online:** 08.12.2020 • **Final Version:** 26.07.2021

Abstract: Multiple instance learning (MIL) aims to classify objects with complex structures and covers a wide range of real-world data mining applications. In MIL, objects are represented by a bag of instances instead of a single instance, and class labels are provided only for the bags. Some of the earlier MIL methods focus on solving MIL problem under the standard MIL assumption, which requires at least one positive instance in positive bags and all remaining instances are negative. This study proposes a linear programming framework to learn instance level contributions to bag label without emposing the standart assumption. Each instance of a bag is mapped to a pseudo-class membership estimate and these estimates are aggregated to obtain the bag-level class membership in an optimization framework. A simple linear mapping enables handling various MIL assumptions with adjusting instance contributions. Our experiments with instance-dissimilarity based data representations verify the effectiveness of the proposed MIL framework. Proposed mathematical models can be solved efficiently in polynomial time.

Key words: Multiple instance learning, classification, linear programming, optimization

1. Introduction

Multiple instance learning (MIL) concerns with classifying objects where each object is represented with a bag containing multiple instances. Compared to standard supervised learning problems, where each instance has a label, only the bags are labeled. MIL respects the complete internal structure of an object with a collection of multiple instances. For example, images are generally represented by a collection of patches in computer vision. This way, certain problems regarding the location or scale invariance can be avoided. Moreover, MIL framework is suitable to a diverse domain of applications such as molecule activity prediction [2], image categorization [3], web mining [4], and audio recording classification [5]. In MIL, the label information is provided for bags and instance labels are unknown. Even when instance labels are known, there should be a rule/model providing the bag label information. Suppose in an image classification problem, the aim is to classify a person riding a horse. Certain images can have patches labeled as person, some others have patches from horse class. An image containing both defines the positive class in this scenario. In any case of (labeled/unlabeled) instances, bag-level summary of the instance distribution is required. To resolve this problem, most of the existing studies make assumptions regarding the instance labels. For example, *standard* MIL assumption prevails in most of the existing MIL approaches. In standard MIL problem, there is at least one positive instance in positive bags and all other instances in given data are negative.

Considering the limited structure of standard MIL, a variety of assumptions on relating instance labels with bag labels are introduced in [6] as *generalized MIL*. In generalized MIL, a certain portion of potentially

*Correspondence: eskucukasci@ticaret.edu.tr

positive instances must be contained in positive bags. Moreover, these positive instances may belong to different data regions of the instance-feature space and are effective on the bag labels. As a generalized assumption, [7] proposed so called *collective* assumption in which each instance equally and independently contributes to the bag label. A wide range of MIL methods prioritize generalized MIL to embrace different MIL applications by managing multiinstance data [8]. To tackle generalized MIL problems, we predict bag class labels by aggregation of instance contributions. Instance-level scores are obtained by an appropriate mapping function of feature weights. Then, a bag is represented by simply averaging the instance-level scores, which is analogous to the collective assumption. This kind of approach deals with a variety of MI assumptions by optimizing feature weights to assess contribution of each instance to the bag label.

Researchers make use of margin maximization based approaches to solve MIL problem [9, 10]. Generally, interbag margin is maximized but the ways of relating instance margin to bag margin differ. More importantly, most of the existing optimization-based methods suffer from scalability problems, which is a major challenge in MIL problems. Considering the limitations of previous approaches, we propose a novel MIL framework. As opposed to margin maximization based MIL models, we build MI classifiers using a simplified optimization framework. Our approach models the contributions of instances to the bag labels rather than individually labeling them. The instance level contributions are implicitly mapped into a latent variable to obtain the bag class membership estimates.

Figure 1 shows the way of mitigating instance information to obtain a bag-level mapping on an illustrative example from UCSB Breast Cancer dataset [11]. Two cellular images belonging to malignant (positive) class and benign (negative) class are considered as bags. Instances of the bags are sampled as square patches of the images on a grid as exemplified in Figure 1. In classification, the aim is to predict the label of a bag given its set of instances. Instance-level estimates between 0 and 1 are calculated by a linear decision function. For each bag, scores of corresponding instances are averaged to assess bag-level class probability estimate. Classification scores of the bags in Figure 1 are predicted as 0.76 for the positive bag, and 0.22 for the negative bag by simply averaging the pseudo-class memberships of corresponding instances.

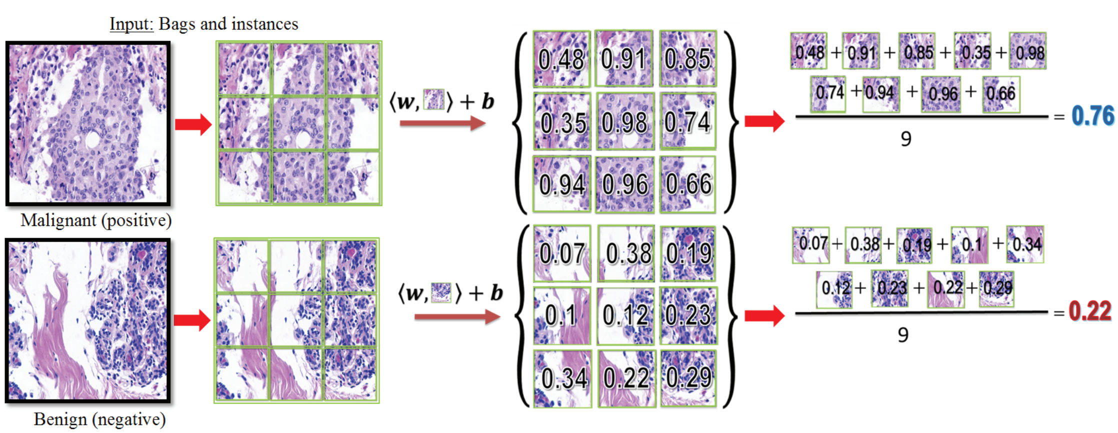


Figure 1. An example of bag class membership estimation.

In our proposal, we also process all training instances and their relationships to determine bag classes. It is shown in [12] that there is weak correlation between bag-level and instance-level performance of MIL classifiers. Hence, instance labels are not necessarily to be predicted correctly and true labels of instances are not known in most of the datasets. In the described example, only the final bag label estimate is sufficient for diagnosis of the disease as shown in Figure 1. This way, instances and corresponding bags are related without enforcing any requirements on the binding MIL assumption. Note that certain informative instances are prioritized by using a scoring idea to assess bag-level estimates. Similarly, insignificant instances are ineffective through proper determination of their scores.

Resulting classifiers are linear functions in the given feature space and have low capability of modeling nonlinear decision boundaries. An appropriate transformation of the original features is needed to apply classifiers to nonlinear data. As mentioned in [13], bags are not independently identically distributed samples of the underlying instance-feature space. Exploiting unsupervised dissimilarities leads to capture the unknown and potentially nonlinear relationships between instances from positive and negative bags. To capture nonlinear relationships among all training instances, we consider an instance dissimilarity based data representation. The new features are the dissimilarities to all training instances which embed bags to a higher dimensional space.

We compare our learning procedure with state-of-the-art MIL methods on a wide range of MIL benchmark datasets to highlight the classification success on different application domains. Section 2 is an overview of related works. Section 3 provides the formal description and proposed linear optimization based MIL framework. The datasets, computational results and discussions are presented in Section 4. Finally, conclusions and the overview of the future research directions are given in Section 5.

2. Related work

Most of the instance-level MIL approaches adopt standard MIL assumption. The first MIL paper [2] introduces formal descriptions of both MIL problem and standard MIL assumption; whereas, [14] presents a survey on standard MIL methods. In addition to the first MIL method axis parallel rectangles (APR) [2] and Citation-kNN [15], a generative method diverse density (DD) [16] and its variant EM-DD [17] also solve standard MIL problem. A famous MIL method, MILES [3] performs embedded instance selection iteratively and assumes instances in both positive and negative bags belong to the target concept.

Aforementioned methods incorporate machine learning algorithms and their performance depend on the adaptation process to given data, such as fine tuning of parameters and data preprocessing. Hence, it is hard to prove that these methods suit up to a wide range of datasets. Mathematical programming approaches are also considered to solve MIL problems. MIL formulations in the literature are extensions of generic SVM model [9, 10, 18–20] where instance level margin maximization is performed for bag classification initially assuming that all instances in positive bags are positive. To compensate the impact of this assumption, a witness selection procedure is employed [9, 10, 20]. For each bag from positive class, an instance is selected as a witness to represent that bag. However, only standard MIL assumption suits this specification. In sparse transductive MIL method [18] solves a non-convex formulation of MIL problem. In mi-SVM and MI-SVM formulations [9], new constraints are added to the SVM formulation satisfying existence of witnesses. 1-norm SVM-based formulation in [19] is a linear program with bilinear constraints. MIL problem is formulated as a mixed $0-1$ quadratic programming problem in [20]. In [10], SVM formulations of MIL problem are derived as a hard margin and two soft margin maximization models.

Exact solution methods like concave convex procedure in [18] are time consuming. Heuristic methods proposed in [10, 20] are considerably fast in problems with moderate sized datasets but do not guarantee the quality of final solution [19]. As opposed to quadratic or mixed-integer quadratic programs, we solve models with a linear objective function and constraints. Furthermore, instead of repeatedly solving subproblems, we solve a single linear program, which is solvable in polynomial time.

Dissimilarity based MIL methods [3, 13, 21–23] exploit dissimilarities to the prototypes, a representative set of instances, to extract useful information with various data representations. MILES [3] and MILD [21] assume that instances from different concepts are independently identically distributed; whereas, MILDS [22] and Clustering MIL [23] select only some instances as prototypes thereby waiving instance relationship information. Previously, the multiple concept structure is captured at bag level by using bag kernels [24], Hausdorff distances between bags in Citation-kNN [15] and bag dissimilarities in MInD [25]. We classify bags by simply using the instance dissimilarities. Instance-level relationships are considered to benefit from the informative instances in bags since positive and negative bag classes may possess instances that are very similar to each other.

3. Linear programming for multiple instance learning

3.1. Problem description

In multiple instance learning (MIL), a bag, B_j is formed by n_j many d-dimensional instances $B_j = \{\mathbf{x}_i : \mathbf{x}_i \in \Re^d,\ i = 1, 2, \ldots, n_j\}$. A bag B_j is also associated with a binary class label $y_j \in \{-1, 1\}$. $\chi = \{B_j : j = 1, .., m\}$ is the set of given bags with their corresponding instance vectors. It is practical to transform the original input χ using function $\phi(\mathbf{x}_i)$, which admits to another representation of input data, say χ'. For instance, the similarities to prototype instances [3], or a graph kernel [13] transforms the original data to discover its underlying structure. Given χ or χ' with bag labels $y_j,\ j = 1, \ldots, m$, our MIL task is to predict labels of unseen bags based upon a linear decision function. For each bag, instance-level scores are computed to determine the bag class label.

3.2. The proposed linear programming model of MIL

To formulate MIL problem as a linear programming (LP) model, we define the sets, parameters and decision variables used in the model as follows.

Indices:

$i = 1, 2, \ldots, n$: indices for the instances

$j = 1, 2, \ldots, m$: indices for the bags

Sets:

$J^+ = \{j : y_j = 1\}$: set of positive bags

$J^- = \{j : y_j = -1\}$: set of negative bags

$J = J^+ \cup J^-$: set of all bags

$I^+ = \{i : i \in I_j \wedge j \in J^+\}$: set of instances in positive bags

$I^- = \{i : i \in I_j \wedge j \in J^-\}$: set of instances in negative bags

$I = I^+ \cup I^-$: set of all instances

Parameters:

$\mathbf{x}_i \in \Re^d,\ i = 1, 2, \ldots, n$: instance vectors

$y_j,\ j = 1, 2, \ldots, m$: bag labels

Decision variables:

$\mathbf{w}$: d-dimensional feature weight vector

b: bias of the linear function

$m_i,\ i = 1, 2, \ldots, n$: instance pseudo class memberships

$\beta_j,\ j = 1, 2, \ldots, m$: bag class memberships

σ_{jl}, $j \in J^+$, $l \in J^-$: bag class membership differences

Our learning approach ranks the bags in a binary classification problem. Namely, a positive bag is ranked before an arbitrary negative bag after classification. Area under the ROC curve (AUC) is the most commonly used measure to evaluate the success of ranking problems. Using a least-squares SVM algorithm, [26] solves AUC maximization problem by comparing positive and negative instance pairs. AUC can be calculated using Wilcoxon-Mann-Whitney (WMW) statistic [27], which can be written for positive and negative bags as

$$W = \frac{\sum_{j\in J^+}\sum_{l\in J^-} I(\beta_j, \beta_l)}{|J^+||J^-|},$$

where $I(\beta_j, \beta_l) = \begin{cases} 1 & \text{if} \quad \beta_j > \beta_l, \\ 0 & \text{otherwise.} \end{cases}$

WMW statistic yields the quantity of positive bags having higher rank compared to the negative bags, which is divided by the number of all possible bag pairs. Our LP model minimizes pairwise positive and negative bag class differences, which is equivalent to optimization of the bag ranks [28]. Therefore, comparison of positive and negative bag pairs can also be casted as solving AUC maximization problem.

Instead of labeling each instance individually, determination of class membership scores permits contributions of instances from multiple concepts with different importance degrees to the bag class. Hence, membership values are not assessed by favoring a specific target concept as observed in the standard MIL problem. This property emphasizes the superiority of our approach compared to the margin maximization based methods where standard MIL assumption is deemed [9, 15]. Finally, a linear binary MIL classifier is built by solving the following model:

$$\text{(LP)} \quad \max_{\mathbf{w},b,\boldsymbol{\beta},\mathbf{m},\boldsymbol{\sigma}} \quad \sum_{j\in J^+}\sum_{l\in J^-} \sigma_{jl} \tag{1a}$$

$$\text{st} \quad \langle \mathbf{w}, \mathbf{x}_i \rangle + b = m_i \quad \forall i \in I \tag{1b}$$

$$\beta_j = \frac{1}{n_j}\sum_{i\in I_j} m_i \quad \forall j \in J \tag{1c}$$

$$\beta_j = \beta_l + \sigma_{jl} \quad \forall j \in J^+, \forall l \in J^- \tag{1d}$$

$$0 \leq m_i \leq 1 \quad \forall i \in I \tag{1e}$$

The values of variables $m_i, \forall i = 1, 2, \ldots, n$ correspond to instance pseudo class memberships which are bounded by Constraint (1e). As introduced, $\mathbf{w}$ is the feature weight vector, whereas b is the bias parameter that are optimized to form an instance level separating hyperplane. This hyperplane decides the instance pseudo class memberships in Constraint (1b). Constraint (1c) forms the bag class memberships $\beta_j, \forall j = 1, \ldots, m$ based on the summation of instance pseudo class memberships for each bag, which is normalized with the size of the corresponding bag, n_j. Constraint (1d) characterizes the bag differences for each positive and negative bag pair which are imposed by the slack variables σ_{jl}, $\forall j \in J^+$ and $\forall l \in J^-$. Finally, the objective function (1a) maximizes the summation of these slack variables to maximize bag class separation. The resulting model is efficient to solve since it has a linear objective function and constraints. All the instances in training bags constitute to the classifier during optimization. LP solution provides a classifier $\langle \mathbf{w}, \mathbf{x}_i \rangle + b$ which determines instance pseudo-class membership value for an arbitrary d-dimensional instance vector $\mathbf{x}_i$, i.e. $m_i = \langle \mathbf{w}, \mathbf{x}_i \rangle + b$.

For each instance in the dataset, a membership value between 0 and 1 must be decided to map the bag level estimates onto the 0 to 1 interval. We regard this membership value as pseudo class label estimate. If the membership value is less than a threshold, the instance can be assigned to the negative class. Otherwise, the instance is considered to belong to the positive class. The threshold can be selected based on the highest accuracy level on training bags. We assess the pseudo-membership values of instances to find bag-level estimates, not for instance labeling since the actual instance labels are not known in MIL tasks. Class membership estimate for a bag B_j is determined by averaging pseudo class membership values of its possessed instances as $\beta_j = \frac{1}{n_j} \sum_{i \in I_j} m_i, \quad \forall j \in J$. This representation eliminates single witness instance selection encountered in previous proposals and leads to an optimization problem with continuous variables and linear constraints. To classify a test bag, instance level scores are calculated and then averaged to find bag class label estimates. Such an approach is simple and efficient to implement and there are no hyperparameters that need to be tuned.

3.3. Data representation

In MIL, it is not enough to describe objects with multiple instance vectors, the relationships between these vectors must also be represented. The researchers conducted MIL experiments on various data representations by calculating the dissimilarities to selected prototypes [3, 21, 22, 25, 29]. In our LP-based MIL framework, we preprocess the input data to allow learning different characteristics of MIL datasets. Solving LP model produces a decision boundary by means of a linear classifier. Most of MIL datasets are formed of complex objects with potentially nonlinear instance relationships. The input data can be transformed to carry out nonlinear classification in a new, possibly higher dimensional space. A linear classifier is simple to apply and capable of nonlinear separation in the new feature space [30].

Given a set of bags $\mathcal{X} = \{B_1, \ldots, B_m\}$, each bag B_j is composed of n_j many instances. The original instance-feature space is described with d many features. Initially, both training set and test set are preprocessed by standardization using the feature means and standard deviations. Preliminarily, we processed pairwise training instance dissimilarities to learn a MIL classifier. The dissimilarities between instances $\mathbf{x}_i$ and x_k are calculated by using the squared Euclidean distance $\delta_{ik} = (\mathbf{x}_i - \mathbf{x}_k)^T (\mathbf{x}_i - \mathbf{x}_k)$. In a test bag, distances to all training instances are calculated for each instance of that bag. The dimensionality of the new space equals to total number of instances in training bags, i.e., n and the new representation is referred to as $\mathrm{R}^{\text{instance}}$. When n is large, there are large number of variables in LP model which introduce computational difficulties. Moreover, since the $n \times n$ dimensional large and dense instance dissimilarity matrix forms a mathematical model with dense columns. Consequently, the solution time is affected especially for large datasets. Curse of dimensionality and overfitting due to noisy features in the enlarged representation are categorized as the further problems.

To solve LP model on large-scale MIL problems, we offer a simplified version of the first data representation using clustering. Clustering instances is conducted in MIL setting either to detect the target concept [23] or to obtain a new bag-level data representation [31]. In our clustering-based data representation, cluster centers are selected as prototypes. After clustering the instances using k-means algorithm, instance-to-prototype distances build up the input data. Since dimensionality of the input dissimilarity matrix is decreased by clustering (i.e., there exists κ many clusters), clustering-based data representation is advantageous in datasets with large number of instances. We define the dissimilarity between instance $\mathbf{x}_i$ and cluster center $\mathbf{c}_j$ as

$r_{ij}^{c} = (\mathbf{x}_i - \mathbf{c}_j)^T (\mathbf{x}_i - \mathbf{c}_j)$ where $\mathbf{c}_1, \ldots, \mathbf{c}_\kappa$ are the cluster centers. As a result, each instance is described by a κ-dimensional feature vector. In the final representation, which is denoted by $\mathrm{R}^{\mathrm{cluster}}$, the total number of distance calculations are reduced compared to $\mathrm{R}^{\mathrm{instance}}$ since the selected prototypes are cluster centers instead of all training instances. Since instance label information and binding MI assumption are the two main ambiguities of MIL problems, determination of the informative instance dissimilarities is necessary to remove uncertainty in bag classification. The two alternative representations can be tested on a subset of the given data to understand the underlying structure of the whole data. Simple calculations are performed by selected Euclidean distance metric and no parametrization is required to obtain $\mathrm{R}^{\mathrm{instance}}$ representation. In order to reduce computational time, $\mathrm{R}^{\mathrm{cluster}}$ representation can be exploited.

4. Experiments and results

4.1. Experimental setup and evaluation criteria

Initially, we transform the data to zero mean and unit variance. We perform 5 repeats of a stratified ten-fold cross validation to evaluate the classifier performance on each dataset. LP problems are modeled in Gurobi Python interface and solved using Gurobi 7.5 [32]. Input data representations are acquired using scikit-learn [33] library. All the experiments are carried out on a Windows 10 system with dual core CPU (i5-3470, 3.2 GHz) and 12 GB of RAM. In order to perform a fair comparison over state-of-the-art MIL methods, we use the same train/test split indices for each method and experiment. All the scripts, datasets and cross-validation indices are made available on our supporting page [34]. $\mathrm{R}^{\mathrm{instance}}$ representation has no parameters to be predetermined whereas $\mathrm{R}^{\mathrm{cluster}}$ has the input parameter number of clusters κ. We simply identify value of κ using the elbow method based on total within cluster variance and increase the gain in computational time. After learning the representations, LP formulation in Model (1) is solved to obtain the bag classifier. The convergence tolerance for the barrier algorithm is set to 0.01 and default values of the solver are used for the other parameters. Finally, state-of-the art approaches are experimented via their provided MATLAB [35] implementations. We followed the settings proposed by the authors. MInD [25] employs default parameters. The parameters of miFV [36] are selected by an inner ten-fold cross-validation.

Performance of a MIL classifier can be evaluated the area under of the receiver operating characteristic curve (ROC) [37]. ROC curve plots the true positive rate versus the false positive rate of a classifier depending on all decision thresholds. The area under ROC curve (AUC) is a commonly used metric to compare different classification algorithms. AUC is a more discriminative measure than accuracy [38] since a predetermined decision threshold is necessary to report accuracy. Besides, AUC maximization is related to maximization of positive and negative bag membership differences in LP model. AUC also improves classification accuracy by ranking positive bags ahead the negative bags, and is therefore an appropriate evaluation metric for our experiments.

4.2. Results

We perform experiments on real world MIL datasets to verify the effectiveness of our approach. MIL datasets are described in Table 1 in our webpage [34] and are categorized based on the application domain. To the best of our knowledge, this is the largest MIL dataset repository with reported results on a proposed MIL framework. Each dataset has different characteristics such as number of bags, number of instances in bags and number of features. For some datasets such as Corel [3] and Birds [5], class imbalance occurs at bag-level. Another property of the datasets is discussed in [39] is the low proportion of positive instances in positive bags, as observed in Newsgroups [13]. As a consequence, we tackled MIL problems from different application domains and investigate the utility of our MIL framework across various data characteristics.

To demonstrate the effectiveness and superiority of LP-MIL on real-world datasets, we also experimented the following baseline methods: miFV [36] and dissimilarity-based representations (MInD) [25] with $\mathrm{D}_{\mathrm{meanmin}}$ representation. We solve LP problem (Model (1)) on $\mathrm{R}^{\mathrm{instance}}$ and $\mathrm{R}^{\mathrm{cluster}}$ representations of the datasets described in Table 1 [34]. At first, the significance of the differences are discussed according to the procedure recommended by [40]. A Friedman test [41] is applied to the ranks of the algorithms over all datasets. Since the null hypothesis that all methods have equal AUC performance at the 0.05 level, we proceed with the Nemenyi test [42] to check whether the pairs of classifiers are significantly different from each other. Pairwise differences of the methods are significant if their average ranks differ by at least the critical difference (CD). The resulting CD value for four classifiers at significance level 0.05 is 0.561. By using the rankings of the algorithms on each dataset and the average ranks, a CD diagram [40] shown in Figure 2 is obtained. Performances of LP with $\mathrm{R}^{\mathrm{instance}}$, MInD with $\mathrm{D}_{\mathrm{meanmin}}$ and miFV are not significantly different from each other according to the differences demonstrated in Figure 2. miFV and LP with $\mathrm{R}^{\mathrm{cluster}}$ are not significantly different from each other since their average rank difference is below the CD. Performance of LP model critically differs when either $\mathrm{R}^{\mathrm{instance}}$ or $\mathrm{R}^{\mathrm{cluster}}$ representations form the input data.

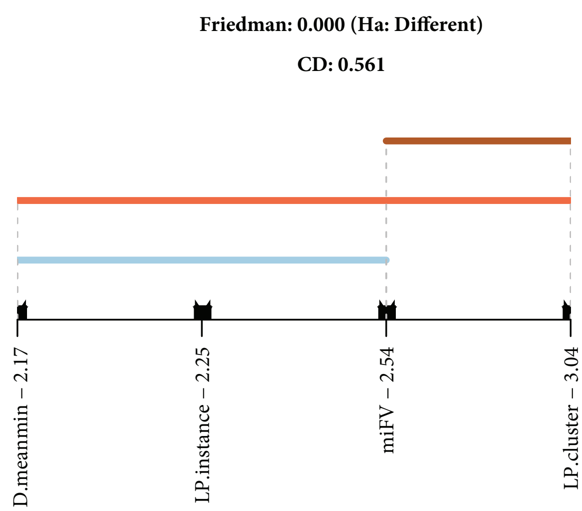
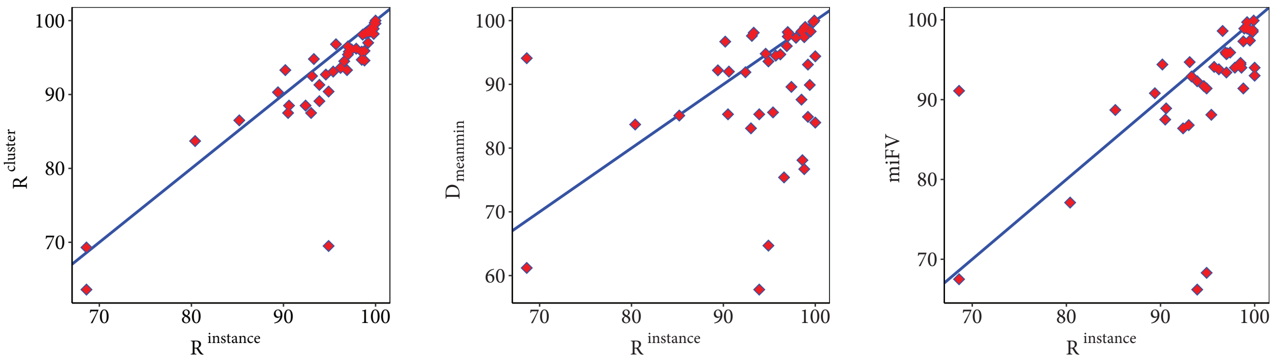


Figure 2. The average ranks for MIL methods on 71 datasets based on mean AUC performance. The critical difference at 0.05 is 0.561.

Scatter plots in Figure 3 shows the pairwise comparisons of the approaches. Two methods equally perform on a dataset if the corresponding point falls on the line $x = y$. The points falling below the line $x = y$ represent the datasets that are more accurately classified by the method on the x axis. Otherwise if a point is above the line $x = y$, the approach on the y axis is more successful on the corresponding dataset. Figure 3a shows the scatter plot comparison of LP results on $\mathrm{R}^{\mathrm{instance}}$ and $\mathrm{R}^{\mathrm{cluster}}$ representations and performance of $\mathrm{R}^{\mathrm{instance}}$ is more successful in 48 datasets. As seen in Figures 3b and 3c, AUC results of LP with $\mathrm{R}^{\mathrm{instance}}$ are competitive with the other two methods. However, on a group of datasets performances of both $D_{meanmin}$ and miFV are superior, which are the text classification datasets. In real-world MIL applications except for text classification, LP with $R^{instance}$ is the leading method as the ranking results in Figure 4 indicates that and its difference with all other methods is larger than the CD 0.733.



(a) AUC comparison of $R^{instance}$ and $R^{cluster}$ representations

(b) AUC comparison of LP with $R^{instance}$ and MInD with $D_{meanmin}$

(c) AUC comparison of LP with $R^{instance}$ and miFV

Figure 3. Pairwise AUC comparison of various MIL methods on 71 real-world datasets.

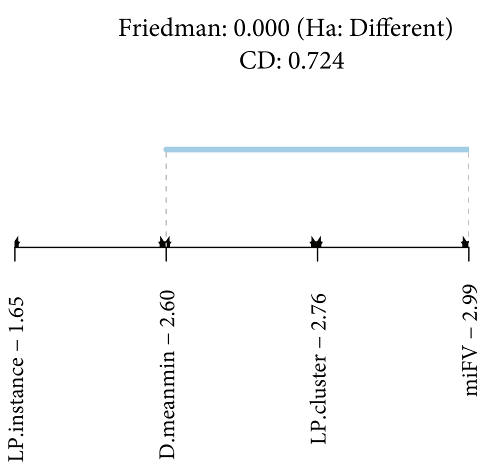


Figure 4. The average ranks for MIL methods on 42 datasets based on mean AUC performance. The critical difference at 0.05 is 0.733.

AUC results of all methods on 71 datasets are provided in Table . LP with $R^{instance}$ model has superior performance on Musk 1 and Mutagenesis 2. The best AUC result on Protein dataset is obtained by LP with $R^{cluster}$. Result of LP with $R^{instance}$ on Protein dataset is not provided due to the memory restrictions. In Musk 2, MInD with $D_{meanmin}$ has the best AUC. Best average results for Mutagenesis 1 are obtained by miFV and followed by LP with $R^{cluster}$. In most of the Corel image datasets, LP with $R^{instance}$ is the leading method together with the image datasets UCSB Breast Cancer, Elephant, Fox and Tiger. MInD with $D_{meanmin}$ also successful on Corel image datasets. MInD with $D_{meanmin}$ has the best performance on Newsgroups datasets whereas miFV performs better than other methods in Web datasets. Finally, LP with $R^{instance}$ representation is quite successful compared to the other methods in Birds datasets.

Table . AUC and standard error (× 100) results of various MIL methods. 10 fold cross-validation is repeated 5 times.

Dataset	Algorithm AUC (%)			
	LP			
	$R^{instance}$	$R^{cluster}$	MInD ($D_{meanmin}$)	miFV
Musk 1 ♣	95.7 (0.9)	**96.8 (0.8)**	94.5 (1.2)	94.1 (1.2)
Musk 2 ♣	93.1 (1.0)	92.7 (1.1)	**97.6 (0.8)**	94.7 (1.2)
Mutagenesis 1 ♣	85.2 (1.5)	86.7 (1.3)	85.1 (1.2)	**88.7 (1.2)**
Mutagenesis 2 ♣	**78.8 (3.9)**	78.5 (4.0)	64.7 (5.3)	68.3 (5.0)
Protein ♣	-	**83.9 (1.4)**	52.3 (3.7)	80.0 (1.9)
Elephant ♥	**94.9 (0.5)**	90.5 (1.0)	93.6 (0.9)	91.4 (0.9)
Fox ♥	**68.6 (1.4)**	64.2 (1.5)	61.2 (1.7)	67.5 (1.5)
Tiger ♥	**90.5 (0.9)**	89.3 (1.0)	85.3 (1.1)	87.5 (1.1)
Corel, African ♥	94.5 (0.6)	93.2 (0.7)	**96.7 (0.4)**	94.4 (0.6)
Corel, Antique ♥	89.4 (0.8)	90.0 (0.5)	**92.2 (0.6)**	90.8 (0.6)
Corel, Battleships ♥	93.3 (0.6)	95.2 (0.4)	**98.1 (0.2)**	92.9 (0.6)
Corel, Beach ♥	**99.5 (0.1)**	98.8 (0.2)	98.3 (0.4)	97.4 (0.4)
Corel, Buses ♥	**97.9 (0.2)**	96.3 (0.3)	97.3 (0.4)	94.0 (0.7)
Corel, Cars ♥	94.6 (0.6)	92.6 (0.7)	**94.8 (0.5)**	91.7 (0.7)
Corel, Desserts ♥	**98.8 (0.1)**	95.9 (0.4)	97.4 (0.3)	97.3 (0.4)
Corel, Dinosaurs ♥	**98.5 (0.2)**	95.3 (0.3)	98.3 (0.2)	94.4 (0.5)
Corel, Dogs ♥	**92.4 (0.6)**	88.6 (0.8)	91.9 (0.7)	86.4 (1.2)
Corel, Elephants ♥	97.0 (0.2)	96.4 (0.2)	**98.2 (0.2)**	95.7 (0.4)
Corel, Fashion ♥	98.9 (0.4)	98.1 (0.1)	**99.0 (0.1)**	98.9 (0.2)
Corel, Flowers ♥	**96.2 (0.4)**	93.8 (0.5)	94.7 (0.6)	93.8 (0.6)
Corel, Food ♥	**99.8 (0.0)**	98.3 (0.1)	**99.8 (0.1)**	98.7 (0.1)
Corel, Historical ♥	**99.8 (0.0)**	98.8 (0.1)	**99.8 (0.0)**	98.5 (0.3)
Corel, Horses ♥	90.6 (0.6)	89.3 (0.7)	**92.0 (0.6)**	88.9 (0.8)
Corel, Lizards ♥	97.1 (0.3)	95.7 (0.5)	**98.0 (0.3)**	95.8 (0.5)
Corel, Mountains ♥	99.9 (0.1)	99.7 (0.1)	**100 (0.0)**	99.9 (0.0)
Corel, Skiing ♥	**96.9 (0.3)**	93.1 (0.5)	96.0 (0.3)	95.9 (0.4)
Corel, Sunset ♥	80.4 (1.2)	83.1 (0.9)	**83.7 (1.0)**	77.1 (1.3)
Corel, Waterfalls ♥	97.0 (0.3)	95.4 (0.3)	**97.5 (0.2)**	93.4 (0.5)
UCSB Breast Cancer ♥	**93.0 (2.0)**	90.3 (2.2)	83.1 (2.7)	86.8 (2.5)
Newsgroups 1, alt.atheism ♠	47.0 (2.5)	66.8 (2.8)	**94.1 (1.0)**	91.1 (1.2)
N.g. 2, comp.graphics ♠	61.0 (2.3)	50.4 (3.0)	**89.8 (1.6)**	57.2 (3.2)
N.g. 3, comp.os.ms-windows.misc ♠	44.6 (2.8)	63.4 (2.5)	**81.0 (2.1)**	66.8 (2.2)
N.g. 4, comp.sys.ibm.pc.hardware ♠	53.0 (2.7)	56.5 (3.2)	**85.7 (2.2)**	69.5 (2.4)
N.g. 5, comp.sys.mac.hardware ♠	50.6 (2.2)	64.6 (3.2)	**85.2 (1.6)**	65.0 (2.6)
N.g. 6, comp.windows.x ♠	59.5 (2.6)	57.8 (2.8)	**89.0 (1.7)**	82.2 (2.0)
N.g. 7, misc.forsale ♠	53.5 (2.3)	56.9 (3.1)	**79.0 (2.0)**	72.6 (2.5)
N.g. 8, rec.autos ♠	48.5 (2.5)	43.0 (3.3)	**87.0 (1.7)**	72.7 (2.5)
N.g. 9, rec.motorcycles ♠	63.0 (2.8)	43.8 (2.7)	32.6 (3.2)	**81.2 (2.4)**
N.g. 10, rec.sport.baseball ♠	64.3 (2.4)	49.8 (3.0)	**91.4 (1.4)**	86.4 (1.8)
N.g. 11, rec.sport.hockey ♠	49.0 (2.5)	45.8 (3.2)	**95.8 (0.8)**	87.9 (1.5)
N.g. 12, sci.crypt ♠	52.2 (2.6)	55.5 (2.8)	84.0 (1.9)	**85.1 (1.8)**
N.g. 13, sci.electronics ♠	45.8 (2.1)	48.8 (4.0)	**94.6 (1.0)**	61.6 (2.6)
N.g. 14, sci.med ♠	61.2 (2.5)	46.8 (3.2)	**94.2 (0.8)**	84.3 (1.7)
N.g. 15, sci.space ♠	43.0 (2.3)	51.6 (3.1)	90.5 (1.4)	**82.9 (1.9)**
N.g. 16, soc.religion.christian ♠	41.6 (2.7)	43.7 (3.0)	**89.8 (1.4)**	84.9 (1.5)

Table . (Continued).

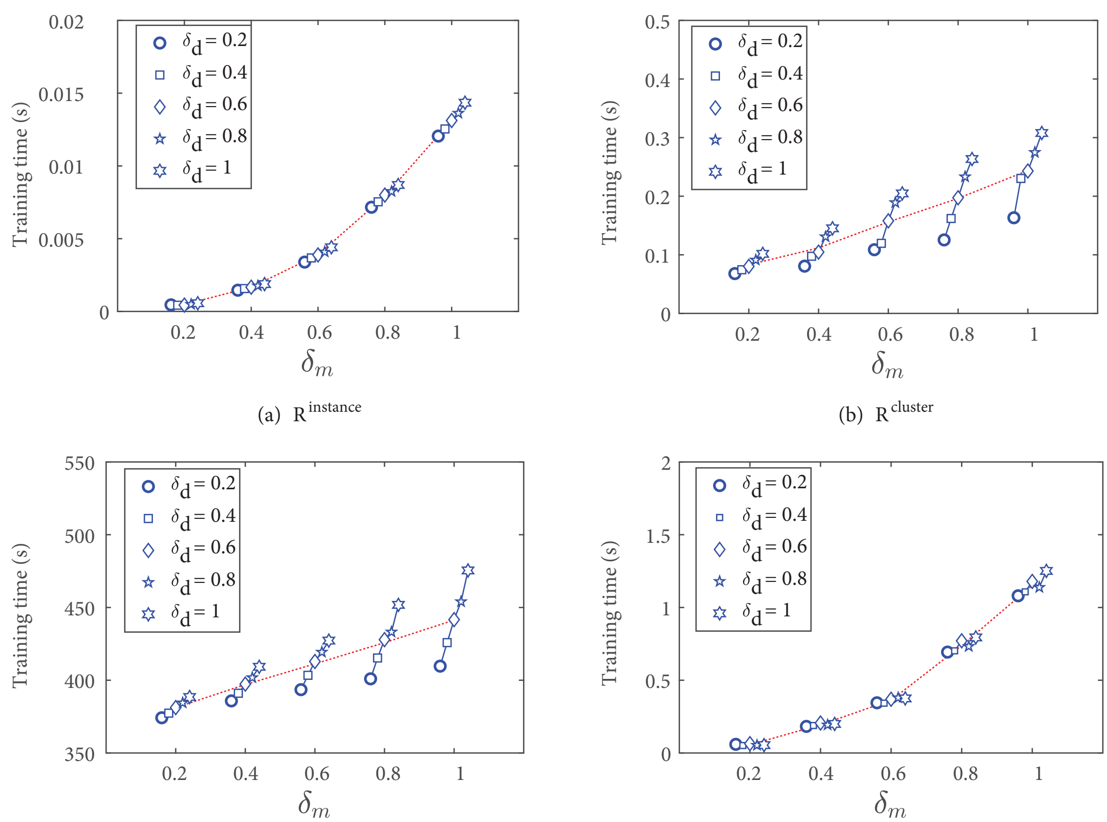
Dataset	Algorithm AUC (%)			
	LP			
	$R^{instance}$	$R^{cluster}$	MInD ($D_{meanmin}$)	miFV
N.g. 17, talk.politics.guns ♠	41.6 (2.7)	50.8 (2.8)	**87.4 (1.5)**	82.7 (2.0)
N.g. 18, talk.politics.mideast ♠	56.7 (2.5)	49.0 (3.1)	**87.4 (1.7)**	85.8 (1.9)
N.g. 19, talk.politics.misc ♠	51.5 (1.9)	50.8 (2.3)	**80.2 (1.9)**	67.2 (2.9)
N.g. 20, talk.religion.misc ♠	38.6 (2.3)	61.9 (2.7)	**83.4 (2.2)**	80.9 (2.3)
Web 1 ♠	75.9 (3.0)	64.2 (3.2)	63.4 (4.2)	**83.2 (2.3)**
Web 2 ♠	46.3 (4.1)	**64.7 (3.6)**	47.4 (4.2)	37.1 (2.5)
Web 3 ♠	64.5 (4.2)	62.2 (3.9)	70.8 (4.6)	**73.3 (3.6)**
Web 4 ♠	74.1 (3.7)	60.4 (3.8)	79.9 (3.6)	**81.2 (3.4)**
Web 5 ♠	**73.2 (3.5)**	53.4 (4.0)	71.1 (3.7)	68.7 (3.4)
Web 6 ♠	56.4 (4.4)	41.7 (4.4)	52.5 (4.2)	**64.6 (3.6)**
Web 7 ♠	64.3 (2.9)	46.1 (3.2)	69.0 (2.8)	**69.7 (3.4)**
Web 8 ♠	50.7 (3.0)	46.9 (2.4)	40.9 (2.6)	**53.7 (2.4)**
Web 9 ♠	44.0 (3.2)	45.5 (3.0)	73.5 (2.7)	**68.5 (3.1)**
Birds, Brown creeper ♦	**99.4 (0.1)**	98.4 (0.2)	89.9 (0.5)	98.8 (0.2)
Birds, Chestnut-backed chickadee ♦	**93.9 (0.4)**	88.8 (0.7)	85.3 (0.8)	92.3 (0.8)
Birds, Dark-eyed junco ♦	**95.4 (0.6)**	93.4 (0.7)	85.6 (1.3)	88.1 (1.2)
Birds, Hammonds flycatcher ♦	**100.0 (0.0)**	**100 (0.0)**	94.4 (0.7)	94.0 (0.7)
Birds, Hermit thrush ♦	**93.9 (1.4)**	90.9 (1.0)	57.8 (4.4)	66.2 (3.1)
Birds, Hermit warbler ♦	**98.6 (0.2)**	98.2 (0.2)	78.1 (1.5)	94.0 (0.6)
Birds, Olive-sided flycatcher ♦	**97.4 (0.2)**	96.2 (0.3)	89.6 (0.6)	95.9 (0.4)
Birds, Pacificslope flycatcher ♦	96.6 (0.3)	94.5 (0.4)	75.4 (1.0)	**98.6 (0.2)**
Birds, Red-breasted nuthatch ♦	**98.5 (0.2)**	94.7 (0.4)	87.6 (0.7)	94.6 (0.5)
Birds, Swainsons thrush ♦	**98.8 (0.2)**	94.5 (0.4)	76.7 (1.7)	91.4 (1.0)
Birds, Varied thrush ♦	**100.0 (0.0)**	99.6 (0.1)	84.0 (1.2)	93.0 (0.7)
Birds, Western tanager ♦	**99.2 (0.1)**	97.0 (0.3)	84.9 (1.8)	98.9 (0.2)
Birds, Winter wren ♦	99.2 (0.1)	98.5 (0.2)	93.1 (0.7)	**99.7 (0.1)**

MIL application categories: ♣ molecular activity prediction, ♥ image annotation, ♠ text classification, ♦ audio recording classification.

4.3. Computational time analysis

Time complexity of obtaining $R^{instance}$ representation using Euclidean distances to instances in training bags is $\mathcal{O}(n^2d)$. We use k-means clustering algorithm to form the $R^{cluster}$ representation. Time complexity of k-means algorithm is $\mathcal{O}(In\kappa d)$ where κ is the number of clusters and I is the necessary number of iterations until convergence. After determining the κ many cluster centers, it takes $\mathcal{O}(n\kappa d)$ times to have the final $R^{cluster}$ representation. LP problems belong to the complexity class P [43]. LP solutions are generated in polynomial time since we use barrier solver of Gurobi version 7.5. The testing times after LP solutions are $\mathcal{O}(n)$ for $R^{instance}$ and $\mathcal{O}(\kappa)$ for $R^{cluster}$. We report training and testing times of data representation learning and the time taken to build a classifier which is the model solution time. We also report representation learning times of miFV [36] and MInD [25] with $D_{meanmin}$. Both miFV [36] and MInD [25] represents bags using a new bag-level feature vector. Then, bag representation vectors form the input of the linear SVM classifier. miFV [36] employs LibLinear package [44] which takes $\mathcal{O}(n)$ times, whereas MInD [25] uses LiBSVM [45] with a learning time that scales between $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$. Prediction time of a test bag takes $\mathcal{O}(h)$ times where h is the dimensionality of the obtained bag representation. The testing times of LP solutions and SVM classifiers of miFV [36] and MInD [25] are negligible since only a few vector multiplications and arithmetic operations are performed.

In order to observe the time complexity, pseudo-synthetic datasets have various properties such as number of bags and features are generated. All the methods are experimented on pseudo-synthetic datasets that originate from Elephant dataset. Proportion of bags δ_m and proportion of features δ_d are selected from the set $\{0.2, 0.4, 0.6, 0.8, 1\}$. We repeat 10 replications of each setting combination and plot the average results. Figure 5 shows representation learning times of LP-MIL, miFV [36] and D_{meanmin} [25] on the training set. D_{meanmin} [25] and R^{cluster} increases linearly in terms of the increase in number of features and number of bags. In R^{instance} representation and miFV [36], a cubic growth is followed as the number of bags increases.



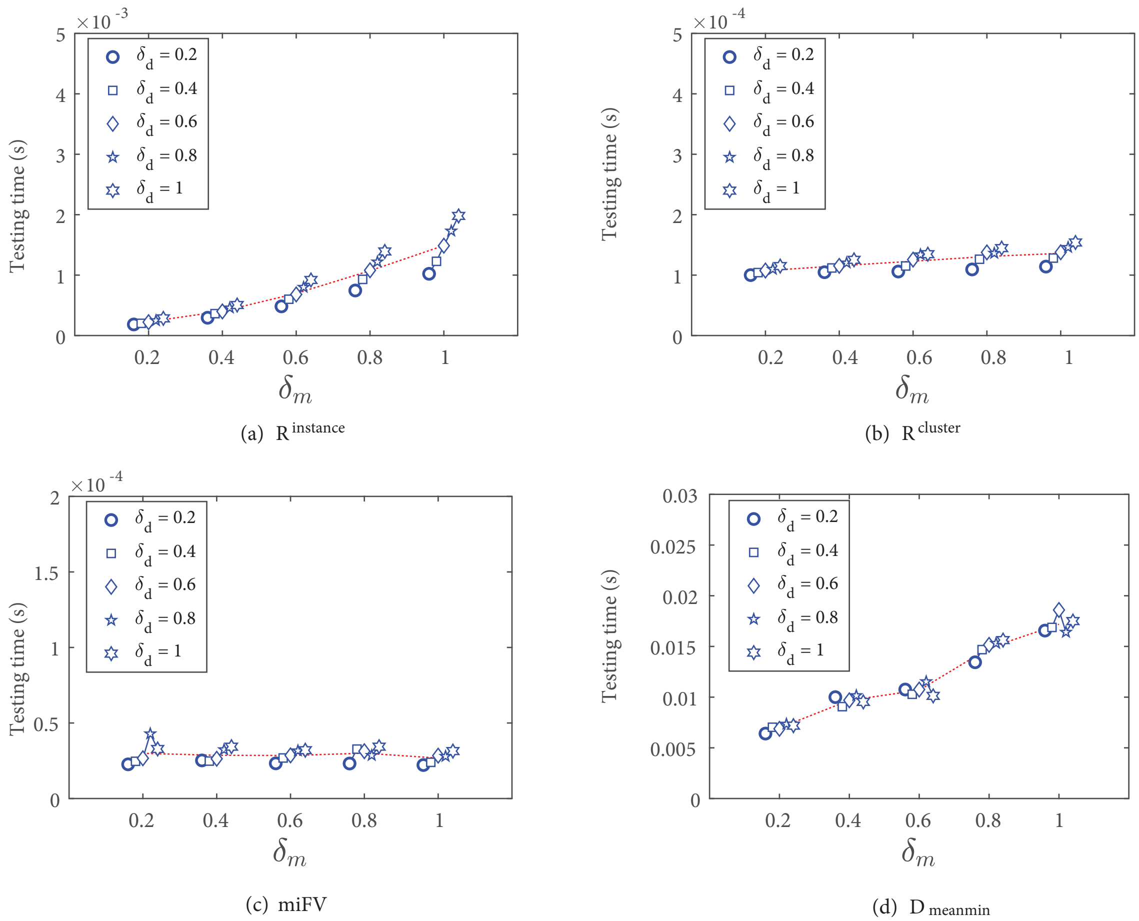
(a) R^{instance} (b) R^{cluster}

(c) miFV (d) D_{meanmin}

Figure 5. Training times of LP-MIL, miFV and D_{meanmin} on Elephant dataset with changing values of δ_m and δ_d.

It can be seen from Figure 6 that testing times of miFV [36] and R^{cluster} representation are robust to the changes in the data size properties. Effect of distance calculations degrade representation learning times both on training and test sets when number of bags and number of features are increased in R^{instance} representation and D_{meanmin} [25].

The performance of LP-based MIL especially depends on the model solution time. Once the LP model is built, the elapsed time during optimization is the classifier building time. Figure 7 shows the changes in model solution times for R^{instance} and R^{cluster} representations. Since dimensionality of R^{instance} is proportional



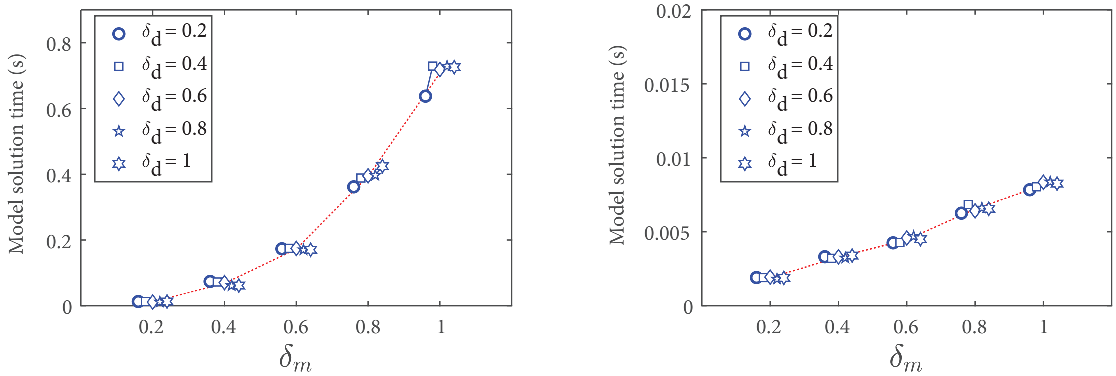
(a) R^{instance} (b) R^{cluster} (c) miFV (d) D_{meanmin}

Figure 6. Testing times of LP-MIL, miFV and D_{meanmin} on Elephant dataset with changing values of δ_m and δ_d.

to number of the training instances, LP solution times can be challenging in datasets with large number of bags or instances as demonstrated in Figure 7a. R^{cluster} representation is simple and generally low-dimensional compared to R^{instance}. Moreover, linear increase in the solution time curve in Figure 7b when solving LP formulation on R^{cluster} representation with increasing number of bags promotes this representation on large datasets.

5. Conclusion

In this paper, we propose a multiple instance learning framework including a new mathematical model of multiple instance classification and enhanced data representations. We efficiently solve the MIL problem without imposing strict assumptions on object descriptions. Our approach embeds instance relationships via inputting various data representations and determines class memberships of the objects. To the best of our knowledge, this is the first linear programming based classification approach in MIL. We compare our learning procedure with state-of-the-art MIL methods on a wide range of machine learning datasets to highlight the classification success on different application domains. Unlike the previous mathematical models of MIL, we do not force regular margin maximization. This leads to avoiding quadratic optimization, which is computationally more difficult than linear programming. Moreover, a common initialization setting of previous models is that all the instances in positive bags are positive and all the instances in negative bags are negative. This strong



(a) $R^{instance}$ + LP (b) $R^{cluster}$ + LP

Figure 7. Solution time of LP on representations $R^{instance}$ and $R^{cluster}$ of Elephant dataset with changing values of δ_m and δ_d.

assumption is not required in our approach since we only calculate pseudo-class memberships of instances regardless of the class label of their owner bag. We also exploit different data representations to improve success of the linear classifier. Instance dissimilarity spaces are constructed to represent the input data to perform nonlinear separation. In datasets with large number of instances, it is computationally demanding to form the new instance-feature space. In order to reduce amount of distance calculations between pairs of instances, we employed data clustering. Instead of instance dissimilarities, distances to the centers of generated clusters are the new features.

In this work, linear programs are solved to perform MI classification. Proposed mathematical models are efficient to solve on different input data representations. Processing the instance-level relationships and forming the bag label estimates using the instance-level scores deliver promising classification success on diversified real world MIL applications. As an extension, MIL can be used in large scale data mining applications requiring decentralized data storage. To decrease the solution times and considering the restrictions on data availability in such applications, subsets of the original data can be used to form a MI classifier. Inspections on the potential loss in classification accuracy due to not being able to process whole data may give rise to a reformulation of the proposed model. A commonly seen property in optimization-based data mining approaches is overfitting. Both data representation and classifier generation processes may reinforce this situation. Potential overfitting problems on some MIL datasets can be recovered by using an ensemble formed by repeatedly solving mathematical models on different subsamples of the data.

References

[1] Kaliberda M, Lytvynenko L, Pogarsky S. Method of singular integral equations in diffraction by semi-infinite grating: H-polarization case. Turkish Journal of Electrical Engineering & Computer Sciences 2017; 25 (6): 4496-4509. doi: 10.3906/elk-1703-170

[2] Dietterich TG, Lathrop RH, Lozano-Pérez T. Solving the multiple instance problem with axis-parallel rectangles Artificial Intelligence. 1997; 89 (1-2): 31-71. doi:10.1016/s0004-3702(96)00034-3

[3] Chen Y, Bi J, Wang JZ. MILES: Multiple-instance learning via embedded instance selection IEEE Transactions on Pattern Analysis and Machine Intelligence. 2006;28 (12):1931-1947. doi:10.1109/TPAMI.2006.248

[4] Zhou ZH, Jiang K, Li M. Multi-instance learning based web mining Applied Intelligence. 2005;22 (2): 135-147. doi:10.1007/s10489-005-5602-z

[5] Briggs F, Lakshminarayanan B, Neal L, Fern XZ, Raich R et al. Acoustic classification of multiple simultaneous bird species: a multi-instance multi-label approach The Journal of the Acoustical Society of America. 2012; 131 (6): 4640-4650. doi:10.1121/1.4707424

[6] Scott S, Zhang J, Brown J. On generalized multiple-instance learning International Journal of Computational Intelligence and Applications. 2005; 5 (01): 21-35.

[7] Xu X. Statistical Learning in Multiple Instance Problems Hamilton, New Zealand: The University of Waikato; 2003.

[8] Amores J. Multiple instance classification: Review, taxonomy and comparative study Artificial Intelligence 2013; 201: 81-105. doi:10.1016/j.artint.2013.06.003

[9] Andrews S, Tsochantaridis I, Hofmann T. Support vector machines for multiple-instance learning In: Advances in Neural Information Processing Systems 15. MIT Press; 2003: 561-568.

[10] Poursaeidi MH, Kundakcioglu OE. Robust support vector machines for multiple instance learning Annals of Operations Research. 2014; 216 (1): 205-227. doi:10.1007/s10479-012-1241-z

[11] Kandemir M, Zhang C, Hamprecht FA. Empowering multiple instance histopathology cancer diagnosis by cell graphs In: Medical image computing and computer-assisted intervention : MICCAI International Conference on Medical Image Computing and Computer-Assisted Intervention. vol. 17. Springer; 2014. pp. 228–235.

[12] Vanwinckelen G, Tragante do OV, Fierens D, Blockeel H. Instance-level accuracy versus bag-level accuracy in multi-instance learning Data Mining and Knowledge Discovery. 2016; 30 (2): 313–341. doi:10.1007/s10618-015-0416-z

[13] Zhou ZH, Sun YY, Li YF. Multi-instance learning by treating instances as non-I.I.D. samples In: Proceedings of the 26th International Conference On Machine Learning, ICML 2009. ACM; 2009: 1249-1256.

[14] Zhou ZH. Multi-instance learning: A survey Department of Computer Science & Technology, Nanjing University, Tech Rep. 2004.

[15] Wang J, Zucker JD. Solving multiple-ınstance problem: a lazy learning approach In: In Proc. 17th International Conference on Machine Learning; 2000. p. 1119-1125.

[16] Maron O, Lozano-Pérez T. A framework for multiple-instance learning Advances in Neural Information Processing Systems; 1998: 570-576.

[17] Zhang Q, Goldman SA. Em-dd: An improved multiple-instance learning technique In: Advances in Neural Information Processing Systems; 2002: 1073-1080.

[18] Bunescu RC, Mooney RJ. Multiple instance learning for sparse positive bags In: ACM International Conference Proceeding Series. vol. 227. ACM; 2007. pp. 105–112. doi:10.1145/1273496.1273510

[19] Mangasarian OL, Wild EW. Multiple instance classification via successive linear programming Journal of Optimization Theory and Applications 2008; 137 (3): 555-568. doi:10.1007/s10957-007-9343-5

[20] Kundakcioglu OE, Seref O, Pardalos PM. Multiple instance learning via margin maximization Applied Numerical Mathematics. 2010; 60 (4): 358-369. doi:10.1016/j.apnum.2009.05.013 Available from: http://www.sciencedirect.com/science/article/pii/S0168927409001007

[21] Li WJ, Yeung DY. MILD: Multiple-instance learning via disambiguation IEEE Transactions on Knowledge and Data Engineering 2010; 22 (1): 76-89. doi:10.1109/TKDE.2009.58

[22] Erdem A, Erdem E. Multiple-instance learning with instance selection via dominant sets In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 7005 LNCS. Springer; 2011: 177-191. doi:10.1007/978-3-642-24471-1_13

[23] Tax DMJ, Hendriks E, Valstar MF, Pantic M. The detection of concept frames using clustering multi-instance learning In: Proceedings - International Conference on Pattern Recognition. IEEE; 2010: 2917-2920. doi:10.1109/ICPR.2010.715

[24] Gärtner T, Flach PA, Kowalczyk A, Smola AJ. Multi-instance Kernels. In: Proceedings of the Nineteenth International Conference on Machine Learning; 2002. pp. 179-186.

[25] Cheplygina V, Tax DMJ, Loog M. Multiple instance learning with bag dissimilarities Pattern Recognition 2015; 48 (1): 264-275. doi:10.1016/j.patcog.2014.07.022

[26] Pahikkala T, Airola A, Suominen H, Boberg J, Salakoski T. Efficient AUC maximization with regularized least-squares Frontiers in Artificial Intelligence and Applications; 2008 (173): 12-19.

[27] Mann HB, Whitney DR. On a test of whether one of two random variables is stochastically larger than the other The Annals of Mathematical Statistics 1947; 18 (1): 50-60. doi:10.1214/aoms/1177730491

[28] Ataman K, Street WN, Zhang Y. Learning to rank by maximizing AUC with linear programming In: IEEE International Conference on Neural Networks - Conference Proceedings. IEEE; 2006. pp. 123-129. doi:10.1109/ijcnn.2006.246669

[29] Fu Z, Robles-Kelly A, Zhou J. MILIS: Multiple instance learning with instance selection IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011; 33 (5): 958–977. doi:10.1109/TPAMI.2010.155

[30] Duin RPW. The dissimilarity representation for pattern recognition , a tutorial vol. 64. World Scientific; 2009.

[31] Zhou ZH, Zhang ML. Solving multi-instance problems with classifier ensemble based on constructive clustering Knowledge and Information Systems 2007; 11 (2): 155-170. doi:10.1007/s10115-006-0029-3

[32] Gurobi Optimization Inc Gurobi optimizer reference manual; 2014. Available from:http://www.gurobi.com

[33] Pedregosa F, Varoquaux G, Buitinck L, Louppe G, Grisel O et al. Scikit-learn: Machine Learning in Python Journal of Machine Learning Research. 2015;19 (1):29–33.

[34] Küçükaşcı ES, Baydoğan MG. Bag encoding strategies in multiple instance learning problems; 2018. doi:10.1016/j.ins.2018.08.020 Available from: http://ww3.ticaret.edu.tr/eskucukasci/multiple-instance-learning/

[35] MATLAB version 8.5.0.197613 (R2015a) Natick, Massachusetts; 2015.

[36] Wei XS, Wu J, Zhou ZH. Scalable algorithms for multi-instance learning IEEE Transactions on Neural Networks and Learning Systems. 2017;28(4):975–987. doi:10.1109/TNNLS.2016.2519102

[37] Majnik M, Bosnić Z. ROC analysis of classifiers in machine learning: A survey Intelligent Data Analysis. 2013;17 (3):531–558. doi: 10.3233/IDA-130592

[38] Ling CX, Huang J, Zhang H. AUC: A better measure than accuracy in comparing learning algorithms In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 2671. Springer; 2003. p. 329–341. doi: 10.1007/3-540-44886-1_25

[39] Carbonneau MA, Granger E, Raymond AJ, Gagnon G. Robust multiple-instance learning ensembles using random subspace instance selection Pattern Recognition. 2016;58:83–99. doi:10.1016/j.patcog.2016.03.035

[40] Demšar J. Statistical comparisons of classifiers over multiple data sets Journal of Machine Learning Research. 2006;7:1–30.

[41] Friedman M. A Comparison of Alternative Tests of Significance for the Problem of m Rankings The Annals of Mathematical Statistics. 1940;11 (1):86–92. doi:10.1214/aoms/1177731944

[42] Nemenyi P. Distribution-free Multiple Comparisons Princeton University; 1963.

[43] Nelder JA, Mead R. A Simplex Method for Function Minimization The Computer Journal. 1965;7 (4):308–313. doi:10.1093/comjnl/7.4.308

[44] Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ. LIBLINEAR: A library for large linear classification Journal of Machine Learning Research. 2008;9(8):1871–1874. doi:10.1145/1390681.1442794

[45] Chang CC, Lin CJ. LIBSVM: A Library for support vector machines ACM Transactions on Intelligent Systems and Technology. 2011;2 (3):27. doi:10.1145/1961189.1961199