Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Optimizing Bag Features for Multiple-Instance Retrieval

Zhouyu Fu[1], **Feifei Pan**[2], **Cheng Deng**[3], **Wei Liu**[4]
[1]School of Computing, Engineering & Mathematics, University of Western Sydney, NSW, Australia
[2]New York Institute of Technology, New York, NY, USA
[3]School of Electronic Engineering, Xidian University, Xi'an, China
[4]IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
z.fu@uws.edu.au, fpan@nyit.edu, chdeng@mail.xidian.edu.cn, weiliu@us.ibm.com

Abstract

Multiple-Instance (MI) learning is an important supervised learning technique which deals with collections of instances called bags. While existing research in MI learning mainly focused on classification, in this paper we propose a new approach for MI retrieval to enable effective similarity retrieval of bags of instances, where training data is presented in the form of similar and dissimilar bag pairs. An embedded scheme is devised as encoding each bag into a single bag feature vector by exploiting a similarity-based transformation. In this way, the original MI problem is converted into a single-instance version. Furthermore, we develop a principled approach for optimizing bag features specific to similarity retrieval through leveraging pairwise label information at the bag level. The experimental results demonstrate the effectiveness of the proposed approach in comparison with the alternatives for MI retrieval.

Introduction

Multiple-Instance (MI) learning is an active research area in machine learning, which deals with classification of bags of instances (Dietterich, Lathrop, and Lozano-perez 1997). In a standard MI problem, each training example is a bag containing a number of instances. Each instance is represented by a feature vector. Both bags and instances have class labels, but only bag labels are available for the learning task. Bag and instance labels are related by the MI assumption which states that each positive bag should contain at least one positive instance, whereas all instances are negative in a negative bag.

Many real-world applications can be naturally cast to MI learning problems. One example is content-based image retrieval (CBIR) (Chen, Bi, and Wang 2006; Li et al. 2009; 2011), where every image (bag) is segmented into different regions (instances) and features are extracted from each region. An irrelevant (negative) image contains only irrelevant regions, whereas a relevant (positive) image contains at least a relevant region and possibly irrelevant regions. Other application areas for MI learning are drug activity detection, visual tracking and audio retrieval (Dietterich, Lathrop, and Lozano-perez 1997; Babenko, Yang, and Belongie 2011; Fu et al. 2013).

Despite the importance of MI learning, existing methods mainly focused on the classification task. To apply these techniques to the retrieval task, one has to cast retrieval to classification by treating examples in the same class as similar objects and those from different classes as dissimilar ones. This is overly restrictive for practical applications, as it is not always feasible to know the class labels of training examples. For retrieval, we only need to know the similarity of two objects instead of the class memberships.

In this paper, we address MI retrieval, a new problem in MI learning that deals with the retrieval of similar bags of instances. Unlike existing MI classification algorithms that use bag labels for training and return a classifier at output, in MI retrieval, supervised information is provided in the form of similarity labels for bag pairs rather than class labels for individual bags. The purpose of MI retrieval is to learn a bag-level distance metric that can be used for ranking the bags by similarity.

We propose a two step algorithm for MI retrieval. Firstly, we employ a bag feature encoding scheme to convert each bag of instances into a single bag feature vector in accordance with the label constraints of MI assumption. After that, we develop an optimization algorithm to select the prototypes used to construct the bag-level feature vectors. In this way, we can optimize the bag feature vectors which are best aligned with the provided similarity labels for the retrieval task. As a result of optimization, distances between similar bag pairs are minimized while distances between dissimilar pairs are maximized with the learned metric.

The paper has two major contributions in the following:

1. It proposes the first supervised method for similarity retrieval of MI data using only pairwise label information. The method does not require bag labels for training.
2. It develops a prototype optimization algorithm to produce a discriminative bag-level feature encoding for similarity retrieval. The algorithm is able to optimize prototypes freely in the instance space and thus offers more flexibility and strength than existing feature encoding schemes (Chen, Bi, and Wang 2006; Fu, Robles-Kelly, and Zhou 2011) that are only able to pick prototypes directly from the training instances.

Related Work

MI classification can be addressed at either instance or bag level. Instance-level methods provide a bottom-up approach, where models are trained for predicting instance labels. Bag predictions can then be made by aggregating instance prediction results based on the MI assumption. This leads to a latent variable problem as instance labels are not observed and need to be inferred from the training data and bag labels. While early methods on MI learning use various heuristics to locate true positive regions in instance feature space (Dietterich, Lathrop, and Lozano-perez 1997; Maron and Lozano-Perez 1998; Zhang and Goldman 2002), more recent instance-level methods attempt to adapt standard classification techniques to the MI setting, most notably the Support Vector Machine (SVM) (Andrews, Tsochantaridis, and Hofmann 2003; Cheung and Kwok 2006; Jia and Zhang 2008; Li et al. 2009; Li and Sminchisescu 2010; Li et al. 2011; Wang et al. 2011) and ensemble methods (Zhou and Zhang 2007; Babenko, Yang, and Belongie 2011; Doran and Ray 2013).

Instance-level methods usually involve solving a non-trivial inference problem on instance labels. This often leads to complex mixed-integer type of optimization. To circumvent this issue, bag-level classifiers have been proposed as alternatives. These methods take a top-down approach to MI classification by creating a bag-level representation that preserves the label constraints. The representation could be in the form of bag feature vectors (Chen, Bi, and Wang 2006; Fu, Robles-Kelly, and Zhou 2011), distance metrics (Wang and Zucker 2000), or kernels (Gartner et al. 2002; Tao et al. 2008; Zhou, Sun, and Li 2009). Consequently, the original MI problem is converted into a single-instance problem that can be solved using standard techniques.

Despite the large number of methods proposed for MI learning, they are all focused on either supervised or semi-supervised classification tasks (Jia and Zhang 2008) and require at least bag labels for classifier training. This makes them infeasible for the MI retrieval task discussed in this paper, as neither instance nor bag labels are available for the learning task. The only information given is the similarity of each bag pair selected for training.

Another technique related to both MI learning and retrieval is the MI metric learning (Jin, Wang, and Zhou 2009; Guillaumin, Verbeek, and Schmid 2010). Its purpose is to learn an instance-level metric that improves the performance of bag-level retrieval based on either standard or citation K-Nearest Neighbor (KNN) models tailored to MI classification setting (Wang and Zucker 2000). Again, bag labels are required in MI metric learning for the inference of instance labels and optimization of instance-level metrics. This is very different from the proposed MI retrieval algorithm, which does not require bag labels for training and operates in bag feature space directly for similarity retrieval.

Optimal Feature Embedding for MI Retrieval

Preliminaries and Overview of Approach

In this section, we define the MI retrieval problem and discuss how it is related to classification. After that, we present an overview of our approach to MI retrieval.

Let $\{\mathcal{X}_\iota|_{\iota=1}^{l}\}$ denote a MI data set composed of l bags. Each bag $\mathcal{X}_\iota = \{\mathbf{x}_{\iota,p}\}_{p=1}^{n_\iota}$ contains a set of instances represented by instance feature vectors $\mathbf{x}_{\iota,p}$, where ι and p denote bag and instance indices, and n_ι is the number of instances in bag ι. Let y_ι and $y_{\iota,p}$ denote the label value for bag ι and the pth instance in bag ι respectively, where $y_\iota = 1$ for positive bag and $y_\iota = 0$ for negative one. By the MI assumption, instance and bag labels are related by the following constraints

$$y_\iota = \max_{p=1}^{n_\iota} y_{\iota,p} \qquad \forall \iota \tag{1}$$

At training stage, only bag labels are provided and instance labels are not available.

Multiclass MI learning can be defined in a similar fashion, by treating it as multiple binary problems and using a single label variable for each class. Specifically, let y_ι^c denote the label value for bag ι and class c, with $y_\iota^c = 1$ if bag ι belongs to class c and $y_\iota^c = 0$ otherwise. For each class c, bag labels y_ι^c and corresponding instance labels $y_{\iota,p}^c$ are related by the same constraints in eq. (1).

MI retrieval is different from the classification problem above on how supervised information is provided. The purpose of retrieval is to determine whether two objects are similar or not. It can be treated as a binary classification problem on pairs of objects, with positive labels for similar pairs and negative labels for dissimilar pairs. Hence for MI retrieval training, given a data set $\{\mathcal{X}_\iota|_{\iota=1}^{l}\}$, we can randomly sample a set of bag pairs given by the index set $\mathcal{S} = \{(i,j)|i,j = 1,\ldots,l, i \neq j\}$. The corresponding label set is $\{s_{i,j}\}_{(i,j)\in\mathcal{S}}$, where $s_{i,j}$ is the similarity label for bag pair (i,j). $s_{i,j} = 1$ if bags i and j are similar and $s_{i,j} = 0$ if they are dissimilar. It can be seen that similarity labels $s_{i,j}$ are implicitly related to bag labels by

$$s_{i,j} = sgn(\mathbf{y}_i^t \mathbf{y}_j) \tag{2}$$

where $\mathbf{y}_i = [y_i^1, \ldots, y_i^r]$ is the label vector for bag i, r is the number of classes, sgn is the sign function taking the value of 1 for positive input and 0 otherwise. Thus, two bags are similar if they belong to the same class and share true positive instances from that class.

Clearly, MI retrieval defines a more generic scenario that includes MI classification as a special case, as one can infer the similarity label on bag pairs given the bag labels on the right-hand-side of eq. 2 but not vice versa. With MI classification, only the instance labels are implicit and bag labels are provided for training. With MI retrieval, neither instance nor bag labels are available for training. Moreover, the number of classes is implicit. This motivates a different approach to MI retrieval instead of applying standard MI learning methods to retrieval.

Due to the presence of too many latent variables in the MI retrieval scenario, we adopt a top-down bag-level approach that sidesteps the difficult inference problems encountered with instance-level methods. Our approach is visualized in Figure 1. First, we leverage a feature encoding scheme that maps each bag of instances into a single feature vector. The feature encoding is parametrized by a set of prototypes in

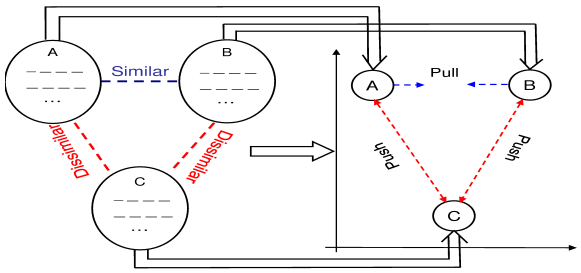


Figure 1: Diagram of proposed MI retrieval solution.

the instance space. Hence one can control the bag feature vectors by purely adjusting the prototypes. To utilize the discriminant label information for bag pairs, the optimal prototypes should return bag feature vectors such that similar bags are pulled together and dissimilar bags are pushed away in the bag feature space, as shown in Figure 1. Consequently, given a bag for query, one can simply search for similar bags in the database based on the distances between bag feature vectors parametrized by the optimized prototypes.

Note that a similar strategy was employed by metric learning (Jain et al. 2012) and supervised hashing (Liu et al. 2012) techniques for learning discriminative feature subspaces and hash functions in single instance scenarios. The proposed approach extends the idea to the MI scenario and provides an integral solution that directly optimizes the bag-level feature encoding for MI retrieval.

Bag Feature Encoding

The label constraints in eq. 1 indicate that only the maximum value is relevant when predicting the bag label from instance prediction results. Similarly, for nearest neighbour search of bags, we only need to consider the closest instances in two bags. This implies the use of the following Hausdorff distance for bag comparisons, which was adopted in the citation KNN framework (Wang and Zucker 2000)

$$d^H_{\mathcal{X}_i,\mathcal{X}_j} = \min_{p=1}^{n_i} \min_{q=1}^{n_j} d_{\mathbf{x}_{i,p},\mathbf{x}_{j,q}} = \min_{p=1}^{n_i} \min_{q=1}^{n_j} \|\mathbf{x}_{i,p} - \mathbf{x}_{j,q}\| \quad (3)$$

Equivalently, we can map the above distance to the similarity metric below using the exponential operator

$$s^H_{\mathcal{X}_i,\mathcal{X}_j} = \max_{p=1}^{n_i} \max_{q=1}^{n_j} e^{-\gamma\|\mathbf{x}_{i,p}-\mathbf{x}_{j,q}\|^2} \quad (4)$$

Despite its simplicity, the above bag similarity metric is quite sensitive to noises and outliers. A single outlier can contaminate the similarity computation. It also lacks the flexibility to adapt to different data distributions. To overcome this issue, we can use a reference point $\mathbf{z}$ for bag similarity calculation in the following

$$\begin{aligned} s'_{\mathcal{X}_i,\mathcal{X}_j} &= \max_{p=1}^{n_i} \max_{q=1}^{n_j} e^{-\gamma(\|\mathbf{x}_{i,p}-\mathbf{z}\|^2+\|\mathbf{x}_{j,q}-\mathbf{z}\|^2)} \quad (5) \\ &= (\max_{p=1}^{n_i} e^{-\gamma\|\mathbf{x}_{i,p}-\mathbf{z}\|^2}) \ (\max_{q=1}^{n_j} e^{-\gamma\|\mathbf{x}_{j,q}-\mathbf{z}\|^2}) \end{aligned}$$

It is easy to verify that the above modified similarity metric provides a lower bound on the metric in eq. 4. Moreover, it can be decomposed into the product of two terms as shown above. Each term only depends on instances in one bag and thus can be treated as a feature value for that bag.

To further improve robustness and flexibility, we can use multiple reference points to encode each bag into a similarity based feature vector in the following

$$\begin{aligned} \tilde{\mathbf{b}}_i &= \frac{1}{\|\mathbf{b}_i\|}\mathbf{b}_i \quad (6) \\ \mathbf{b}_i &= [b_{i,1}, b_{i,2}, \ldots, b_{i,m}] \\ b_{i,u} &= \max_{p=1}^{n_i} e^{-\gamma\|\mathbf{x}_{i,p}-\mathbf{z}_u\|^2} \end{aligned}$$

where $\{\mathbf{z}_u\}_{u=1}^m$ is a set of m reference points chosen to compute the bag feature vectors. These are prototypes in the instance space that can be used to control bag-level feature encoding. $\tilde{\mathbf{b}}_i$ is the normalized feature vector for bag i utilized for bag-level comparison in the retrieval task. It is scaled from the raw bag feature vector $\mathbf{b}_i$ with m components $b_{i,u}$. Each $b_{i,u}$ is a feature component for bag i obtained by decomposing the similarity metric in eq. 5 with prototype $\mathbf{z}_u$. Note that the bag feature vectors defined above are entirely parametrized by the prototypes $\mathbf{z}_u$. This motivates the prototype optimization approach in the following section which produces bag feature vectors that best preserve the discriminant information in pairwise bag labels.

The bag feature representation developed in eq. 6 establishes a generic encoding scheme for converting a MI problem into a single instance one. It reduces to the MILES encoding scheme (Chen, Bi, and Wang 2006; Fu, Robles-Kelly, and Zhou 2011) when unnormalized feature values are used. Normalization is crucial for the proposed method here. Firstly, for L_2 normalized feature vectors, we can directly relate distance to correlation and develop a simpler formulation for prototype optimization. We can also eliminate the undesirable scaling effect in raw similarity values with feature normalization. Moreover, both encoding schemes discussed in (Chen, Bi, and Wang 2006; Fu, Robles-Kelly, and Zhou 2011) are only able to select prototypes directly from training instances. On the other hand, our prototype optimization algorithm can directly optimize the locations of prototypes by solving a continuous optimization problem, thus offering more flexibility in the resulting bag-level feature encoding.

The proposed feature encoding scheme is also more efficient than bag distance comparison. To compute the minimum distance in eq. 3, one needs to enumerate all pairs of instances between two bags. Plus, all training instances need to be stored in the database for future queries. This is $O(n^2d)$ time complexity for a single operation of distance calculation and $O(lnd)$ space complexity for data storage, where d is the dimension of instance space. In contrast, calculating the bag feature vector is $O(nmd)$ for each bag, but this step can be done offline. It takes $O(lm)$ space to store all pre-computed feature vectors, and a single distance calculation only costs $O(m)$ time. Both are much lower than the case of direct bag distance comparison.

Prototype Optimization

The quality of bag feature vectors derived in eq. 6 completely depends on the choice of prototypes used for calculating the feature values. Choosing good prototypes to produce a discriminative feature encoding for the retrieval task is a non-trivial issue. It can potentially get more complicated for real-world MI retrieval problems with multiple classes and multimodal instance distributions. Furthermore, the lack of explicit class membership information makes it hard to apply density based methods such as diverse density (Maron and Lozano-Perez 1998) or kernel density estimation (Fu, Robles-Kelly, and Zhou 2011) to locate good prototypes in the instance space.

In this section, we devise a prototype optimization algorithm to solve the above problem. The proposed algorithm only uses the similarity labels as supervised information to guide the selection of prototypes that achieves maximal discriminability between similar and dissimilar bag pairs. Ideally, the chosen prototypes should produce bag feature vectors such that the feature vectors for similar bag pairs are pulled together while the feature vectors for dissimilar bag pairs are pushed away in the bag feature space.

The use of L_2 normalization in eq. 6 offers a simple solution. Since all bag feature vectors are located on the unit ball and each feature component $s_{i,u}$ takes positive values, we can easily verify that the correlation values between any two bag-level feature vectors are bounded in the range of $[0, 1]$. Therefore, the correlation values can be utilized to reliably measure the degree of similarity. A correlation value close to 1 indicates the bags are similar, and a correlation value close to 0 indicates dissimilarity. This leads to the following least squares formulation for prototype optimization

$$\min_{\mathbf{z}} Q(\mathbf{z}) = \sum_{(i,j)\in\mathcal{S}} \left(\rho_{i,j} - s_{i,j}\right)^2 \tag{7}$$

where $\rho_{i,j} = \tilde{\mathbf{b}}_i^T \tilde{\mathbf{b}}_j$ is the shorthand for the correlation value between the normalized bag feature vectors for bag i and bag j defined in eq. 6. The similarity label $s_{i,j}$ takes the value of 1 for a similar pair and 0 for dissimilar pair. By minimizing the above cost function, the correlations between similar pairs are maximized by driving their values towards 1, whereas the correlations between dissimilar pairs are minimized by driving their values towards 0.

With normalized feature vectors, correlations are directly related to the distances. Specifically, the following one-to-one map between correlation and distance holds

$$d_{i,j} = \|\tilde{\mathbf{b}}_i - \tilde{\mathbf{b}}_j\| = \sqrt{2 - 2\rho_{i,j}}$$

Distances between different bag pairs are also bounded in the range $[0, \sqrt{2}]$. A smaller distance between two bags indicates a stronger correlation and a similar pair, while a larger distance indicates otherwise.

Hence, we can also define an equivalent optimization problem to eq. 7 in the form of distances as follows

$$\min_{\mathbf{z}} Q^d(\mathbf{z}) = \sum_{(i,j)} \left(d_{i,j} - \sqrt{2 - 2s_{i,j}}\right)^2 \tag{8}$$

By optimizing the cost function above, similar bag pairs are pulled together in the bag feature space while dissimilar pairs are pushed away.

The distance-based cost function defined above relates our prototype optimization algorithm to metric learning techniques (Jain et al. 2012), as both aim to minimize the distances between similar examples and maximize the distances between dissimilar ones. Although it is possible to apply metric learning to fixed bag feature vectors induced from predefined prototypes, the performance relies heavily on the quality of feature vectors and the choice of prototypes in the feature encoding step. On the other hand, the proposed prototype optimization approach aims to directly optimize the bag feature encoding in a supervised fashion by utilizing the similarity labels between bag pairs and thus is able to maximally preserve the discriminative information in training.

In the following, we still solve the correlation based problem formulation, as it leads to simpler gradient calculation. By taking the derivative of the cost function in eq. 7 with respect to each $\mathbf{z}_u$, we have

$$\begin{aligned}
\frac{\partial Q(\mathbf{z})}{\partial \mathbf{z}_u} &= \sum_{(i,j)} (\rho_{i,j} - s_{i,j}) \sum_v \left(\frac{\partial \tilde{b}_{i,v}}{\partial \mathbf{z}_u} \tilde{b}_{j,v} + \frac{\partial \tilde{b}_{j,v}}{\partial \mathbf{z}_u} \tilde{b}_{i,v} \right) \\
\frac{\partial \tilde{b}_{i,v}}{\partial \mathbf{z}_u} &= \begin{cases} \dfrac{1}{\|\mathbf{b}_i\|} \dfrac{\partial b_{i,u}}{\partial \mathbf{z}_u} - \dfrac{\tilde{b}_{i,v}\tilde{b}_{i,u}}{\|\mathbf{b}_i\|} \dfrac{\partial b_{i,u}}{\partial \mathbf{z}_u} & \forall v = u \\ -\dfrac{\tilde{b}_{i,v}\tilde{b}_{i,u}}{\|\mathbf{b}_i\|} \dfrac{\partial b_{i,u}}{\partial \mathbf{z}_u} & \forall v \neq u \end{cases}
\end{aligned} \tag{9}$$

Since $b_{i,v}$ is obtained from a max operator in eq. 6, it is not differentiable with respect to $\mathbf{z}_u$. To calculate the term $\frac{\partial b_{i,u}}{\partial \mathbf{z}_u}$, we use the sub-derivative instead, by simply treating the feature value $s_{i,v}$ equal to the maximum value on the right hand side of the equation in each iteration. The resulting partial derivative is then given by

$$\begin{aligned}
\frac{\partial b_{i,u}}{\partial \mathbf{z}_u} &= 2\gamma s_{i,u} (\mathbf{x}^*_{i,u} - \mathbf{z}_u) \\
\mathbf{x}^*_{i,u} &= \arg\max_{\mathbf{x}\in\mathcal{X}_i} e^{-\gamma\|\mathbf{x}-\mathbf{z}_u\|^2} = \arg\min_{\mathbf{x}\in\mathcal{X}_i} \|\mathbf{x} - \mathbf{z}_u\|^2
\end{aligned} \tag{10}$$

where $\mathbf{x}^*_{i,u}$ denotes the nearest neighbor of prototype $\mathbf{z}_u$ among all instances in bag i . Based on the feature encoding scheme presented above, only the closest instance contributes to the corresponding feature value and needs to be considered in derivative calculation. However, note that the nearest instance might change over the iterations. Thus $\mathbf{x}^*_{i,u}$ values need to be updated for each iteration.

By plugging eq. 10 into eq. 9 and making necessary simplifications, we can obtain the final partial sub-derivative with respect to $\mathbf{z}_u$ in the following

$$\begin{aligned}
\frac{\partial Q(\mathbf{z})}{\partial \mathbf{z}_u} =& 4\gamma \sum_{i,j} (\rho_{i,j} - s_{i,j}) [\beta_{i,j,u} (\mathbf{x}^*_{i,u} - \mathbf{z}_u) \\
&+ \beta_{j,i,u} (\mathbf{x}^*_{j,u} - \mathbf{z}_u)] \\
\beta_{i,j,u} \triangleq& \tilde{b}_{i,u}\tilde{b}_{j,u} - \rho_{i,j}\tilde{b}^2_{i,u}
\end{aligned} \tag{11}$$

Given the sub-gradients, we can then employ any first-order iterative optimization algorithm to solve the formulated problem. In this paper, we used a conjugate gradient method with a maximum of 100 iterations. Compared to standard gradient descent, conjugate gradient method enjoys faster convergence and can also be applied to non-differentiable objective function as long as the sub-gradient exists (Wolfe 1975).

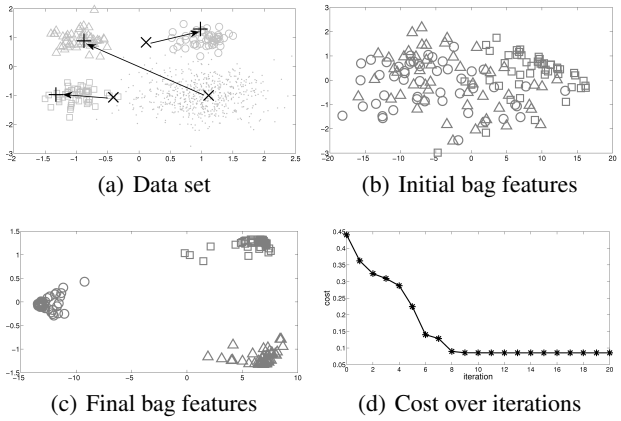
As the cost function is non-convex, proper initialization is crucial to avoid obtaining poor solutions. A reasonable choice is to apply a clustering algorithm on the training instances and select the cluster centroids as the initial prototypes. Compared with random selection of training instances, clustering is more appropriate since the cluster centroids roughly spread over the instance space and cover the modes of the training instance distribution.

Experiments

Synthetic Data

In our first experiment, we demonstrate the effectiveness of prototype optimization for bag-level feature embedding with synthetic data. A three-class multiple instance data set was created and shown in Figure 2(a). The data set contains 50 bags from each of the three classes. Each bag contains one positive and four negative instances. The negative instances for all bags were generated from a Gaussian distribution $N([1,-1]^T, 0.5^2)$, and the positive instances for each class were generated from Gaussian distributions $N([1,1]^T, 0.25^2)$, $N([-1,1]^T, 0.25^2)$, and $N([-1,-1]^T, 0.25^2)$ respectively. The positive instances from three different classes were marked by circles, triangles and squares in the plot, and the negative instances were marked by dots. Intuitively, this problem can be solved using bag feature vectors computed from three prototypes located at the centers of the three positive distributions. However, the initial prototypes returned by cluster centroids, marked by 'x' signs in the plot, were far from the desirable configuration. This is because negative instances dominate in the training instances, making the cluster centroids bias towards the negative distribution.

To optimize over the initial prototypes, we randomly generated 10 similar and 10 dissimilar constraints from each bag based on class labels. Final prototypes returned by the optimization algorithm were marked by '+' signs in the same plot. We notice that the optimized prototypes cover all the positive classes and are close to the centers of the Gaussian distributions from which positive data were generated. To compare the effectiveness of prototype optimization, we also visualize the bag feature vectors for the initial and optimized prototypes in Figures 2(b) and 2(c) respectively. Principle component analysis was used to project the original three-dimensional feature vectors into the two-dimensional plane in the two plots. The bag feature vectors produced by the optimized prototypes clearly outperform those produced by the initial prototypes by pulling together features from similar bags and pushing away features from dissimilar bags. Figure 2(d) shows that the cost function value decreases monotonically over the iterations. The reductions are more significant for the first few iterations, indicating the effectiveness of prototype optimization.



(a) Data set (b) Initial bag features

(c) Final bag features (d) Cost over iterations

Figure 2: Prototype optimization on synthetic data set.

Benchmark Data

Data Sets We used four real-world benchmark data sets in our experiment, including two data sets on drug activity prediction (*MUSK1* and *MUSK2*), two data sets on image categorization (*COREL-10* and *COREL-20*). *MUSK1* and *MUSK2* were first introduced in (Dietterich, Lathrop, and Lozano-perez 1997) and have been widely used as the standard benchmarks for MI learning. *MUSK1* contains 47 positive bags and 45 negative ones, with an average of 5.2 instances in each bag. *MUSK2* contains 39 positive bags and 63 negative ones, with an average of 64.7 instances in each bag. *COREL-10* and *COREL-20* were from the COREL image collection and introduced in (Chen, Bi, and Wang 2006). *COREL-20* contains 2000 JPEG images from 20 categories, and *COREL-10* contains 1000 images from 10 categories. Each image is segmented into 2-13 regions from which texture features are extracted, which give a bag of instances representation at image level.

Methods Existing supervised methods in MI classification all require bag labels for classifier training. This is not the case for the MI retrieval scenario as only similarity labels are provided for training. Hence, it is impossible to apply these methods directly to the MI retrieval task. To this end, we employed the following unsupervised methods as baselines and compared them with the proposed supervised MI retrieval method on the retrieval performance.

- *minDist* - the Hausdorff distance defined in eq. 3. This is the same bag distance metric used by the Citation KNN classifier (Wang and Zucker 2000).
- *meanSim* - the similarity metric modified from eq. 4 by replacing the max with the mean operator. Note *meanSim* is equivalent to the MI kernel developed in (Gartner et al. 2002) for the classification problem.
- *bagFeat* - distances derived from bag feature vectors defined in eq. 6. Based on how prototypes are chosen for feature encoding, we have three variants: *bagFeatR* for

random prototypes, *bagFeatC* for prototypes chosen from cluster centroids, *bagFeatO* is our proposed supervised method with optimized prototypes.

Experiment Setup and Evaluation For the COREL data sets, we randomly split each data set into two equal halves as the training and query sets. For the smaller MUSK data sets, we adopted 10-fold validation for each single training and query round. Since bag labels are available for all four data sets, we synthesized the pairwise labels by randomly selecting 10 bags from the same and different classes respectively to form similar and dissimilar pairs for each bag in the training set. We then discarded the bag labels and used the labeled pairs only for training. For *bagFeat*, the γ parameter in eq. 6 was empirically set to the inverse of the instance feature dimension. Moreover, we tested different number of prototypes (32, 64 and 128) for all three bagFeat variants.

For each method and data set, the experiment was repeated 10 times using different random data partitions. Query results for each method are ranked in increasing distances or decreasing similarities. To evaluate the retrieval performance, we calculated the mean average precision (MAP) and precision for top-K retrievals (PrecK, with K=10 in our experiment) from the query results.

$$Prec@K = \frac{1}{l_q} \sum_{j=1}^{l_q} \sum_{i=1}^{K} \frac{s_{p_i^j, j}}{K} \tag{12}$$

$$MAP = \frac{1}{l_q} \sum_{j=1}^{l_q} \sum_{k=1}^{l} \sum_{i=1}^{k} \mathbb{1}_{s_{p_k^j, j}=1} \frac{s_{p_i^j, j}}{k} \tag{13}$$

where i and j denote the indices of training and query objects, l_q is the number of queries, p_k^j denotes the rank of the kth bag for querying bag j in the query set. Note the MAP measure defined above is calculated by matching individual query results directly with similarity labels. It is different from the MAP measure for classification (Jin, Wang, and Zhou 2009), which evaluates query results by checking their consistency with ground truth class labels and is not applicable to the MI retrieval setting. The MAP measure used here does not require bag labels and can be calculated solely based on the input bags and similarities.

Results Table 1 shows the retrieval results obtained by different methods on benchmark data. From the table, we can see that feature encoding based methods obtain relatively better performance than bag-level distance or similarity comparison. *meanSim* performs rather poorly on all data sets, whereas the performance of *minDist* is on par with *bagFeatR*. For the *bagFeat* variants, the proposed *bagFeatO* returns the best results on all data sets, achieving quite a significant gain over all other methods in both evaluation measures. This empirically demonstrates the effectiveness of prototype optimization for MI retrieval. Furthermore, the performance gaps between different number of prototypes are not as significant as the gaps between different initializations for *bagFeat*. We can see that the proposed *bagFeatO* with 32 prototypes still produces far better results than *bagFeatR* and *bagFeatC* with 64 or 128 prototypes.

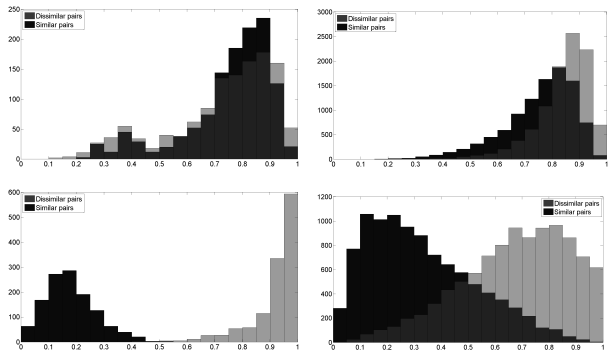


Figure 3: Distance distributions for similar (in dark shade) and dissimilar (in bright shade) pairs on MUSK2 (left) and COREL-20 (right) data sets. Top row: histograms for *bagFeatC*; bottom row: histograms for *bagFeatO*.

To further visualize the effectiveness of prototype optimization, we made histogram plots to capture the distance distributions for similar and dissimilar pairs on the MUSK2 and COREL-20 data sets in Figure 3, where the top and bottom rows in the figure show the distance distributions before and after optimization. We can clearly see that distances for similar pairs are driven towards the lower bound of 0 and distances for dissimilar pairs are driven towards the upper bound of 1 after optimization. It is particularly obvious for the MUSK2 data, which explains the great competitive edge *bagFeatO* gains over other methods on this data set.

Conclusions

In this paper, we proposed the first approach for MI retrieval with similarity labels. Our approach converts bags of instances into single feature vectors using a feature encoding scheme, and employs a prototype optimization algorithm to produce optimized bag features for similarity retrieval.

The proposed approach relies on unambiguous notion of similarity. This is not the case for multi-labeled data, where each example can have multiple class labels. Intensive research was done recently in the area of multi-instance multi-label (MIML) learning (Zhou et al. 2012; Nguyen et al. 2014). It would be interesting to extend our approach to the MIML retrieval scenario in the future.

Another interesting future direction is to investigate the use of an overcomplete set of prototypes for bag feature encoding and the metric learning techniques for dimensionality reduction (Jain et al. 2012). A larger number of prototypes helps retain the discriminant information in training with increased computation cost. Hence, one needs to carefully balance the trade-off between efficiency and performance.

Acknowledgments

The authors would like to thank Dr. Fuxin Li for proofreading an earlier version of the manuscript and making valuable comments that help improve the quality of the paper.

Table 1: Performance comparison of different methods for MI retrieval on benchmark data.

Methods		Prec@10				MAP			
		MUSK1	MUSK2	COREL-10	COREL-20	MUSK1	MUSK2	COREL-10	COREL-20
minDist		0.681	0.630	0.496	0.330	0.628	0.606	0.385	0.227
meanSim		0.641	0.546	0.271	0.159	0.621	0.588	0.227	0.128
$m = 32$	bagFeatR	0.625	0.566	0.510	0.325	0.591	0.571	0.368	0.204
	bagFeatC	0.660	0.588	0.544	0.348	0.610	0.576	0.392	0.219
	bagFeatO	**0.969**	**0.871**	**0.671**	**0.449**	**0.956**	**0.885**	**0.588**	**0.350**
$m = 64$	bagFeatR	0.631	0.564	0.535	0.340	0.599	0.571	0.382	0.211
	bagFeatC	0.639	0.586	0.547	0.354	0.601	0.579	0.390	0.217
	bagFeatO	**0.941**	**0.880**	**0.667**	**0.463**	**0.942**	**0.895**	**0.591**	**0.358**
$m = 128$	bagFeatR	0.638	0.562	0.551	0.356	0.607	0.571	0.393	0.220
	bagFeatC	0.639	0.576	0.554	0.364	0.610	0.576	0.397	0.226
	bagFeatO	**0.964**	**0.885**	**0.713**	**0.495**	**0.949**	**0.892**	**0.628**	**0.383**

References

Andrews, S.; Tsochantaridis, I.; and Hofmann, T. 2003. Support vector machines for multiple-instance learning. In *NIPS*, 561–568.

Babenko, B.; Yang, M.-H.; and Belongie, S. 2011. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Analysis and Machine Intelligence* 33(8):1619–1632.

Chen, Y.; Bi, J.; and Wang, J. 2006. Miles: Multiple-instance learning via embedded instance selection. *IEEE Trans. Pattern Analysis and Machine Intelligence* 28(12):1931–1947.

Cheung, P.-M., and Kwok, J. T.-Y. 2006. A regularization framework for multiple-instance learning. In *ICML*.

Dietterich, T. G.; Lathrop, R. H.; and Lozano-perez, T. 1997. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence* 89:31–71.

Doran, G., and Ray, S. 2013. Smile: Shuffled multiple-instance learning. In *AAAI*.

Fu, Z.; Lu, G.; Ting, K.-M.; and Zhang, D. 2013. Learning sparse kernel classifiers for multi-instance classification. *IEEE Trans. Neural Networks & Learning Systems* 24(9):1377–1389.

Fu, Z.; Robles-Kelly, A.; and Zhou, J. 2011. Milis: Multiple instance learning with instance selection. *IEEE Trans. Pattern Anal. and Mach. Intell.* 33(5):958–977.

Gartner, T.; Flach, A.; Kowalczyk, A.; and Smola, A. J. 2002. Multi-instance kernels. In *ICML*, 179–186.

Guillaumin, M.; Verbeek, J.; and Schmid, C. 2010. Multiple instance metric learning from automatically labeled bags of faces. In *ECCV*.

Jain, P.; Kulis, B.; Davis, J.; and Dhillon, I. S. 2012. Metric and kernel learning using a linear transformation. *JMLR* 13:519–547.

Jia, Y., and Zhang, C. 2008. Instance-level semi-supervised multiple instance learning. In *AAAI*.

Jin, R.; Wang, S.; and Zhou, Z.-H. 2009. Learning a distance metric from multi-instance multi-label data. In *CVPR*.

Li, F., and Sminchisescu, C. 2010. Convex multiple-instance learning by estimating likelihood ratio. In *NIPS*.

Li, Y.-F.; Kwok, J. T.; Tsang, I.; and Zhou, Z.-H. 2009. A convex method for locating regions of interest with multi-instance learning. In *ECML*.

Li, W.; Duan, L.; Xu, D.; and Tsang, I. W. 2011. Text-based web image retrieval using progressive multiple instance learning. In *ICCV*.

Liu, W.; Wang, J.; Ji, R.; Jiang, Y.-G.; and Chang, S.-F. 2012. Supervised hashing with kernels. In *CVPR*.

Maron, O., and Lozano-Perez, T. 1998. A framework for multiple-instance learning. In *NIPS*, 570–576.

Nguyen, C.-T.; Wang, X.; Liu, J.; and Zhou, Z.-H. 2014. Labeling complicated objects: Multi-view multi-instance multi-label learning. In *AAAI*.

Tao, Q.; Scott, S.; Vinodchandran, N. V.; Osugi, T.; and Mueller, B. 2008. Kernels for generalized multiple-instance learning. *IEEE Trans. Pattern Analysis and Machine Intelligence* 30(12):2084–2097.

Wang, J., and Zucker, J. D. 2000. Solving the multiple-instance problem: A lazy learning approach. In *ICML*, 1119–1125.

Wang, H.; Huang, H.; Kamangar, F.; Nie, F.; and Ding, C. 2011. Maximum margin multi-instance learning. In *NIPS*.

Wolfe, P. 1975. A method of conjugate subgradients for minimizing nondifferentiable functions. *Mathematical Programming Study* 3:145–173.

Zhang, Q., and Goldman, S. 2002. Em-dd: An improved multiple-instance learning technique. In *NIPS*, 1073–1080.

Zhou, Z.-H., and Zhang, M.-L. 2007. Solving multi-instance problems with classifier ensemble based on constructive clustering. *Knowledge and Information Systems* 11(2):155–170.

Zhou, Z.-H.; Zhang, M.-L.; Huang, S.-J.; and Li, Y.-F. 2012. Multi-instance multi-label learning. *Artificial Intelligence* 176(1):22912320.

Zhou, Z.-H.; Sun, Y.-Y.; and Li, Y.-F. 2009. Multi-instance learning by treating instances as non-i.i.d. samples. In *ICML*.