Discriminative and Consistent Similarities in Instance-Level Multiple Instance Learning

Mohammad Rastegari[1], Hannaneh Hajishirzi[2], Ali Farhadi[2,3]
[1]University of Maryland, [2]University of Wasington, [3]Allen Institute for AI
mrastega@cs.umd.edu, hannaneh,ali@washington.edu

Abstract

In this paper we present a bottom-up method to instance-level Multiple Instance Learning (MIL) that learns to discover positive instances with globally constrained reasoning about local pairwise similarities. We discover positive instances by optimizing for a ranking such that positive (top rank) instances are highly and consistently similar to each other and dissimilar to negative instances. Our approach takes advantage of a discriminative notion of pairwise similarity coupled with a structural cue in the form of a consistency metric that measures the quality of each similarity. We learn a similarity function for every pair of instances in positive bags by how similarly they differ from instances in negative bags, the only certain labels in MIL. Our experiments demonstrate that our method consistently outperforms state-of-the-art MIL methods both at bag-level and instance-level predictions in standard benchmarks, image category recognition, and text categorization datasets.

1. Introduction

Multiple-instance learning (MIL) [13] addresses a variation of classification problems where complete labels of training examples are not available. In the MIL setup, training labels are assigned to *bags* of instances rather than individual instances. In most standard MIL setups, a bag is positive if it contains at least one positive instance, and is negative if all of its instances are negative. The standard task in MIL is to classify unknown bags of instances (e.g., [31, 48]). However, several application domains require instance-level predictions (e.g., [43]). For example, in image segmentation (instances are superpixels, bags are images) the main goal is to find the exact regions of an image that correspond to the objects of interest.

Instance-level MIL has been approached either by a complex joint optimization over bag and instance classifiers (e.g., [1]) or by identifying positive instances followed by bag classification. Latter involves similarity-based reasoning where most methods either use standard similarity functions (e.g., [48]) or learn a global similarity function for all instances (e.g., [43]). Standard similarity functions are not necessarily discriminative (orange dashed links in Fig. 1) and cannot discover common properties among positive instances. Globally learned similarity functions cannot encode different types of similarities that tie together positive instances within groups (Fig. 1).

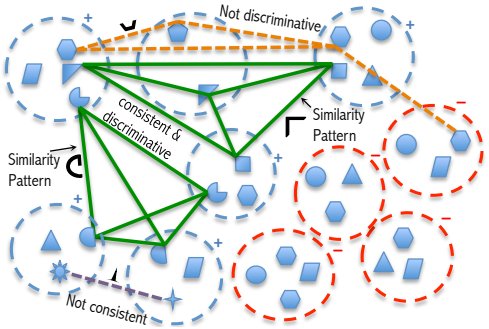

Figure 1. Discriminative and Consistent Similarities are shown by green cliques. Orange similarities are consistent but not discriminative (similar to negatives). Purple similarity is discriminative but not consistent.

In this paper, we introduce a new method for the problem of instance-level MIL with globally-constrained reasoning about local pairwise discriminative similarities. We introduce a novel approach that learns similarity functions specific to each instance and reasons about the underlying structure of similarities between positive instances using our notion of consistent similarities (Green cliques in Fig. 1). We introduce a discriminative notion of similarity that enables learning a similarity function for each pair of instances in positive bags (similarity patterns in Fig. 1). Typically, learning a similarity function requires training labels for similar instances. However, instance-level labels are not available in MIL. We use negative bag labels as the only certain labels in MIL to learn our discriminative similarity function. Instances in positive bags are similar if they are *similarly different* form instances in negative bags.

Pairwise similarities are not always transitive and can be

confused with coincidental patterns in a high-dimensional feature space [38] (Purple dashed links in Fig. 1). For example, two images a and c cannot be similar to each other only because they are similar to another image b; a might be similar to b because both show a sunset over an ocean, and c might be similar to b because of coincidental patterns of similarity. A reliable pairwise similarity should be globally consistent across several pairs (green links in Fig. 1). We introduce a novel clique-based notion of similarity that measures global consistency of pairwise similarities.

We formulate the discovery of positive instances as a ranking problem where top rank instances in positive bags are highly and consistently similar to each other. The bag labels provide constraints to our optimization problem; real positive instances inside each bag should rank higher than negative instances in negative bags. We show that a random-walk based ranking algorithm that uses our globally-consistent pairwise similarities outperforms state-of-the-art MIL results in MIL benchmarks, text categorization, and image segmentation.

Related Work: Over the course of the previous decades several interesting approaches address the problem of bag-level multiple instance learning. Some examples include [31, 32, 44, 45, 1, 42, 11, 4, 6, 29, 46, 16, 37, 45, 10, 48, 5, 39, 2, 36, 17]. Please see [47] for a complete survey; space does not allow a comprehensive literature review. A group of approaches to instance-level MIL use joint optimization over bags and instances. This optimization is modeled in mi-SVM [1] by a max margin formulation and [28] in a convex form. In MILES [9], most discriminative instances are selected by an L1-regularized bag-level classifier. In these settings, solving a joint optimization can lead to an NP-hard problem that has to be approximated. To avoid this problem our method firs discovers positive instances and then uses them to predict bag-level labels.

Similar to our method, other instance-level MIL approaches separate the two steps of discovering positive instances and bag classification. Xiao et al. [43] design a method that assigns two values to instances by measuring their similarity to positive and negative classes and then use a similarity weighted large margin model to learn the final classifier. Jia and Zhang [24] use two different loss functions for negative and positive bags used in a semi-supervised fashion. Fu and Robles [15] introduce an EM-like algorithm that iterates between selecting positive prototypes and updating classifiers. Kim and Torre [25] approach MIL by Gaussian process latent variable models. Deselaers and Ferrari [12] consider each bag as a node in a conditional random field. In contrast, our method takes both local and global information into account through discriminative and consistency similarities between instances. In addition, our method uses a global ranking method for optimization. Due to these properties, our results show consistent improvement over previous MIL methods in different domains.

Discriminative similarities have shown to be effective in the natural language processing community [18, 26] for aligning sentences to events. Unlike our method, previous work does not incorporate consistency of similarities.

2. Overview of Our Method: mi-Sim

Fig. 2 sketches an overview of our method. At training, bags B^{tr} of instances X^{tr} along with bag-level labels b^{tr} are known; Instance-level labels are not given. Our training algorithm has two main steps: discovering "correct" positive instances $\mathcal{L}^+$ among all instances X^{tr^+} in positive bags based on the training bag labels (Fig. 2 Step 1.a) and training a final binary classifier using the discovered positive instances $\mathcal{L}^+$ and instances X^{tr^-} in negative bags in the training set (Fig 2 Step 1.b).

At test time neither bag labels nor instance-level labels are known. We test our method on how well it can predict both bag-level and instance-level labels. For testing (Fig. 2 Step 2), we use the final binary classifier to predict labels of individual instances in the test set. Bag-level labels are then predicted using instance-level labels; a bag is positive if it includes at least one predicted positive instance.

3. Discovering Positive Instances

Training involves discovering positive instances $\mathcal{L}^+$ using bag-level labels in the training set (Fig. 2 Step 1.a). Following most previous works in MIL (e.g., [12, 40]) we assume that negative instances are negative in their own specific ways while positive instances are similar to each other. More intuitively, if positive bags have negative instances that are consistently similar to other negative instances in other positive bags, the instance discovery becomes ill defined. Our experimental results suggest that this is a mild assumption.

We formulate the positive instance discovery problem as the problem of searching for assignments of positive instances that maximize similarities between discovered positive instances. In particular, our goal is to find the best labeling $\mathcal{L}$ to training instances such that discovered positive instances $\mathcal{L}^+$ are highly and consistently similar. The constraint in the optimization enforces each positive bag to have at least one positive instance.

$$\max_{\mathcal{L}} \sum_{x_i,x_j \in \mathcal{L}^+} \mathcal{F}(x_i,x_j) \qquad (1)$$
$$\sum_{k \in B'} \mathcal{L}_k \geq 1 \quad \forall B' \in B^{tr^+}, \;\; \mathcal{L}_l \in \{0,1\} \;\; \forall l \in X^{tr}$$

where $\mathcal{L}_l$ is the predicted label of x_l, $\mathcal{L}^+$ are discovered positive instances, B^{tr^+} corresponds to all positive bags

mi-Sim: Multiple Instance Learning

- Let B^{tr} be the set of training bags where every bag B_i^{tr} has a binary label $b_i \in \{-1, 1\}$
- Let X^{tr^+} and X^{tr^-} be the set of individual instances in positive and negative training bags, respectively
- Let B^{test} be the set of bags at test time and X^{test} be the set of all individual instances in test bags.
- Let $\mathcal{L}^+$ be the set of discovered positive instances at training.

1. **Training Phase**
   (a) $\mathcal{L}^+ \leftarrow$ **Discovering positive instances from training examples** (Sec.3)
      i. **Compute pairwise discriminative similarity** $\mathcal{S}(x_i, x_j)$ (Sec. 3.1)
         A. Train $iSVM$ with x_i as the positive example and X^{tr^-} as negative examples.
         B. $\mathcal{S}(x_i, x_j) \leftarrow$ discriminative similarities between all instances in positive bags using Equation 3
      ii. **Compute consistency similarity** $\mathcal{C}(x_i, x_j)$ (Sec. 3.2)
      iii. **Select positive instances with ranking** (Sec. 3.3)
         A. $CSG \leftarrow$ the graph in which instances x_i and x_j are connected with a weight $\mathcal{S}(x_i, x_j) + \mathcal{C}(x_i, x_j)$
         B. $\mathcal{R}_{x_i} \leftarrow$ the rank of each node x_i in CSG with a random walk algorithm (Equation 6)
         C. $\mathcal{L}^+ \leftarrow$ Pick top ranking instances based on $\mathcal{R}_{x_i}$ in each bag as discovered positive instances
   (b) $SVM^{final} \leftarrow$ Train an SVM with $\mathcal{L}^+$ as positive examples and X^{tr^-} as negative examples.
2. **Testing Phase**
   (a) Instance labes: $\mathcal{L}^+_{test} \leftarrow$ set of test instances classified as positive using SVM^{final}
   (b) Bag labels: for every bag B_i^{test}, assign a positive label if it includes at least one positive instance from $\mathcal{L}^+_{test}$

Figure 2. Our MIL method, mi-Sim, and discovering positive instances at training.

in the training set, and $\mathcal{F}$ is a form of similarity function between pairs of instances x_i and x_j. Below we describe how to model $\mathcal{F}(x_i, x_j)$ as a combination of discriminative $\mathcal{S}(x_i, x_j)$ and consistent similarities $\mathcal{C}(x_i, x_j)$.

Discriminative similarity: Positive instances should not only be similar to each other but also be different from negative instances. Learning a similarity function for positive instances require having a labeled set of positive instances, but instance-level labels are not available. We anchor the learning of our similarity function on the only certain labels in MIL setup; negative bag labels. All instances in the negative bags are known to be negative. We propose to encode the similarity between two positive instances based on *how similarly they differ from negative instances* (Sec. 3.1).

Consistency of similarities: Pairwise similarities are not always transitive, they can be confused with coincidental patterns in the feature space [38]. For example, two images may become similar because they both depict the same scene or may be similar because of some irrelevant accidents in the feature space. Coincidental patterns in the feature space, by definition, are not repetitive. This suggests that a reliable pairwise similarity is the one that is homogeneous across several pairs. We introduce the notion of consistent similarity and measure the consistency of a similarity based on the cardinality of the clique containing both instances (Sec. 3.2).

Positive instance discovery as ranking: We aim to discover assignments of positive instances such that it maximizes the similarities between positive instances, the discrimination between positive and negative instances, and the consistency of similarities between positive instances. Searching for the optimal assignments of instance labels is a challenging optimization problem that is even hard to approximate. We formulate this problem by optimizing for a global ranking $\mathcal{R}_{x_u}$ for each instance x_u such that the top rank instances in each bag jointly maximize the similarity S and consistency C among positive instances:

$$\max_{\mathcal{R}} \sum_{x_i, x_j \in \mathcal{L}^+} \mathcal{S}(x_i, x_j) + \mathcal{C}(x_i, x_j) \qquad (2)$$
$$\mathcal{R}_{x_u} \succ \mathcal{R}_{x_v} \quad \forall x_u \in \mathcal{L}^+, x_v \in (X^{tr^+} \setminus \mathcal{L}^+)$$
$$\mathcal{L}^+ = \cup_n \Delta^{\mathcal{R}}_{B_n^{tr}}$$

where X^{tr^+} corresponds to all the instances in positive bags, $\mathcal{L}^+$ is the discovered set of positive instances, $\mathcal{R}_{x_u}$ is the ranking score of the instance x_u in its bag, $\Delta^{\mathcal{R}}_{B_n^{tr}}$ corresponds to top-ranked instances based on ranking $\mathcal{R}$ in positive bag B_n^{tr}.

Finding the optimal ordering of instances according to the above optimization (Equation 2) is a challenging combinatorial optimization problem (NP-Hard). We approximate the above optimization with a random-walk based ranking that respects both similarity and consistency of similarities (Sec.3.3). To this end, we build a *Consistent Similarity Graph* (CSG) for instances in positive bags in the training set. Nodes in this graph are instances in positive bags and edges represent discriminative pairwise similarities along with their consistency scores. The weight

of an edge between two nodes x_i and x_j correspond to $\mathcal{S}(x_i, x_j) + \mathcal{C}(x_i, x_j)$.

Definition 1 (*Consistent Similarity Graph (CSG)*)**.** A CSG is an undirected weighted graph $CSG = (V, E)$ where a node $x_i \in V$ corresponds to a training instance x_i in a positive bag and an edge $e_{ij} \in E$ connects x_i to x_j, if $\mathcal{S}(x_i, x_j) > 0$. The weight e_{ij} is $\alpha.\mathcal{S}(x_i, x_j) + \mathcal{C}(x_i, x_j)$ where α is the balancing factor that takes into account the differences between the scales of $\mathcal{S}$ and $\mathcal{C}$.

We approximate the best ranking by performing a random walk on this graph. The main intuition is nodes that are adjacent to high rank nodes with large edge scores will get high ranks. There are several results on why a random walk based approach results in decent approximations of the optimal ranking [33, 40, 34, 3, 35, 18, 26].

3.1. Pairwise Discriminative Similarity

In this section we describe how to derive similarity $\mathcal{S}(x_i, x_j)$ between two instances x_i and x_j (Fig.2 step 1.a.i). We base our notion of similarity on using negative labels which are the only certain labels in MIL. We learn what is unique about each instance in positive bags by discriminating it from all instances in the negative bags; these are the examples we are sure they are not similar to positive instances. We learn an instance by what it is *not* like. We do this by fitting an SVM with only one positive instance and a large number of negative instances.

Recently in computer vision, Exemplar Support Vector Machines have shown great success in learning what is unique about an image that can distinguish it from all other images [30]. Despite being susceptible to overfitting, the hard negative mining method in [30] gets away from this issue. By fitting a classifier to only one positive instance and a large number of negative instances we learn how to weight features against each other in a discriminative manner. If a learned model for each instance produces a positive score when applied to another instance, the two instances are considered to be similar.

Based on our discriminative notion of similarity, two instances in positive bags are similar if they are similarly different from negative instances. To setup notations, for an instance $x_i \in X^{tr^+}$, we fit a linear SVM (called iSVM) to x_i as a positive training example and use instances X^{tr^-} in negative bags as negative examples. We balance the positive and negative examples by weighting them accordingly. The confidence of applying the SVM over a new instance x_j is computed as $\Upsilon_{i,j} = w_i^T \cdot x_j$, where w_i is the learned weight vector for instance x_i.

The discriminative similarity between two instances x_i and x_j is defined according to their mutual confidences. The confidence scores are not directly comparable. To calibrate, we use the order of each instance among all the other instances. The similarity of two instances x_i and x_j is $1/\varphi(i,j).\varphi(j,i)$ where $\varphi(i,j)$ represents the order of x_j among all the instances with positive confidence when classified by w_i.

$$\mathcal{S}(x_i, x_j) = \begin{cases} \frac{1}{\varphi(i,j)\cdot\varphi(j,i)} & \text{if } \Upsilon_{i,j} > 0 \text{ and } \Upsilon_{j,i} > 0 \\ 0 & otherwise \end{cases} \tag{3}$$

$\varphi(i,j)$ is the rank of x_j among all the training instances x_k where $\Upsilon_{i,k} > 0$.

3.2. Consistency of Similarities

In this section we describe how to compute the consistency $\mathcal{C}(x_i, x_j)$ of the similarity between two instances x_i and x_j (Fig.2 step 1.a.ii). Pairwise similarities are not always transitive, they can be confused with coincidental patterns in the feature space [38]. This exposes subtleties to reasoning based on pairwise similarities. For example, two images may become similar because they both depict the same scene or may be similar because of some irrelevant accidents in the feature space. These accident happen largely because feature spaces are not perfect representations of the world, and similarities are typically modeled by some sort of a global distance in a high dimensional space. Similarities typically expose several modalities out of which very few are desirable.

Coincidental patterns in the feature space, by definition, are not repetitive. This suggests that a reliable pairwise similarity is the one that is homogeneous across several pairs in a large clique of instances. We define a consistency score between two instances as the size of the largest maximal clique that contains both instances. If a pairwise similarity is consistent then several homogeneous similarities can be found, thus the maximal clique is large in size. If a pairwise similarity is not consistent the corresponding maximal clique is small, resulting in a small consistency score. In CSG the nodes are all the samples from positive bags and edges are only between nodes that their discriminative similarity are greater than zero.

The notion of a clique is slightly rigid for pairwise similarities due to uncertainties in our discriminative similarity measure and inherent variations among positive instances. For these reasons we use the notion of *quasi-cliques* [8] that relaxes the constraint of completeness in cliques.

Definition 2 (*Quasi Clique*)**.** A graph $G = (V, E)$ is a γ-quasi clique if $|E| \geq \lfloor \gamma \binom{|V|}{2} \rfloor$, where $0 < \gamma \leq 1$.

A quasi-clique essentially represents a group of instances that are densely similar to each other. Under settings of MIL, it is natural to assume that positive instances correspond to higher cardinality quasi-cliques with high degrees of inner-clique-similarity.

Definition 3 (*Maximal Quasi-Clique*)**.** A maximal quasi-clique in an undirected graph is a quasi-clique that cannot be extended by adding any more node.

For each subgraph G_i induced by a node x_i there always exists a maximal quasi-clique containing x_i. Every node might appear in more than one maximal quasi-clique.

We adopt a greedy approach [8] to find maximal quasi-cliques in a graph constructed by connecting similar instances in positive bags. For every node x_i in $CSG = (V, E)$, our method iteratively collects a set of vertices $\mathcal{Q}_i$ to form a quasi-clique corresponding to the node x_i. Initially, $\mathcal{Q}_i$ is set to x_i. At each iteration a candidate set of instances $\mathcal{P}_i$ is a set of nodes $x_j \in V \setminus \mathcal{Q}_i$ that are connected to a large portion of current nodes in $\mathcal{Q}_i$.

$$\mathcal{P}_i = \left\{ \forall x_j \notin \mathcal{Q}_i, \sum_{x_k \in \mathcal{Q}_i} 1[e_{kj} > 0] \geq \gamma |\mathcal{Q}_i| \right\} \quad (4)$$

where $1[.]$ is a 0-1 indicator function of its boolean argument. Then, a node $x_j^* \in \mathcal{P}_i$ will be added to $\mathcal{Q}_i$ where $x_j^* = \arg\max_{x_j \in \mathcal{P}_i} \sum_{x_k \in \mathcal{Q}_i} e_{kj}$.

We iterate until there is no new node that can be added to $\mathcal{Q}_i$ i.e., $\mathcal{P}_i = \emptyset$.

Corollary 1. $\mathcal{Q}_i$ generated by the above algorithm returns a maximal γ-quasi-clique.

Finally, we model consistency similarity between two instances x_i and x_j as the size of the largest maximal quasi-clique that includes the edge between x_i and x_j.

Definition 4 (*Consistency Score*)**.** Let e_{ij} be the edge connecting x_i and x_j in the graph. Let $\mathcal{Q}_h$ denote different maximal quasi-cliques that include nodes x_i and x_j. Consistency $\mathcal{C}(x_i, x_j)$ of the edge e_{ij} is the size of the largest maximal quasi-clique that includes both x_i and x_j.

$$\mathcal{C}(x_i, x_j) = \begin{cases} \max_h \{|\mathcal{Q}_h|\} & \forall h \; x_i, x_j \in \mathcal{Q}_h \\ 0 & \nexists h \quad x_i, x_j \in \mathcal{Q}_h \end{cases} \quad (5)$$

3.3. Ranking Instances in CSG

In this section we describe how to rank instances to optimize for Equation 2 (Fig.2 step 1.a.iii). The optimal ranking imposes high consistent similarities among top ranked instances in positive bags. To rank, we perform a random-walk algorithm to propagate scores in the consistent similarity graph. Recall that in CSG, we assign a node for each instance in positive bags. The edge scores are combinations of discriminative similarity score and the consistency score.

We rank nodes in CSG by computing ranking scores of all the instances X^{tr^+} in positive bags. We then select highest-scoring nodes as positive instances. The ranker in CSG should assign high ranks to nodes which are highly and consistently similar to many high rank instances. We adopt a random walk algorithm similar to Google PageRank [7] that encourages propagating scores among edges that have high similarity score.

Our algorithm computes a ranking score $\mathcal{R}_{x_i}$ corresponding to every node in the graph. The algorithm iteratively computes the score of a node according to Equation 6. At every iteration, $\mathcal{R}_{x_i}$ is the expected sum (with probability d) of scores of the adjacent nodes (computed at the previous iteration) and the self confidence value:

$$\mathcal{R}_{x_i} = (1-d)\Upsilon_{i,i} + d \sum_{x_j \in \mathcal{N}(x_i)} \mathcal{R}_{x_j} \cdot e_{ij} \quad (6)$$

where e_{ij} is the normalized weight of an edge in CSG, $\Upsilon_{i,i}$ (as defined in section 3.1) is the confidence of each instance classifier on its own instance, $\mathcal{N}(x_i)$ is the set of adjacent nodes to x_i in CSG, and d is a damping factor. $\mathcal{R}_{x_i}$ is initialized by random values.

At iteration 1, only direct edges (length 1) are considered; $\mathcal{R}_{x_i}$ only adds up the scores of the nodes that are directly linked to x_i. In next iterations, longer paths are considered; the effect of indirectly linked nodes to x_i is included in the scores of adjacent nodes. We control the expected length of the paths with a damping factor d.

Once the scores of each instance is known, the most probable positives in each bag would be the highest scoring ones. In our experiments, we select positive instances if their ranking score is higher than 0.9. Knowing the positive instances in each bag, bag-level and instance-level MIL become straight forward.

4. Experimental Setup

Tasks: We compare our method with bag-level and instance-level state-of-the-art methods in different datasets. Training sets in all the experiments only include bag-level labels. During training our method discovers positive instances in each bag. Once positive instances are discovered we train SVM_{final} using discovered positive instances as positive examples and all the examples in the negative bags as negative instances. At test time, in bag-level classification, bag-level labels are inferred from the instance-level predictions. A bag will be positive if it contains at least one predicted positive instance. In addition, we evaluate our algorithm for positive instance discovery (instance-level predictions).

Datasets: *Benchmark datasets*: We evaluate our method on benchmark datasets for MIL including Drug Activity Prediction (*Musk1*, *Musk2*) and Localized Content-based Image Retrieval (*Elephant*, *Tiger* and *Fox*). More details on these datasets can be found in [1].

COREL-2000: We evaluate our method on COREL-2000, which is a standard dataset for image categorization using

MIL and has been studied by previous work in MIL. It contains 1000 images in 20 categories of COREL. For every image, regions are extracted via segmentation, and each segment represented by a 9-dimensional feature vector. To cast image categorization to an MIL problem, every image is a bag of regions. The regions are considered as instances. An image is positive if at least one of its regions contain the object of interest. More details on this dataset can be found in [10, 9].

IL-MSRC: In order to evaluate instance-level MIL on image categorization tasks we introduce the Instance-level MSRC (IL-MSRC) dataset. We use images of MSRC dataset (30 images per category). We perform segmentation on each image using superpixel extraction of [14]. We then use ground-truth segments provided by MSRC to label each superpixel. A superpixel is positive, if it has more than 50% overlap (pixel area) with ground-truth segment otherwise it is negative. Images are bags and superpixels are instances in each bag.

20 Newsgroup: These datasets include texts from *20 newsgroups* corpora. This dataset has been introduced in miGraph [48] to evaluate the role of MIL in text categorization. Each dataset has 100 bags: 50 positive and 50 negative. For every category, every positive bag includes a few news which are randomly selected for that category and many unrelated news. The main characteristic of this dataset is that instance-level labels are available and it has low witness rate (3%) i.e., there is almost one positive instance in each positive bag.

Parameters: Our method is not sensitive to the choice of parameters across different domains. All the parameters are fixed across all the experiments except thresholding on the final SVM classifier. We have to do this because the literature does not provide precision-recall values.

We set the parameter γ for finding quasi-cliques to 0.9 in all the experiments across both texts and images. We set α in Definition 1 to $exp(10, \lfloor \log \frac{\mathcal{C}(x_i,x_j)}{\mathcal{S}(x_i,x_j)} \rfloor)$ to take into account the differences between the scales of the similarity scores and consistency scores. Our ranking algorithm produces a ranking score for all instances between 0 and 1. Instances with a score higher than 0.9 is considered as positive instance. For training iSVMs, the trade off parameter c is default and we compensate for imbalanced data by weighting negative instances according to the size of the negative set. Damping factor is set to 0.8 in all the experiments as suggested by Google. The final threshold over SVM scores are determined by cross-validation following the experimental setting in [28, 48]. To have an accurate comparison, in each experiments we followed the settings proposed by previous works on each experiment. Results of other methods are reported from the published work with the exception of experiments on IL-MSRC where we use the publicly available codes.

Algorithm	Musk1	Musk2	Elephant	Fox	Tiger
mi-Sim	91.2±1.5	**92.4**±2.1	**89.7**±0.8	**68.1**±2.3	**88.1**±1.5
Instance-level approaches					
mi-SVM	87.4	83.6	82.0	58.2	78.9
GPMIL	89.47	87.25	83.8	65.75	87.37
SMILE	91.3	91.6	85.8	67.7	86.5
MI-CRF	88.0	84.3	85.0	67.5	83.0
MIMN	86	90	89	64	87
MILES	86.3±1.4	87.7±1.4	N/A	N/A	N/A
IL-SMIL	84.2±4.8	83.8±4.2	82.0±2.7	57.1±4.5	80.3±3.3
SVR-SVM	87.9±1.7	85.4±1.8	85.3±2.8	63.0±3.5	79±3.4
Bag-level approaches					
miGraph	88.9±3.3	90.3±2.6	86.8±0.7	61.6±2.8	86.0±1.6
MIGraph	90.0±3.8	90.0±2.7	85.1±2.8	61.2±1.7	81.9±1.5
MI-Kernel	88.0±3.1	89.3±1.5	84.3±1.6	60.3±1.9	84.2±1.6
MI-SVM	77.9	84.3	81.4	59.4	84.0
MissSVM	87.6	80.0	N/A	N/A	N/A
MIForest	85	82	84	64	82
MILboost	82.3	85.7	80.7	55.2	80.7
DD	88.0	84.0	N/A	N/A	N/A
EM-DD	84.8	84.9	78.3	56.1	72.1
PPMM	**95.6**	81.2	82.4	60.3	82.4
APR	92.4	89.2	N/A	N/A	N/A

Table 1. Accuracy of our method, mi-Sim, in predicting bag labels by our approach compared to state-of-the-art bag-level and instance-level MIL approaches on benchmark tasks.

5. Experimental Results

To show the generality of our method, we apply our system to different MIL benchmarks, image, and text domains and show improvement over general MIL methods.

5.1. Bag-Level Predictions

System Performance on Standard Benchmarks: We compare our method, mi-Sim, with previous bag-level and instance-level methods for MIL on benchmark datasets. We report the accuracy of prediction of bag labels in percent via ten times 10-fold cross validation. Our method outperforms all the other methods on all these datasets except PPMM[41] and APR[13] on *Musk1*. PPMM uses an exhaustive search that may be prohibitive in practice. APR has been designed specially for the drug activity prediction.

System Performance on Text Categorization: MIL has shown to be very effective in text categorization. Table 2(a) shows the results of comparisons on text categorization over twenty datasets introduced in miGraph [48]. We report ten times 10-fold cross validation following the experimental setting of miGraph and SVR-SVM [28]. Our method outperforms both methods in most datasets with a large margin. The results can further be improved using more advanced textual features to compute discriminative similarities [23, 19, 22, 20].

System Performance on Image Categorization: Previ-

Table 2(a). Text Categorization

Dataset	MI-Kernel	miGraph	miGraph-web	SVR-SVM	**mi-Sim**
alt.atheism	60.2 3.9	65.5±4.0	82.0±0.8	83.5±1.7	**86.4**±3.1
comp.graphics	47.0±3.3	77.8±1.6	84.3±0.4	85.2±1.5	**88.5**±1.2
comp.windows.misc	51.0±5.2	63.1±1.5	70.1±0.3	66.9±2.6	**72.3**±2.1
comp.ibm.pc.hardware	46.9±3.6	59.5±2.7	79.4±0.8	70.3±2.8	**85.3**±3.2
comp.sys.mac.hardware	44.5±3.2	61.7±4.8	81.0±0	78.0±1.7	**85.1**±1.8
comp.window.x	50.8±4.3	69.8±2.1	79.4±0.5	83.7±2.0	**86.3**±1.7
misc.forsale	51.8±2.5	55.2±2.7	71.0±0	72.3±1.2	**77.6**±2.7
rec.autos	52.9±3.3	72.0±3.7	83.2±0.6	78.1±1.9	**85.4**±1.5
rec.motorcycles	50.6±3.5	64.0±2.8	70.9±2.7	**75.6**±0.9	74.4±1.7
rec.sport.baseball	51.7±2.8	64.7±3.1	75.0±0.6	76.7±1.4	**82.4**±2.3
rec.sport.hockey	51.3±3.4	85.0±2.5	92.0±0	89.3±1.6	**93.0**±1.9
sci.crypt	56.3±3.6	69.6±2.1	70.1±0.8	69.7±2.5	**78.8**±2.0
sci.electronics	50.6±2.0	87.1±1.7	94.0±0	91.5±1.0	**94.6**±1.8
sci.med	50.6±1.9	62.1±3.9	72.1±1.3	74.9±1.9	**82.5**±2.1
sci.space	54.7±2.5	75.7±3.4	79.4±0.8	83.2±2.0	**86.1**±2.6
soc.religion.christian	49.2±3.4	59.0±4.7	75.4±1.2	83.2±2.7	**84.6**±2.4
talk.politics.guns	47.7±3.8	58.5±6.0	72.3±1.0	73.7±2.6	**78.4**±3.1
talk.politics.mideast	55.9±2.8	73.6±2.6	75.5±1.0	80.5±3.2	**85.3**±2.1
talk.politics.misc	51.5±3.7	70.4±3.6	72.9±2.4	72.6±1.4	**76.1**±2.6
talk.religion.misc	55.4±4.3	63.3±3.5	67.5±1.0	71.9±1.9	**80.6**±4.9

Table 2(b). Image Categorization
Dataset: COREL-2000

mi-Sim	**74.2**:[72.7,75.1]
MIGraph	72.1:[71.0,73.2]
miGraph	70.5:[68.7,72.3]
MI-Kernel	72.0:[71.2,72.8]
MI-SVM	54.6:[53.1,56.1]
DD-SVM	67.5:[66.1,68.9]
MissSVM	65.2:[62.0,68.3]
MILES	68.7:[77.3,70.1]

Table 2(c). Dataset: IL-MSRC

mi-Sim	**73.24**
BoW	64.2
miGraph	70.12
MI-SVM	67.32

Table 2(d). Ablation Study

Full System	**88.1**
−Consistent	82.8
−Discriminative	83.7
−Ranking	82.1
Gaussian	83.1

Table 2. Accuracy of our method, mi-Sim, vs. state of the art in predicting bag labels on (a) the text categorization task across different datasets (b) image categorization on the dataset COREL-2000, (c) image categorization on the dataset IL-MSRC, and (d) ablation study on the benchmark Tiger dataset.

ous researchers show an interesting application of MIL in image categorization. Tables 2(b) and (c) show the results of our method versus previous MIL techniques for image categorization on COREL2000 and IL-MSRC. We used the same experimental setting as the previous work [48, 9] , repeat five times 5-random partitioning, and report the overall accuracy of 95% confidence intervals. For our new dataset IL-MSRC, we compared our method with BoW (simple bag-of-words model), MI-SVM, and mi-Graph (features are BOW of SIFT with 1000 codebooks). Our method outperforms all the previous techniques with a large margin. The results can further be improved using more advanced vision-based features to compute discriminative similarities [27].

Contribution of System Components: Table 2(d) shows the accuracy for different controls on benchmark datasets to show the importance of each of the components. (−consistency) removes consistent similarities and only considers pairwise discriminative similarities, (−discriminative) removes discriminative similarities and only considers consistent similarities on edges. (−ranking) replaces random walk ranking with a baseline of using the degree of each node as its final score; this baseline examines the importance of our random walk algorithm to approximate optimization 2. (Gaussian) replaces the discriminative similarity with ϵ-graph of Gaussian similarity in a same way as it is proposed in miGraph (highest competition with our system). Results shows that each component in our model plays an important role in our final system and removing each component drops the performance with a big margin.

System Running Time: Our system, for every instance, trains one iSVM which is a linear classifier and can be learned efficiently. We compare the running time of our full system with miGraph and miSVM on *Tiger* dataset which has 1220 instances and 200 bags. The training time (in second) of our method, miGraph and miSVM are 3.23, 3.12, 856 respectively and testing times are 2.51, 2.71, 2.63. In addition, the training time of our method for each iSVM classifier in MSRC dataset takes less than 0.5 sec.

5.2. Instance-Level Predictions

Table 1 shows that our method outperforms previous instance-level MIL methods in bag-level predictions. To show that our approach is also successful in instance-level predictions, we evaluate our system on *20 Newsgroups* and our new dataset *IL-MSRC* that include instance-level labels.

20 Newsgroups: Figure 3 compares the instance-level accuracy for our method and mi-SVM[1] on all the datasets in the *20 newsgroup* using F1-measure. Positive instances found by our method and mi-SVM are those that have highest score within each bag. We significantly outperform mi-SVM in all those datasets.

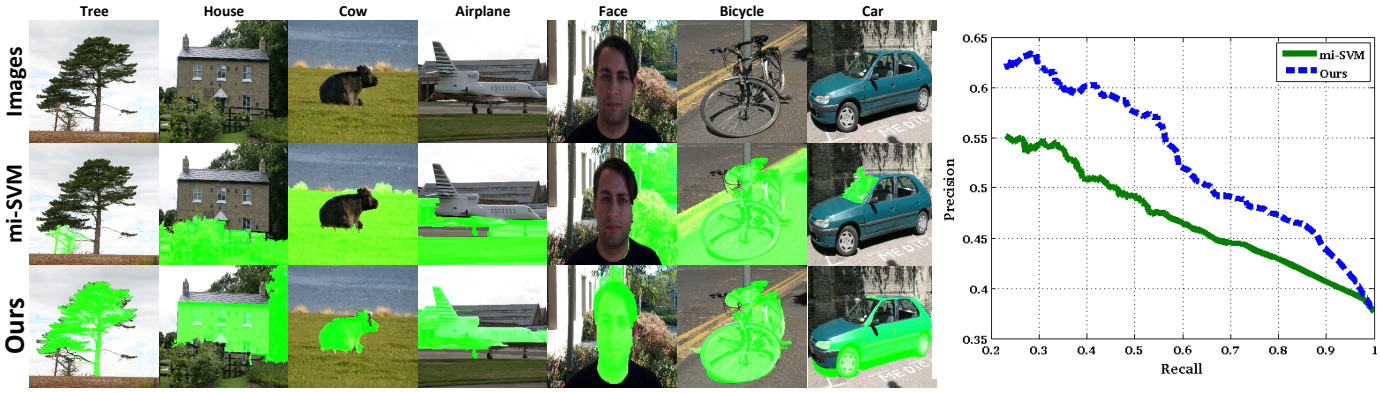

Figure 4. Positive Instance Discovery: Our method is capable of discovering positive instances (superpixels that correspond to the object of interest) in bags(images). (*left*) The first row shows 7 images from IL-MSRC. The second and third rows show discovered superpixels using miSVM and our method, respectively. (*right*) Precision-Recall curve for instance-level predictions in *IL-MSRC*

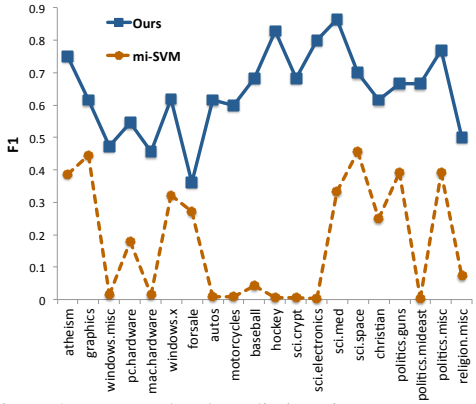

Figure 3. Instance-level predictions in *Newsgroup 20*

Instance-level MSRC: The task is to find which superpixel corresponds to the object of interest. Figure 4(*right*), shows the precision-recall curve on the *IL-MSRC* dataset. The precision-recall curve is traced by a threshold on the instance-level scores. Positive instances are those that are above a threshold (T) within each bag. We do this because there could be several positive instances in each bag. Our method significantly outperforms mi-SVM. Figure 4(*left*) depicts qualitative examples of the regions discovered by our method and mi-SVM. Note that image segmentation is not the focus of this paper. We used this task to showcase the generality of our approach. Our features are very simple and we avoid any vision specific tweaks to be comparable with other MIL methods.

6. Conclusion and Future Work

In this paper we show improvement over the state-of-the-art MIL methods by reasoning about discriminative and consistent similarities. Our method is widely applicable; We show our method produces promising results across text and vision tasks with no change. One potential case of failure of our method is when positive examples are very different from each other and hence they do not follow any particular structure. Under this setting, many other MIL methods would fail as well.

Pairwise similarities are modeled by discriminating each instance from known negative instances (the only certain labels in MIL). This requires training one SVM for every instance. This is computationally less attractive. However, each exemplar SVM is a linear SVM and these classifiers can be learned efficiently in parallel. Since we have a fixed set of negative examples, the computation of exemplar classifiers can be very fast using whitening techniques [21]. Pairwise similarities are not always transitive. This means that reasoning based on pairwise similarities should be observant about the kind of similarities being utilized. In this paper we show that by measuring the consistency of similarities one can select "good" similarities to reason about. This leads to more reliable discovery of positive instances.

Discovering positive instances using only pairwise similarities is a typical example of discovering a global structure by aggregating local evidences. This is a challenging task that appears in a lot of machine learning problems and is susceptible to having issues with local optimum. By augmenting the local evidence with global or structural cues one can aim for better local optimum. This makes local reasoning to be more informed about the global structure. The consistency of similarities, at least in the way we model it, can be thought of as a structural cue that is coupled with local cues, pairwise similarities.

References

[1] Andrews, Tsochantaridis, and Hofmann. SVMs for multiple-instance learning. In *NIPS*, 2003.

[2] S. Andrews and T. Hofmann. Multiple instance learning via disjunctive programming boosting. In *NIPS*, 2004.

[3] Arasu, Cho, Garcia-Molia, Paepcke, and Raghavan. Searching the web. *ACM Trans. on internet technologies*, 2001.

[4] P. Auer. On learning from multi-instance examples. In *ICML*, 1997.

[5] Babenko, Varma, Dollár, and Belongie. Multiple instance learning with manifold bags. In *ICML*, 2011.

[6] A. Blum and A. Kalai. A note on learning from multiple-instance examples. Kluwer Academic Publishers, 1997.

[7] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 1998.

[8] Brunato, Hoos, and Battiti. On effectively finding maximal quasi-cliques in graphs. In *CLIO*, 2007.

[9] Chen, Bi, and Wang. Miles: Multiple-instance learning via embedded instance selection. *TPAMI*, 2006.

[10] Chen and Wang. Image categorization by learning and reasoning with regions. *JMLR'04*, 2007.

[11] Chevaleyre and Zucker. Solving multiple-instance and multiple-part learning problems with decision trees and rule sets. In *AAI*, 2001.

[12] Deselaers and Ferrari. A conditional random field for multiple-instance learning. In *ICML*, 2010.

[13] Dietterich., Lathrop, and Lozano. Solving the multiple instance problem with axis-parallel rectangles. *AI*, 1997.

[14] Felzenszwalb and Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[15] Fu and Robles. An instance selection approach to multiple instance learning. In *CVPR*, 2009.

[16] Gehler and Chapelle. Deterministic annealing for mil. *JMLR'07*, 2007.

[17] H. Hajimirsadeghi, J. Li, G. Mori, M. Zaki, and T. Sayed. Multiple instance learning by discriminative training of markov networks. In *UAI*, pages 262–271, 2013.

[18] H. Hajishirzi, M. Rastegari, A. Farhadi, and J. Hodgins. Semantic understanding of professional soccer commentaries. In *UAI*, 2012.

[19] H. Hajishirzi, W.-t. Yih, and A. Kolcz. Adaptive near-duplicate detection via similarity learning. In *ACM SIGIR*, pages 419–426, 2010.

[20] H. Hajishirzi, L. Zilles, D. S. Weld, and L. S. Zettlemoyer. Joint coreference resolution and named-entity linking with multi-pass sieves. In *EMNLP*, 2013.

[21] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012.

[22] M. J. Hosseini, H. Hajishirzi, O. Etzioni, and N. Kushman. Learning to solve arithmetic word problems with verb categorization. In *EMNLP*, 2014.

[23] W. Hwang, H. Hajishirzi, M. Ostendorf, and W. Wu. Aligning sentences from standard wikipedia to simple wikipedia. In *NAACL*, 2015.

[24] Jai and Zhang. Instance-level semisupervised multiple instance learning. In *AAAI08*, 2008.

[25] M. Kim and F. D. la Torre. Gaussian processes multiple instance learning. In *ICML*, 2010.

[26] R. Koncel-Kedziorski, H. Hajishirzi, and A. Farhadi. Multi-resolution language grounding with weak supervision. In *EMNLP*, pages 386–396, 2014.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105. 2012.

[28] Li and Sminchisescu. Convex multiple-instance learning by estimating likelihood ratio. In *NIPS*, 2010.

[29] Long and Tan. Pac learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. In *COLT*, 1996.

[30] T. Malisiewicz, A. Gupta, and A. Efros. Ensemble of exemplar-svms for object detection. In *ICCV*, 2011.

[31] O. Maron and T. Lozano-Prez. A framework for multiple-instance learning. In *NIPS*, 1996.

[32] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *ICML*, 1998.

[33] R. Mihalcea and D. Radev. *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press, 2011.

[34] Muthukrishnan, Radev, and Mei. Simultaneous similarity learning and feature-weight learning for document clustering. In *TextGraphs*, 2011.

[35] Picardello and Woess. Random walks and discrete potential theory. 2000.

[36] W. Ping, Y. Xu, J. Wang, and X.-S. Hua. FAMER: Making multi-instance learning better and faster. In *SDM*, 2011.

[37] Scott, Zhang, and Brown. On generalized multiple-instance learning. In *IJCAI*, 2003.

[38] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. *Proceedings of ACM SIGGRAPH ASIA*, 2011.

[39] Viola, Platt, and Zhang. Multiple instance boosting for object detection. In *NIPS*, 2006.

[40] Wang, Li, and Zhang. Multiple-instance learning via random walk. In *ECML*, 2006.

[41] Wang, Yang, and Zha. Adaptive p-posterior mixture-model kernels for mil. ICML, 2008.

[42] J. Wang, Zucker, and Jean-Daniel. Solving multiple-instance problem: A lazy learning approach. In *ICML*, 2000.

[43] Xiao, Liu, Cao, Jie, and Wu. Smile: A similarity-based approach for multiple instance learning. In *ICDM*, 2010.

[44] C. Yang. Image database retrieval with multiple-instance learning techniques. In *ICDE*, 2000.

[45] Zhang, Goldman, Yu, and Fritts. Content-based image retrieval using mil. In *ICML*, 2002.

[46] Q. Zhang and S. A. Goldman. EM-DD: An improved multiple-instance learning technique. In *NIPS*, 2001.

[47] Zhou. Multi-instance learning: A survey. In *TR*. AI Lab Department of Computer Science Technology Nanjing University, 2004.

[48] Zhou, Sun, and Li. Multi-instance learning by treating instances as non-i.i.d. samples. In *ICML*, 2009.