MISSL: Multiple-Instance Semi-Supervised Learning

Rouhollah Rahmani RAHMANI@WUSTL.EDU
Sally A. Goldman SG@WUSTL.EDU
Department of Computer Science and Engineering, Washington University, St. Louis, MO, 63130 USA

Abstract

There has been much work on applying multiple-instance (MI) learning to content-based image retrieval (CBIR) where the goal is to rank all images in a known repository using a small labeled data set. Most existing MI learning algorithms are non-transductive in that the images in the repository serve only as test data and are not used in the learning process. We present MISSL (Multiple-Instance Semi-Supervised Learning) that transforms any MI problem into an input for a graph-based single-instance semi-supervised learning method that encodes the MI aspects of the problem simultaneously working at both the bag and point levels. Unlike most prior MI learning algorithms, MISSL makes use of the unlabeled data.

1. Introduction

There has been significant work on applying multiple-instance (MI) learning to content-based image retrieval (CBIR) where the goal is to rank all images in a known repository using a small labeled data set. Most existing MI learning algorithms are non-transductive in that the images in the repository serve only as test data and are not used in the learning process. However, the data repository is available when the classifier is being built, so a transductive semi-supervised MI learning approach is desirable. Since labeled data in CBIR systems typically come from users during an interactive session, it is important to obtain good results using a very small amount of labeled data. Thus semi-supervised learning (SSL), which aims to make use of the large amount of unlabeled data to improve accuracy, is a natural technique to consider.

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

Recently there has been much work on graph-based SSL (Blum & Chawla, 2001; Blum et al., 2004; Joachims, 2003; Zhu et al., 2003; Zhu & Lafferty, 2005; Zhou et al., 2005; Burges & Platt, 2005). For excellent overviews of the area of SSL see Zhu (2005) and Chapelle et al. (2005). Since many graph-based methods are inherently *transductive* in nature, they are a good fit for CBIR.

Most work on SSL has been in the standard single-instance setting, whereas much of the work on CBIR formulates the problem as a MI learning problem. In the MI model each example is represent as a collection (or *bag*) of d-dimensional points where each dimension corresponds to a feature in the representation. The MI model is well suited for CBIR since there is natural ambiguity as to what portion of each image is important to the user, and most of the image may be irrelevant. When applied to CBIR, each bag correspond to an image, and each point in the bag corresponds to a region of the image.

Many researchers have applied supervised MI learning to CBIR (Maron & Lozano-Pérez, 1998; Maron & Ratan, 1998; Zhang et al., 2002; Andrews et al., 2002; Chen & Wang, 2004; Rahmani et al., 2005; Bi et al., 2005). In order to apply a graph-based SSL approach to the MI setting, some way is needed to both consider the individual points and the relationships between the points in each bag. Even for the labeled data, there are no labels associated with the points but rather the labels are only given at the bag level. While ideally unlabeled data would always improve performance, a poor match between the problem structure or data distribution with the model can lead to degradation in performance (Tian et al., 2004).

We present a novel way to transform any MI learning problem into an input for a graph-based single-instance SSL learning method where the graph created encodes the MI aspects of the problem simultaneously working at both the bag and point levels. We call this new approach MISSL (Multiple-Instance Semi-Supervised Learning). MISSL differs from most

prior work by using the unlabeled data.

2. Multiple-Instance Learning

We use the following notation throughout this paper. Let $L = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{|L|}, y_{|L|})\}$ be the labeled data and let $U = \{\mathbf{x}_{|L|+1}, \ldots, \mathbf{x}_n\}$ be the unlabeled data. We let $L = L^+ \cup L^-$ where L^+ is the positive labeled data and L^- is the negative labeled data. We are interesting in settings in which $|L| \ll |U|$.

Let t denote the target function where in a noise-free labeled example $(\mathbf{x}_i, y_i)$, $t(\mathbf{x}_i) = y_i$. In the standard supervised learning model, each example $\mathbf{x}$ is a point in a d-dimensional space. For point $\mathbf{x}_i$ we use x_{ik} to denote its k^{th} feature value for $1 \leq k \leq d$.

In the MI model $\mathbf{x}_i = \{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots \mathbf{x}_{i,|\mathbf{x}_i|}\}$ where each $\mathbf{x}_{ij}$ is a d-dimensional point. The standard terminology is to call $\mathbf{x}_i$ a *bag*. We use $\mathbf{x}_{ij}$ to refer to the j^{th} point in the bag, and $\mathbf{x}_{ijk}$ to refer to its k^{th} feature value. Unlike standard supervised learning where the learner can see the label of each point, in MI learning the learner only receives the labels of the bags.

There has been significant research applying supervised MI learning to CBIR (Maron & Lozano-Pérez, 1998; Maron & Ratan, 1998; Zhang et al., 2002; Andrews et al., 2002; Chen & Wang, 2004; Rahmani et al., 2005; Bi et al., 2005). We only discuss the MI learning algorithms most related to this work. The diverse density (DD) algorithm (Maron & Lozano-Pérez, 1998) uses a two-step gradient descent search with multiple starting points to find a hypothesis that maximizes $DD(h, D)$, which is a measure of the likelihood of hypothesis h given the data D.

The EM-DD algorithm (Zhang & Goldman, 2001) treats the knowledge of which instance corresponds to the label of the bag as a missing attribute and applies a variation of EM (Dempster et al., 1977) to convert a multiple-instance learning problem to a standard supervised learning problem. More recently Rahmani et al. (2005), as part of the Accio! CBIR system, introduced a variant of EM-DD, Ensemble EM-DD, in which the average label from all hypotheses already returned from the multiple runs of EM are used to rank the test images.

Some recent work has applied SSL to CBIR (Wu et al., 2000; Dong & Bhanu, 2003; Tian et al., 2004; Zhou & Li, 2005). The two approaches taken by prior work are to either use EM to treat the label in the unlabeled examples as a hidden variable (Wu et al., 2000; Dong & Bhanu, 2003) or to apply co-training (Zhou & Li, 2005). Both of these techniques train a supervised learning algorithm many times, each time with a different set of labeled data created from $L \cup U$.

3. Graph-Based Semi-Supervised Learning Methods

We use much of the notation and intuition given by Zhu and Lafferty (2005) to briefly introduce the graph-based approach for SSL. These methods define a graph with a vertex corresponding to each example in $L \cup U$, and the (weighted) edges reflect the similarity between nearby examples. We assume that the graph $G = (V, E)$ is represented by an $n \times n$ symmetric weight matrix W. The combinatorial Laplacian is $\Delta = D - W$ where the diagonal matrix $D_{ii} = \sum_j w_{ij}$. A common choice for the weight function is

$$w_{ij} = exp\left(-\sum_{k=1}^{d} \frac{(x_{ik} - x_{jk})^2}{\sigma_k^2}\right)$$

where in our work all parameters are scaled equally, so $\sigma_k = \sigma$ for all k.

The basic idea of graph-based methods is to associate a real-valued label f_i with vertex v_i to capture the probability that $\mathbf{x}_i$ is positive under some specified mixture model θ. A regularization framework is used to ensure that $\mathbf{f}$ is consistent with L and that nearby vertices should have similar labels (smoothness). Let $\mathbf{f} = (f_1, \ldots, f_n)$ where $f_i = f(\mathbf{x}_i)$. Typically for $\mathbf{x}_i \in L$ the function $\mathbf{f}$ is constrained so $f_i = y_i$. Minimizing the energy function $\mathcal{E}(\mathbf{f}) = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$ provides a smoothness constraint on $\mathbf{f}$ since the energy is small when $\mathbf{f}$ varies smoothly over the graph. Finally, for $\mathbf{x}_i \in U$, f_i is its predicted label (or can be treated as a probability that the label is 1). The technique we apply (Zhu et al., 2003) uses the harmonic solution $\Delta f = 0$ subject to the constraints specified by L. Since Δ is the combinatorial Laplacian, this implies that for point $\mathbf{x}_i \in U$, f_i is the average of the values of f as neighboring nodes (called the *harmonic* property).

Blum and Chawla (2001) introduced the graph mincut formulation where the positive examples in L serve as the sources and the negative examples in L serve as the sinks. After finding a mincut, all unlabeled nodes on the source side are labeled as positive and those on the sink side are labeled as negative. The mincut is the *mode* of a Markov random field with binary labels (a Boltzmann machine), and can be viewed as minimizing

$$\infty \sum_{i \in L} (f_i - y_i)^2 + \mathcal{E}(\mathbf{f}) \tag{1}$$

subject to the constraint that all $f_i \in \{0, 1\}$. The first term is the loss function that causes infinite loss if any examples in L are mislabeled (and hence fixes their

values), and the second term is the smoothness constraint. A problem with this approach is that it computes the mode but not the marginal probabilities, so it makes predictions for the unlabeled examples without any measure of confidence. Blum et al. (2004) address this problem using randomization.

Zhu et al. (2003) replace the use of discrete Markov random fields by the continuous Gaussian random fields. Specifically, they relax the constraint on Equation (1) to $f_i \in \Re$ (versus $f_i \in \{0,1\}$). This formulation allows for a simple closed-form solution for the node marginal probabilities.

Many of these methods do not scale well since the cost is cubic in the size of the graph. More recently, Zhu and Lafferty (2005) convert the original graph into a much smaller *backbone graph* by applying a mixture model to $L \cup U$. Since the computational cost is very dependent on the size of the graph, learning on this smaller graph is much more efficient.

4. MISSL

In this section we describe MISSL. Before describing the algorithm in depth, we develop the intuition behind the approach. The most natural approach would be to introduce a vertex for each bag and define a distance metric between bags (e.g. the distance between the two most similar points in the bags). But consider two identical images (in a complex scene) that only differ in that one contains a tiger and the other a zebra. The edge connecting these two bags would have a very high weight even though their labels may differ.

In MI learning, the points in negative bags are always true negatives, whereas the points in a positive bag may be true positives or false positives. The target concept is a set of regions in the domain containing only true positive points. Hence negative bags can only tell us where the target regions are not, but never where they are. While points in negative bags have a known label, the points in the positive bags are the only points that can determine where the target regions exist. Thus, unlike single-instance SSL techniques, where the energy functions represent the propagation of both positive and negative energy, in MISSL only positive energy propagation and the dampening thereof is meaningful.

A very important aspect of MISSL is that it works directly in the MI environment. Many MI learning algorithms (e.g. MI-svm, mi-svm, EM-DD, DD-SVM) use an EM-based approach to select a single point from each bag to represent the bag and create their hypotheses, thereby converting the MI problem into a standard supervised learning problem. MISSL works directly with the MI data which precludes the need for the multiple starts that are necessary in these EM-based algorithms. Also by working directly with the MI data, MISSL has the potential to use all information contained within the bags.

Let $h = (h_1, \ldots, h_d)$ by a hypothesis point. We want to define a *similarity measure* between h and p that can be used as the predicted label given to p by h. We want this predicted label to decay with the distance between d and h according to a Gaussian centered around h with a standard deviation of 1. We define $\ell_h(p) = exp\left(-dist^2(h,p)\right)$ where $dist(h,p)$ is the Euclidean distance between h and p. Observe that $\ell_h(p) \in [0,1]$.

The diverse density of point p given L is defined by $DD(p,L) = \Pr(p \mid L) = \Pr(L \mid p)\Pr(p)/\Pr(L)$ where the second step follows from Bayes' rule. Assuming a uniform prior on the hypothesis space, an iid sample, and removing the constant $\Pr(p)/\Pr(L)$ yields $DD'(p,L) = \Pi_{i=1}^{|L|} \Pr((\mathbf{x}_i, y_i)\,|p)$. We estimate $\Pr((\mathbf{x}_i, y_i) \mid p)$ using $\max_j \{1 - |y_i - \ell_p(x_{ij})|\}$. The term inside the maximum is a measure of the likelihood that the j^{th} point in bag $\mathbf{x}_i$ receives label y_i given that p is a target point. The label for each bag is the likelihood for the best point in the bag.

Due to the existence of false positives, not every positive point should propagate positive energy. MISSL uses L to compute $DD'(p,L)$ for every point $p \in U \cup L^+$, and uses this measure to define the strength of p's ability to propagate its positive energy. Points with high DD exist only in regions of the hypothesis space where each positive bag has a nearby point, and all points from negative bags are far. Returning to our example, if the target is a tiger, the segments forming the tiger have the highest DD. Thus it is beneficial for the edge weight to depend on both the distance and DD measures of the endpoints. One can view this as expressing a prior that nearby points in regions of high diverse density should have similar labels, yet nearby points in regions of low diverse density need not have similar labels. One consequence of the above priors is that since negative bags have low diverse density, the edge weight connecting them to unlabeled bags are near zero. Thus negative bags are effectively disconnected from the graph. To dampen positive energy, MISSL uses a single negative vertex connected to each vertex associated with a bag in U.

As a framework, the positive propagation energy function used by MISSL must satisfy: (1) Only true positive points should propagate positive energy to nearby points belonging to unlabeled or positively labeled

bags. (2) An unlabeled point this is near true positive points should propagate positive energy to other nearby unlabeled points, thereby strengthening the label of both their bags. (3) No negative point can ever propagate energy, since the positive energy of a negative point is always zero.

4.1. Propagation Energy Function

The propagation energy function is a normalized function of the DD measure coupled with a steepness factor γ. We let P be the set of points in bags $U \cup L^+$ (i.e. all unlabeled and positive bags). Let $DD'_{best} = \max_{p \in P} DD'(p, L)$. We normalize DD with respect to the given data set by letting $DD^*(p, L) = DD'(p, L)/DD'_{best}$ for all $p \in P$. Observe that this normalization removes any affect of the multiplicative constant, $\Pr(h)/\Pr(L)$, dropped in going from $DD(p, L)$ to $DD'(p, L)$.

Finally, to polarize the DD values between the high and low energy regions, we raise the normalized DD measure to a power γ. We call γ the "steepness factor" which is a critical parameter that must be chosen with relation to whether the target concept is narrow (e.g. a Coke can) or broad (e.g. a mountain). In narrow concepts, such as images of a specific object, the steepness factor should be large. In broad concepts, such as a natural scenes data set, the steepness should be small. Thus, we define the energy $E(p)$ for point p as $E(p) = (DD'(p, L)/DD'_{best})^{\gamma}$.

4.2. Filtering

Once the positive energy $E(p)$ is computed for each point p, a filtering step is performed to remove points with low energy since they are unlikely to be true positives (near a target region) and thus unnecessarily increase the size of the graph. This filtering step also improves performance since it prevents points that are likely to be false positives from propagating energy. We select a filter threshold F by observing that for a point $p \in L$, $DD'(p, L)$ encodes the average similarity from p to a positive bag (defined using a Gaussian drop off with distance). We set $F \in [0, 1]$ to be the minimum similarity we are willing to tolerate as a candidate for a true positive point. If all points in L^+ have a similarity $< F$ from p, then $DD'(p, L) < F^{|L|}$. Thus, only points with $DD'(p, L) \geq F^{|L|}$ are placed in the point-level graph (and hence the graph given to the SSL algorithm). Given an identical target region and an identical amount of noise in each training bag, the DD values are lower for large L than small L. By using $F^{|L|}$ as a threshold, the filter removes a similar number of points, for the same target, regardless of the size of L.

4.3. Point-Level Graph

In order to define the graph given to the single-instance graph-based SSL algorithm, we first define an abstract point-level graph – it need not be explicitly built. A vertex is placed in the point-level graph for each point $p \in U \cup L^+$ for which $DD'(p, L) \geq F^{|L|}$. For each vertex p in the point-level graph, the propagation energy $E(p)$ is computed.

We now define how the edge weights are computed for the point-level graph. We encode both the propagation energy and the similarity between points into the edge weight. For vertices p_i and p_j, we let

$$w_p(p_i, p_j) = \frac{1}{2}(E(p_i) + E(p_j))e^{-dist^2(p_i,p_j)/\sigma^2} \quad (2)$$

So the edge weights range from 0 to 1. If two points have zero distance and maximum propagation energy, $w_p(p_i, p_j) = 1$. If two points have either infinite distance and/or zero propagation energy, $w_p(p_i, p_j) = 0$. The continuous-valued Gaussian drop off could be replaced by connecting all points within radius d.

The value $1/\sigma^2$ controls the amount by which each point's energy radiates from it. In datasets that have narrow target regions, σ should be small, whereas for datasets with broad target regions, σ should be large. Defining the weights in this manner enables regions with high propagation energy to create local connectivity in the graph. Regions with low propagation energy, that are either not densely positive or contain points from negative bags, locally erode the connectivity of the graph. This connection and erosion encodes into the graph which points are candidates to be true positives. Points that are disconnected (orphans) or are not well connected are either false positives or true negatives.

4.4. Bag-Level Graph

We now use the point-level graph to define the bag-level graph that will be used by the single-instance SSL algorithm. A vertex is placed in the bag-level graph for each positive and unlabeled bag. The edge weight between bags b_1 and b_2 is the sum of all edge weights in the point-level graph between points in b_1 and points in b_2. (No points from negative bags are placed in the point-level graph since their energy values are zero, and thus no vertices are placed in the bag-level graph for negative bags.) So for bags $u, v \in U \cup L^+$,

$$w_B(u, v) = \sum_{p_i \in u, p_j \in v} w_p(p_i, p_j). \quad (3)$$

In the bag-level graph, a *dampening vertex* z that propagates negative energy is connected to each unlabeled vertex via an edge of weight ϵ. Thus any unlabeled bag is only labeled as positive when it has

more positive energy flowing to it than the negative energy coming from z. If an unlabeled vertex has more positive energy than negative, then it can pass on its surplus energy to other unlabeled vertices. The same holds for negative energy. By connecting all unlabeled bags to the dampening vertex, any unlabeled bags which are disconnected will receive a label of 0.

The selection of ϵ is critical because it affects the range for the labels for the bags. Since all edges connecting an unlabeled point to the dampening vertex has weight ϵ, the selection of the edge weight between the dampening vertex and each unlabeled point mathematically has no effect on the relative labels given to the unlabeled bags by the Gaussian Fields SSL algorithm(Zhu et al., 2003). However, selecting a value of ϵ that is too small will drive all bags to have a label very close to 1. Similarly, selecting too large of a value for ϵ will drive the label of all bags to be very close to 0. When either of these extremes are approached, roundoff error will occur, and the ranking will be negatively affected. To ensure that the labels occupy the full range of $[0, 1]$, ϵ could be set to half the weight of the maximum total positive energy between any bag in U and all bags connected to U. Since computationally such an approach would be very expensive, we instead set ϵ to the maximum edge weight between a positive bag and unlabeled bag. That is, we let $\epsilon = \max_{u \in U, v \in L^+} w_B(x_u, x_v)$.

There are several important properties of the bag-level graph. First, for MI applications to CBIR, the target concept is complex, represented by multiple points in the positive bag. An important property of the bag-level graph is that multiple points can contribute to the connectivity strength between the two bags. The number of contributing points is only limited by the filtering step, and thus there is no limit to the number of points from a single bag that can contribute to the value of a bag. Given a positive bag and a unlabeled bag, only connected points from high propagation energy regions will contribute significantly to the total connectivity of the bags. These properties together capture the target concept of a weighted set of regions containing only true positives.

Furthermore, multiple edges between bags in the point-level graph influence the positive energy flow between connected unlabeled bags, which increases the likelihood that they receive a similar label. The more points within two bags that are within a close proximity to each other, the stronger the connectivity will be between the bags since each edge in the point-level graph adds to the weight of the corresponding bag-level edge. Positive energy that flows to one bag will flow strongly to other. This can be a desirable property in some datasets but not in others. In particular, it creates a bias towards identifying objects composed of many segments.

Another property of the bag-level graph is that the number of false positives in a bag does not affect the label of the bag. Strong edges between bags exist from contributing points in high energy regions only. Regardless of the number of points from zero energy regions, there is no affect on the connectivity of the graph. Since our energy function has exponential decay, points from low diverse density regions have exponentially less affect on the graph then those from high diverse density regions. This property is very important in CBIR datasets. A target object in a simple background should be recognized just as well as that same object when placed in a complex background.

4.5. MISSL Algorithm

Figure 1 gives an overview of MISSL. We set $\gamma = 4$, $\sigma = 1/2$, and $F = 0.1$, and also applied a preprocessing step to normalize all features over the dataset to be exactly in the range from 0 to 1. Observe that once the bag-level graph is created, it is given to any single-instance SSL algorithm. In our experiments we use the SSL algorithm of Zhu et al. (2003). The hypothesis returned by the SSL algorithm directly labels each vertex in the bag-level graph. For each bag in U, it receives the label from its corresponding vertex, and these labels are used to rank the bags in U.

Currently, MISSL does not perform any scaling on the training data. All dimensions are considered equally important. We believe that improvements will occur if a successful scaling technique that makes use of the unlabeled data is implemented. Current SSL scaling methods such as that in Zhu et al. (2003) cannot be directly used for our algorithm since they assume that the edge weights are a distance function between two feature vectors in a hypothesis space. Scaling cannot be directly applied to the bag-level graph since it does not represent a feature vector and the edges are a sum of the contributing points. A new scale changes the DD of each point. The resulting point-level graph may have lower energy for some previously high energy point and vise versa. To employ a successful scaling method, a careful study of the effect of how scaling at the point level translates to the bag level is needed.

5. Results

For the CBIR data sets, the goal is to rank versus classify the images. As a measure of performance we have chosen to use the area under the ROC curve (Hanley &

MISSL($L = \{L^+, L^-\}, U, \gamma, \sigma, F$)
Let P be the points in $U \cup L^+$
$P = \{p \in P \mid DD'(p, L) \geq F^{|L|}\}$
Let $DD'_{best} = max_{p \in P} DD'(p, L)$
For each $p \in P$
$E(p) = (DD'(p, L)/DD'_{best})^{\gamma}$
For all $p_1, p_2 \in P$
Compute $w_p(p_1, p_2)$ via Equation (2)
For all $b_1, b_2 \in U \cup L^+$
Compute $w_B(b_1, b_2)$ via Equation (3)
Let $\epsilon = \max_{u \in U, v \in L^+} w_B(x_u, x_v)$
Add dampening vertex z to bag-level graph G
For all $b \in U \cup L^+$
Add vertex in G for bag b
Add edge to z with weight $w_B(b, z) = \epsilon$
Let $W = \{w_B(v_i, v_j)\}$ for all $v_i, v_j \in G$
Run SSL(W) and use its result to label the bags

Figure 1. The MISSL Algorithm.

McNeil, 1982). The ROC curve plots the true positive rate (i.e. the recall) as a function of the false positive rate. The area under the ROC curve (AUC) is equivalent to the probability that a randomly chosen positive image will be ranked higher than a randomly chosen negative image. Unlike the precision-recall curve, the ROC curve is insensitive to ratio of positive to negative examples in the image repository. Regardless of the fraction of the images that are positive, for a random permutation the AUC is 0.5.

MISSL converts an image to a bag using the segmentation with neighbors method introduced in the Accio! CBIR system (Rahmani et al., 2005). First all images are transformed into the YCrCb color space and then pre-processed using a wavelet texture filter so that each pixel in the image has three color and three texture features. Each image is segmented using the IHS segmentation algorithm (Zhang & Fritts, 2005). A bag is created for each image I with a point for each segment $s \in I$. Specifically, each $s \in I$ is represented as a point in a 30-dimensional feature space where the first 6 features hold the average color and texture values for s. The remaining dimensions hold 6 features for each of the cardinal neighbors (N,E,S,W) of s. These features hold the difference between the average color and texture values of s and the neighbor.

We use the SIVAL data set obtained from `www.cse.wustl.edu/~sg/accio`. It includes 25 image categories containing 1500 images. The categories consist of complex objects photographed against 10 different, highly diverse backgrounds. The objects may occur anywhere spatially in the image and typically occupy 10-15% of the image.

We compares the performance of MISSL to the Accio! learning algorithm and the Accio! combined with EM to treat the labels for examples in U as hidden variables (Accio!+EM). One difficulty in creating Accio!+EM is that the labels output by the hypothesis of Accio! are good for ranking, but since the range for the labels varies dramatically for different hypotheses it is difficult to threshold them to label (or weakly label) the bags in U. In our implementation of Accio!+EM we normalize the labels produced by Accio! so that the maximum label is 1.0 and then use a threshold of 0.5.

There is a significant difference in computational costs between Accio!+EM, Accio!, and MISSL. For a single run on the same machine with $|L| = 16$ and $|U| = 1484$, Accio!+EM takes between 10 and 20 minutes, Accio! takes between 10 and 20 seconds, and MISSL runs between 30 and 100 seconds (when distances between all points in the repository are precomputed). Accio!'s runtime is sensitive to $|L|$, whereas MISSL's runtime is relatively indifferent to $|L|$. MISSL's run time performance is mostly sensitive to $|U \cup L^+|$. This difference is because Accio! computes the DD value of potential hypotheses millions of times whereas MISSL only computes the DD value once for each point in $U \cup L^+$. So for very large training set sizes, MISSL runs much faster than Accio!.

All results reported for MISSL and Accio! are averaged over 30 independent runs. Since Accio!+EM is very computationally intensive, only 5 independent runs are used. In all cases we report the 95%-confidence intervals. The performance of EM-DD and other prior approaches for the SIVAL data set can be found in Rahmani et al. (2005). Accio! significantly outperforms EM-DD.

Table 1 shows the average AUC values for all 25 categories of SIVAL. The performance of Accio!+EM is poor for two reasons. First, it is hurt by the fact that Accio! ranks well, but does not classify well. Secondly, as demonstrated by Tian et al. (2004), using EM to make use of the unlabeled data causes performance to degrade when the labeled data is not a representative sample. In the results of Table 1, there are only 8 negative examples selected among the 24 non-desired categories (each with 10 different scenes). Thus, the labeled negative data is not representative of the negative examples in U. In a few cases, the performance is good which explains the large confidence intervals in some cases.

MISSL performs quite well for many of the categories in SIVAL. As compared to EM-DD, MISSL performs statistically better in 16 categories and only performs statistically worse in one category. When compared

Table 1. Average AUC values and 95%-confidence intervals for the SIVAL data set trained on 8 randomly selected positive and 8 randomly selected negative bags.

	MISSL	Accio!	Accio! +EM
FabricSoftenerBox	97.7±0.3	86.6±2.9	44.4±1.1
WD40Can	93.9±0.9	82.0±2.4	50.3±3.0
CokeCan	93.3±0.9	81.5±3.4	48.5±25.6
FeltFlowerRug	90.5±1.1	86.9±1.6	51.1±24.8
AjaxOrange	90.0±2.1	77.0±3.4	43.6±2.4
CheckeredScarf	88.9±0.7	90.8±1.5	58.1±4.4
CandleWithHolder	84.5±0.8	68.8±2.3	57.9±3.0
GoldMedal	83.4±2.7	77.7±2.6	42.1±3.6
SpriteCan	81.2±1.5	71.9±2.4	59.2±22.1
SmileyFaceDoll	80.7±2.0	77.4±3.2	48.0±25.8
GreenTeaBox	80.4±3.5	87.3±2.9	46.8±3.5
DirtyRunningShoe	78.2±1.6	83.7±1.9	75.4±19.8
DataMiningBook	77.3±4.3	74.7±3.3	37.7±4.9
BlueScrunge	76.8±5.2	69.5±3.3	36.3±2.5
DirtyWorkGloves	73.8±3.4	65.3±1.5	57.8±2.9
StripedNotebook	70.2±2.9	70.2±3.1	43.5±3.1
CardboardBox	69.6±2.5	67.9±2.2	57.8±4.7
JuliesPot	68.0±5.2	79.2±2.6	51.2±24.5
TranslucentBowl	63.2±5.2	77.5±2.3	47.4±25.9
Banana	62.4±4.3	65.9±3.2	43.6±3.8
RapBook	61.3±2.8	62.8±1.7	57.6±4.8
WoodRollingPin	51.6±2.6	66.7±1.7	52.5±23.9
GlazedWoodPot	51.5±3.3	72.7±2.2	51.0±2.8
Apple	51.1±4.4	63.4±3.3	43.4±2.7
LargeSpoon	50.2±2.1	57.6±2.3	51.2±2.5

to Accio!, MISSL performs statistically better for 9 categories, and statistically worse for 8 categories. In looking more carefully over the results, the categories in which MISSL does not perform well are simple categories in which the object of interest is generally a single segment. Our belief is that there is a bias in MISSL towards objects composed of many segments since then there are many point-level edges all contributing to a single bag-level edge. We need to perform additional experiments to understand what is occurring so that an adjustment can be made. Another possibility is that for these categories, some features are much more important than others and thus the scaling performed by both Accio! and EM-DD could be important to performance. We believe incorporating scaling in MISSL is one important direction for future work.

In Figure 2, we show learning curves that illustrate how performance changes when both the $|L|$ and $|U|$ are individually varied. In the left two plots, the value of $|L|$ is varied (with all remaining images placed in U). For these plots L is randomly selected so that half of the examples are positive and the other half are negative. MISSL is able to reach its peak performance with fewer labeled examples than needed for the other methods. In the right two plots, the value of $|U|$ is varied while fixing L to 20. In these plots,

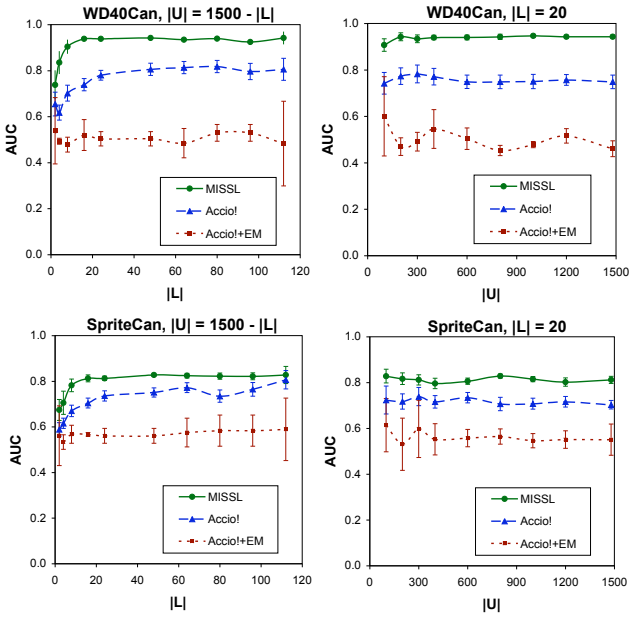

Figure 2. Learning curves (with 95% confidence intervals).

the remaining images are not used. Here we find that MISSL's performance does not change much when U increases. We believe that part of the reason is that MISSL already reaches excellent performance with less than 100 examples (the smallest value for $|U|$) and thus cannot benefit from further examples. We began with $|U| = 100$ since we consider the transductive setting where U is also the test data, and we wanted it sufficiently large to obtain accurate AUC values. In all four curves, MISSL performs better than the other methods and most differences are statistically significant.

6. Conclusion

We have presented MISSL, which we believe is the first multiple-instance SSL algorithm that does not use EM or co-training to make use of the unlabeled data. Thus MISSL does not need to repeatedly run the supervised algorithm with different training data each time. MISSL is very different from all existing MI learning algorithms and has much potential to make use of the large amount of unlabeled data that is typically available for CBIR applications. MISSL has fairly few parameters to tune and is not based on a heuristic search that requires starting from many different points. We have shown that even without learning a scaling for the features, MISSL performs statistically better for many categories of the SIVAL data sets than MI algorithms that learn a scale vector.

Another nice feature of MISSL is that any graph-based SSL algorithm could be use in place of the Zhu et al. (2003) technique. While some of these techniques fit better in our framework than others, there are many

others that could be incorporated. Further testing is needed to determine which one can work best.

There are many interesting directions for future research. Although at this time a method for scaling that uses U does not exist, we can run a supervised MI algorithm which does perform scaling, such as EM-DD, on L before running MISSL. We could then use the scale vectors it learns to scale the DD measure (and distance measure) when creating the point-level graph. We could either scale globally using the best hypothesis generated by EM-DD, or we can scale local regions using the top several hypotheses using the following method. Each scale from a hypothesis of EM-DD has an associated feature vector, representing the area in space for which the scale was generated. Using each of these feature vectors as an epicenter, the hypothesis space can be divided into regions using a Voronoi diagram. Then each point in the dataset can take on the associate scale of the hypothesis feature vector that occurs in its region.

There are also many interesting questions being studied in the area of graph-based SSL techniques that must also be studied in the multiple-instance setting, including studying alternate ways to define the edges within the graph and also to develop techniques that will allow MISSL to scale to much larger unlabeled data sets than is currently possible. We believe that some of the ideas presented by Zhu and Lafferty (2005) could be applied here.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 0329241. We thank Jerry Zhu for his valuable feedback.

References

Andrews, S., Hofmann, T., & Tsochantaridis, I. (2002). Multiple instance learning with generalized support vector machines. *Artificial Intelligence*, 943–944.

Bi, J., Chen, Y., & Wang, J. (2005). A sparse support vector machine approach to region-based image categorization. *IEEE Conf. on Computer Vision and Pattern Recognition*, 1121–1128.

Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. *Proc. 18th Int. Conf. on Machine Learning*, 19–26.

Blum, A., Lafferty, J., Rwebangira, M., & Reddy, R. (2004). Semi-supervised learning using randomized mincuts. *Proc. 21st Int. Conf. on Machine Learning*.

Burges, C., & Platt, J. (2005). Semi-supervised learning with conditional harmonic mixing. In O. Chapelle, B. Scholkopf and A. Z. (Eds.) (Eds.), *Semi-supervised learning*. MIT Press.

Chapelle, O., Scholkopf, B., & Zien, A. (2005). *Semi-supervised learning*. MIT Press.

Chen, Y., & Wang, J. (2004). Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, 913–939.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistics Society*, 1–38.

Dong, A., & Bhanu, B. (2003). A new semi-supervised em algorithm for image retrieval. *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 662–667.

Hanley, J., & McNeil, B. (1982). *The meaning and use of the area under a receiver operating characteristic (ROC) curve*, vol. 143, 29–36.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. *Proc. 20th Int. Conf. on Machine Learning*.

Maron, O., & Lozano-Pérez, T. (1998). A framework for multiple-instance learning. *Neural Information Processing Systems*.

Maron, O., & Ratan, A. (1998). Multiple-instance learning for natural scene classification. *Proc. 15th Int. Conf. on Machine Learning*, 341–349.

Rahmani, R., Goldman, S., Zhang, H., Krettek, J., & Fritts, J. (2005). Localized content-based image retrieval. *MIR*, 227–236.

Tian, Q., Yu, J., Xue, Q., & Sebe, N. (2004). A new analysis of the value of unlabeled data in semi-supervised learning for image retrieval. *Proc. IEEE Int. Conf. on Multimedia Expo*, 1019–1022.

Wu, Y., Tian, Q., & Huang, T. (2000). Discriminant-EM algorithm with application to image retrieval. *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 222–227.

Zhang, H., & Fritts, J. (2005). *Improved hierarchical segmentation* (Technical Report). Washington University in St Louis.

Zhang, Q., & Goldman, S. (2001). EM-DD: an improved multiple-instance learning technique. *Neural Information Processing Systems*.

Zhang, Q., Goldman, S., Yu, W., & Fritts, J. (2002). Content-based image retrieval using multiple instance learning. *Proc. 19th Int. Conf. on Machine Learning*, 682–689.

Zhou, D., Huang, J., & Scholkopf, B. (2005). Learning from labeled and unlabeled data on a directed graph. *Proc. 22nd Int. Conf. on Machine Learning*.

Zhou, Z.-H., & Li, M. (2005). Semi-supervised learning with co-training. *Proc. 19th Int. Joint Conf. on Artificial Intelligence*, 908–913.

Zhu, X. (2005). *Semi-supervised learning literature survey* (Technical Report 1530). Computer Science, University of Wisconsin-Madison. www.cs.wisc.edu/∼jerryzhu/pub/ssl_survey.pdf.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. *Proc. 20th Int. Conf. on Machine Learning*.

Zhu, X., & Lafferty, J. (2005). Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. *Proc. 22nd Int. Conf. on Machine Learning*.