Instance-level Semi-supervised Multiple Instance Learning

Yangqing Jia and **Changshui Zhang**

State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Automation, Tsinghua University, Beijing 100084, China

Abstract

Multiple instance learning (MIL) is a branch of machine learning that attempts to learn information from bags of instances. Many real-world applications such as localized content-based image retrieval and text categorization can be viewed as MIL problems. In this paper, we propose a new graph-based semi-supervised learning approach for multiple instance learning. By defining an instance-level graph on the data, we first propose a new approach to construct an optimization framework for multiple instance semi-supervised learning, and derive an efficient way to overcome the non-convexity of MIL. We empirically show that our method outperforms state-of-the-art MIL algorithms on several real-world data sets.

Introduction

Multiple Instance Learning (MIL) is a branch of machine learning that considers weak supervision information in many applications where unambiguous labels for single instances are difficult to obtain. In MIL, the labels are associated with sets of instances, called *bags*. A bag is labeled positive if it contains at least one positive instance, and negative if all its instances are negative. However, the label of any single instance in the bag is either impossible to obtain or unknown for some reason. The task of MIL is to classify unknown bags (or sometimes single instances) by utilizing the information of the labeled bags.

MIL is first introduced to solve the drug activity prediction problem (Dietterich, Lathrop, and Lozano-Perez 1997), and is further applied to many other applications, among which the most popular two are localized content-based image retrieval (LCBIR) and text categorization (Andrews, Tsochantaridis, and Hofmann 2002; Chen, Bi, and Wang 2006; Chen and Wang 2004; Rahmani and Goldman 2006). In the LCBIR domain, a user provides several images related or not related to the subject s/he is interested in, and the task is to find other related pictures. From a localized view, since an image often contains several objects, what the user really wants is often a certain object in the image. Thus it is appropriate to treat the task from a multiple instance view: if there exists one object that the user is interested in, *i.e.,*

is "positive", the image is classified as positive. Similarly, in the text categorization domain, a document is considered positive if at least one part of the document is related to the subject. This is reasonable especially when the document is long and contains more than one subject.

Different supervised MIL methods have been developed over the past years. Some methods aim to consider the bags as a whole and operates directly on the bags to find their labels, representative algorithms including Diverse Density (DD) (Maron and Lozano-Perez 1998), EM-DD (Zhang and Goldman 2001), DD-SVM (Chen and Wang 2004), and MILES (Chen, Bi, and Wang 2006). Other methods search for the labels of the instances first, and calculate the labels of the bags from the labels of their instances. Many of them have tight relationships with their single instance counterparts, such as the nearest neighbor based Citation-KNN (Wang and Zucker 2000), support vector machine based MI-SVM/mi-SVM (Andrews, Tsochantaridis, and Hofmann 2002) and MissSVM (Zhou and Xu 2007), and so on.

However, in many applications such as drug activity prediction and LCBIR, it may be expensive, time-consuming, or inconvenient to obtain the labels. In such cases, it is worth finding a way to get good results with only a small amount of labeled data. The problem with supervised methods is that they need a comparatively large training set of bags to learn the concept. In the single instance case, semi-supervised learning (SSL) raised the idea of handling the classification task under such circumstance by using unlabeled data. However, currently there has been little research reported on semi-supervised multiple instance learning. This is mainly because in the MIL case, the performance of semi-supervised learning is affected by the ambiguity that the positive instances are not directly given, which results in a non-convex problem and adds difficulty to find the correct target concept.

In this paper, we propose a new semi-supervised multiple instance learning method. Similar to standard graph-based semi-supervised learning in the single instance case, an instance-level graph is constructed to model the similarity between instances. Taking into account the inherent nature that *at least one of the instances* in a positive bag is positive, we introduce new cost criteria and constraints for the semi-supervised learning task. Further, we derive an efficient optimization method to find the solution to the problem. We

---

empirically show that our method outperforms state-of-the-art MIL algorithms on several real-world data sets.

IL-SMIL: Instance-level Semi-supervised Multiple Instance Learning

In this section, after introducing some notations, we formulate the semi-supervised multiple instance learning in an instance-level way and discuss its solution. We will first define the cost criterion based on instance labels, and then derive an efficient sub-optimum solution to the optimization task. The label of each bag is then decided by the label of its corresponding instances.

Notations and Assumption

We denote the data by a set of labeled and unlabeled bags $\{(B_1, y_1^{(B)}), \cdots, (B_{L_B}, y_{L_B}^{(B)}), B_{L_B+1}, \cdots, B_{L_B+U_B})\}$, where the first L_B bags are labeled and the following U_B bags are unlabeled. Each bag B_i is a set of instances, with its label denoted by $y_i^{(B)} \in \{-1, +1\}$, $+1$ for positive and -1 for negative. Without loss of generality, we assume that the first L^+ bags are positive and the following L^- bags are negative ($L^+ + L^- = L_B$). We denote the set of all instances by $X = \{x_1, x_2, \cdots, x_n\}$ where $x_i \in \mathrm{R}^d$ is a d-dimensional feature vector representing an instance, and n is the number of instances. To describe the relationship between bags and instances, we use $x_j \in B_i$ to represent that "x_j is an instance from bag B_i". Without loss of generality, we assume that the first l instances are from labeled bags. The hidden labels of the instances are denoted by $Y = (y_1, y_2, \cdots, y_n)^\top$.

The task of MIL is to learn a soft label function $f : \mathrm{R}^d \rightarrow [-1, 1]$ that learns the label for each instance. We denote the predicted soft label of instance j by $f_j = f(x_j)$. Then, the labels of the bags can be calculated: when the labels take discrete value from $\{-1, 1\}$, a bag's label is 1 if and only if at least one of its instances' labels is 1. For the soft label case, we define a bag B_i's soft label $f_i^{(B)}$ to be determined by the largest value of its instances' soft labels:

$$f_i^{(B)} = \max_{j, x_j \in B_i} f_j. \tag{1}$$

Note again that for either labeled or unlabeled bags, whether a certain instance is positive or negative is unknown to the user. This is essentially what "multiple instance" means.

Instance-level graph

Generally, we assume that all bags are drawn independently from the data distribution, and that the positive instances lie in a certain region in the feature space while the negative ones lie in the remaining space. We also assume that the (soft) label over the feature space is smooth, which is also assumed in most semi-supervised learning methods, and are usually satisfied in most real-time scenarios. Similar to standard SSL, we use an $n \times n$ weight matrix W to model the instance-level graph: W_{ij} is nonzero if and only if instance i is among the k-nearest neighbors of instance j or vice versa. In our paper, we use the Gaussian kernel to calculate the weight as $W_{ij} = \exp\{-\gamma\|x_i - x_j\|^2\}$. Further, we define

the diagonal degree matrix D as $D_{ii} = \sum_j W_{ij}$, and define the graph Laplacian as $L = D - W$ for the discussion below.

The Proposed Method

First, we propose the following cost criterion that is a multiple instance counterpart of the manifold regularization framework(Belkin, Niyogi, and Sindhwani 2005):

$$C_{\mathrm{MI}}(f) = \frac{1}{L_B} \sum_{i=1}^{L_B} V(B_i, y_i^{(B)}, f) + \gamma_A \|f\|_K^2 + \gamma_I \|f\|_I^2. \tag{2}$$

The cost criterion contains one loss function V based on labeled bags, and two regularization terms: $\|f\|_K^2$ minimizes the complexity over a certain Reproducing Kernel Hilbert Space (RKHS), and $\|f\|_I^2$ controls the complexity in the intrinsic geometry of the data distribution. In graph-based SSL, the term $\|f\|_I^2$ is usually approximated by graph Laplacian as

$$\|f\|_I^2 = \frac{1}{n^2} \mathbf{f}^\top L \, \mathbf{f}, \tag{3}$$

where $\mathbf{f} = (f_1, \cdots, f_n)^\top$.

The two regularizers share similar thoughts with single instance learning, because we assume that the soft labels are smooth over the instance-level graph, and the function should have a low complexity in the RKHS. The difficulty with multiple instance learning is that we cannot write the loss function in a convex form such as the squared loss in the single instance case, because the known labels are assigned to bags instead of instances. In the following part, we discuss the loss function for the positive and negative bags separately, starting from the squared loss that is used in most single instance SSL algorithms.

For a negative bag ($i = L^+ + 1, \cdots, L_B$), it is straightforward to see that all instances in the bag are negative, *i.e.*, $y_j = -1$, for all $x_j \in B_i$. Thus we have the penalty term similar to the loss function of single-instance SSL:

$$V(B_i, y_i^{(B)}, f) = \sum_{j:x_j \in B_i} (f_j + 1)^2, \ i = L^+ + 1, \cdots, L_B. \tag{4}$$

Meanwhile, for a positive bag, the case is more complex because positive bags may contain negative instances as well. Actually, only one positive instance is necessary to determine a positive bag. Thus, we define the penalty term for a positive bag to be only related to the instance with the *largest* soft label:

$$V(B_i, y_i^{(B)}, f) = (1 - \max_{j:x_j \in B_i} f_j)^2, \quad i = 1, \cdots, L^+. \tag{5}$$

Note that we do not impose any penalty term on the instances of the bag other than the one with the largest soft label. Some MIL methods (such as mi-SVM and MissSVM) try to determine the labels of all the instances in the positive bag, and attempt to impose a large margin criterion by pushing the soft labels of all the instances to either -1 or 1. However, in many applications such as drug activity prediction and text classification, the change of the instances' labels is considered to be continuous. Then, if positive and negative instances exist simultaneously in a certain bag, there must

be some instances that are close to the decision boundary. Thus, the large margin criterion may not work well in this case, because such labels are inherently ambiguous. Similar arguments have also been proposed in (Zhang and Oles 2000) and (Zhou and Xu 2007).

Thus, we have the following cost criterion for IL-SMIL:

$$
\begin{aligned}
C(f) \;=\; & \gamma_A\|f\|_K^2 + \gamma_I\|f\|_I^2 + \sum_{i=1}^{L^+}(1 - \max_{j:x_j \in B_i} f_j)^2 \\
& + \gamma_L \sum_{i=L^++1}^{L_B} \sum_{j:x_j \in B_i} (f_j + 1)^2.
\end{aligned}
\tag{6}
$$

Because the loss function penalty terms for positive and negative bags have different mathematical forms, a parameter γ_L is used to balance the weight.

Once we have found the optimal labels of the instances by minimizing the cost criterion $C(f)$, the bag-level label of any bag B_i can be calculated by taking the maximum value of its instances' labels using (1). The only problem left is how to solve the optimization task: due to the existence of the $\max(\cdot)$ function in the loss function for positive bags, $C(f)$ is generally non-convex, and cannot be directly optimized. In the following part we will derive a sub-optimum solution to the problem.

Iterative Solution Using CCCP

First, we rewrite the problem in an alternative way by introducing hinge loss parameters:

$$
\begin{aligned}
\underset{f}{\arg\min} \quad & \gamma_A\|f\|_K^2 + \gamma_I\|f\|_I^2 + \sum_{i=1}^{L^+}\xi^2 \\
& + \gamma_L \sum_{i=L^++1}^{L_B} \sum_{j:x_j \in B_i} (f_j + 1)^2
\end{aligned}
\tag{7}
$$

$$
s.t. \quad 1 - \max_{j:x_j \in B_i} f_i \leqslant \xi_i, \quad i = 1, \cdots, L^+ \tag{8}
$$

$$
\xi_i \geqslant 0, \quad i = 1, \cdots, L^+. \tag{9}
$$

This problem is slightly different from the original one. Compare the original cost function (6) and the rewritten cost function (7), if $|f_i| \leqslant 1$ stands for all $i = 1, \cdots, n$, the two function values are strictly equivalent. When there exists some f_i that satisfies $|f_i| > 1$, the criterions (6) and (7) differ in the way of treating these "too correct" values, as the former one imposes penalty while the latter one does not. However, it is worth pointing out that the optimum solutions to the two problems both satisfy $|f_i| \leqslant 1, \forall i = 1, \cdots, n$ (We omit the proof due to space limit here). Thus, rewriting the problem does not affect the optimum solution to the original cost criterion.

Consider the rewritten problem (7)-(9), we can see that the object function is convex, and the L^+ constraints (8) are non-convex but each is the difference between the constant value 1 and a convex $\max(\cdot)$ function (with respect to $\mathbf{f}$). Thus, we adopt the constrained concave convex procedure (CCCP) to find the sub-optimum solution. CCCP is proposed in (Smola, Vishwanathan, and Hofmann 2005) as an

extension of (Yuille and Rangarajan 2003), and is theoretically guaranteed to converge. It works in an iterative way: at each iteration, the 1-st order Taylor expansion is used to approximate the non-convex functions, and the problem is thus approximated by a convex optimization problem. The sub-optimum solution is given by iteratively optimizing the convex subproblem until convergence.

Since the $\max(\cdot)$ function is not differentiable at all points, we employ its subgradient to approximate the 1-st Taylor expansion at the k-th iteration with starting point $\mathbf{f}^{(k)} = (f_1^{(k)}, \cdots, f_n^{(k)})^\top$. For the $\max(\cdot)$ function related to bag B_i, its 1-st order Taylor expansion is approximated as

$$
\left[\max_{j:x_j \in B_i} f_j\right]_{\mathbf{f}^{(k)}} \approx \max_{j:x_j \in B_i} f_j^{(k)} + \Delta_i^\top(\mathbf{f} - \mathbf{f}^{(k)}), \tag{10}
$$

where the subgradient Δ_i is an $n \times 1$ vector whose j-th element is given by

$$
\Delta_{ij} = \begin{cases} \frac{1}{n_i}, & x_j \in B_i \text{ and } f_j^{(k)} = \zeta_i \\ 0, & otherwise \end{cases}. \tag{11}
$$

In the equation above, $\zeta_i = \max_{j:x_j \in B_i} f_j^{(k)}$ is the largest label value in B_i, and n_i is the number of instances that have the largest label value ζ_i. In another word, at every iteration, only the instance (or instances) with the largest label value in B_i is considered in the constraint.

In brief, for the k-th iteration of CCCP, we keep the cost function (7) and constraint (9), and rewrite the non-convex constraint (8) given the previous iteration's output $\mathbf{f}^{(k)}$:

$$
1 - \max_{j:x_j \in B_i} f_j^{(k)} - \Delta_i^\top(\mathbf{f} - \mathbf{f}^{(k)}) \leqslant \xi_i. \tag{12}
$$

This is a linear constraint. The subproblem formed by (7), (12) and (9) is thus convex, whose optimum solution can be calculated within quadratic time complexity.

Kernel Based Representation

In the multiple instance case, the Representer Theorem for manifold regularization (Belkin, Niyogi, and Sindhwani 2005) still holds, *i.e.,* the minimizer of the optimization problem admits an expansion

$$
f^*(x) = \sum_{i=1}^n \alpha_i K(x_i, x), \tag{13}
$$

where $K(\cdot, \cdot)$ is the reproducing kernel of the RKHS. Then, we are able to solve the problem using the n-dimensional expansion coefficient vector $\alpha = (\alpha_1, \cdots, \alpha_n)^\top$. In detail, the subproblem for the k-th iteration of CCCP with respect to α is:

$$
\begin{aligned}
\underset{\alpha}{\arg\min} \quad & \gamma_A \alpha^\top K \alpha + \frac{\gamma_I}{n^2} \alpha^\top K^\top L K \alpha + \sum_{i=1}^{L^+} \xi^2 \\
& + \gamma_L \sum_{i=L^++1}^{L_B} \sum_{j:x_j \in B_i} (K_{j.}\alpha + 1)^2
\end{aligned}
\tag{14}
$$

$$
s.t. \quad 1 - \max_{su B_i} K_{j.}\alpha^{(k)} - \Delta_i^\top(K\alpha - K\alpha^{(k)}) \leqslant \xi_i
$$

$$
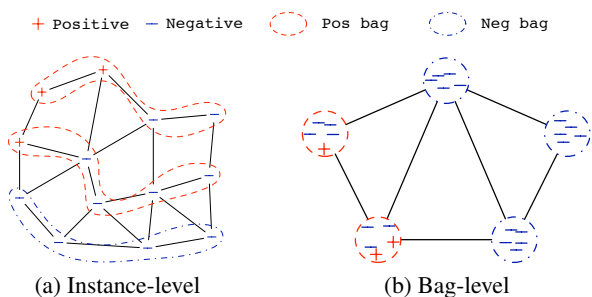\xi_i \geqslant 0, \quad i = 1, \cdots, L^+,
$$

Figure 1: The instance-level and the bag-level graph. The +/- symbols are instances, the black lines are edges in the graph structure, and the dashed circles indicate positive and negative bags.

where with a slight abuse of notation, we use K as the Gram matrix over all instances, and $K_{j.}$ as the j-th row of K. This subproblem is a standard quadratic programming (QP) problem and can be solved by any state-of-the-art QP solvers. Running CCCP iteratively until convergence, we can obtain the sub-optimum solution for the instance labels. The label for each bag is then calculated as the largest label of all its instances using (1).

It is worth pointing out that manifold regularization is a general form of several semi-supervised learning methods such as the Gaussian random fields and harmonic function method (Zhu, Ghahramani, and Lafferty 2003), the local and global consistency method (Zhou et al. 2003), etc. Thus these methods can be easily extended to the multiple instance case using our derivation. Also, our IL-SMIL can be naturally extended to induction similar to (Belkin, Niyogi, and Sindhwani 2005).

Instance-level vs. Bag-level

For the combination of MIL and SSL, (Rahmani and Goldman 2006) has raised the multiple instance SSL problem for CBIR, and proposed a graph-based algorithm called MISSL. The main thought of MISSL is to convert each bag into a single node in the graph. To perform this, it first calculates the similarity between instances via a heat kernel, and uses Diverse Density criterion to calculate the "energy" of each instance, which can be interpreted as the probability that the instance is positive. Then, a *bag-level* graph is constructed, whose nodes are bags and edges are pairwise similarities between bags. A standard SSL algorithm is implemented to get the labels of the bags. An example of the instance-level and bag-level graph is shown in Figure 1.

The aim of the bag-level graph is to fit the MIL problem into the classical SSL framework, and to circumvent the non-convexity problem due to the weak supervision information. This is done by defining a way to represent relationship between bags. However, a bag-level graph may suffer from several shortcomings. First, the similarity between two bags is calculated by a weighted sum of the pairwise similarity between its instances, which is somehow heuristic and does not have a sound theoretical support. Actually, its

performance is greatly affected if the number of instances (and positive instances) per bag varies greatly among different bags, which is very common in real-world applications. Also, the accuracy of the weight (or "energy" as called in (Rahmani and Goldman 2006)) plays an important role, but due to the Diverse Density nature of the calculation, it might not be easy to get accurate energy values with a small amount of training bags, which again results in the ambiguity of the bag-level graph structure. Moreover, the bag-level graph sacrifices the information of instance-level relationships and only stores relationship between bags. This may affect the final result of the method. It is also difficult to discuss how the bags distribute in the abstract "bag-level" space, or whether the distribution satisfies semi-supervised learning's assumptions.

On the contrary, an instance-level graph has several advantages over the bag-level graph. First, it is a straightforward representation of the graph-based learning, and does not need to translate the instance-level similarity measure to the bag-level, which is often heuristic and may bring unnecessary loss of information. Also, the structure of the instance-level relationship has a solid theoretical foundation from the heat kernel and the spectral graph theory (Chung 1997). Actually, one of the major reasons for constructing a bag-level graph is to circumvent the non-convexity problem. We have shown that in our method, the non-convexity of the multiple instance learning has been efficiently solved by our framework together with CCCP, which works well according to the experimental results.

Experiments and Discussion

We evaluate the performance of our method and compare it with the MISSL method on three real-world data sets, namely drug activity prediction, LCBIR, and text categorization[1]. For drug activity prediction, results from two representative supervised methods, namely EMDD and multiple instance SVM, are also reported to test the efficiency of SSL. In all our experiments, the performance is evaluated by the area under the Receiver Operating Characteristic (ROC) curve, abbreviated as AUC, which shows the tradeoff between sensitivity and specificity.

All results reported here are averaged over 20 independent runs. The features are normalized to $[0, 1]$. For our method, the graph is constructed with parameters $k = 15$, $\gamma = 0.05$ for drug activity prediction and LCBIR, and $k = 30$, $\gamma = 0.5$ for text categorization due to large data numbers and sparse features. In the experiments, we fix the parameter γ_L as $\gamma_L = \frac{\#PB}{\#NI}$, where $\#NI$ is the number of labeled negative instances and $\#PB$ is the number of labeled positive bags. Parameters γ_A and γ_I are determined by cross validation and a grid search from 10^{-3} to 10^4. We simply use a linear kernel for the RKHS norm. For the parameters of the MISSL method, we fix $F = 0.1$ and use a grid search for γ (taking integer value from 1 to 4), σ (from 0.01 to 1), and the other parameters for later bag-level SSL. The best result is reported here.

---

[1]The LCBIR and text categorization data sets can be found at http://www.cs.columbia.edu/~andrews/mil/datasets.html.

Method		Musk1	Musk2
EMDD		73.22±6.01	75.10±5.94
MI-SVM	Poly	64.51±14.63	74.37±8.85
	linear	65.43±11.74	71.09±10.89
	RBF	70.79±10.23	80.07±8.24
mi-SVM	Poly	69.61±7.28	71.09±10.53
	linear	68.37±7.32	67.76±11.20
	RBF	73.52±7.75	77.70±8.96
MISSL		76.61±2.71	71.58±2.78
IL-SMIL		**84.17±4.83**	**83.79±4.23**

Table 1: The area under the ROC curve (AUC) in percentage on Musk.

Dataset	MISSL	IL-SMIL
Elephant	78.75±2.86	**82.06±2.70**
Tiger	74.71±3.90	**80.31±3.32**
Fox	**61.12±3.53**	57.05±4.59
TST1	63.38±1.99	**84.72±3.02**
TST2	51.63±1.43	**62.60±1.93**
TST3	54.02±1.90	**60.98±2.92**
TST4	52.49±2.08	**65.59±3.59**
TST7	54.05±1.45	**60.50±2.99**
TST9	59.43±2.26	**64.70±3.62**
TST10	62.22±2.25	**67.59±3.32**

Table 2: Results of LCBIR and Text Categorization.

Drug Activity Prediction

The Musk data set is available from the UCI Repository (Asuncion and Newman 2007). It contains two subsets: Musk1 has 47 positive bags and 45 negative bags with an average of about 5.2 instances per bag; Musk2 has 39 positive bags and 63 negative bags, and the number of instances in each bag is much larger, and differs more significantly than Musk1, ranging from 1 to 1,044 (about 64.5 instances per bag in average). For both sets, each instance is described by a 166-dim vector. We use 5 labeled positive bags and 5 labeled negative bags for Musk1, and 10 respectively for Musk2. The results are presented in Table 1. For the supervised learning methods, their parameters are tuned in a grid search way similar to MISSL.

It can be seen that our method outperforms all other methods on both data sets, especially on Musk1. Also, for Musk1, the two semi-supervised method are superior to all the supervised methods. For Musk2, although MISSL performs poorer (possibly because the number of instances in a bag is much inconsistent between bags), our method is still performing better than the other algorithms. This indicates that for multiple instance problems, employing information about the unlabeled bags is useful for better classification. Moreover, the experiments show that the AUC values of all supervised methods have a large deviant, indicating that the performance is unstable and relies much on the selection of the labeled data. On contrary, the two semi-supervised learning methods are more stable, which is another advantage of utilizing unlabeled bags.

Localized Content-based Image Retrieval

We use the Elephant, Tiger and Fox data sets from (Andrews, Tsochantaridis, and Hofmann 2002) for experiments on the LCBIR application. Each of the three data sets contains 100 positive bags and 100 negative bags of about 1300 instances. Each bag represents an image, and instances in a bag represent different segmented blobs of the corresponding image. We use 10 labeled positive bags and 10 labeled negative bags for all three data sets. The result is shown in Table 2. It can be seen that our method performs better than MISSL on Elephant and Tiger, while on Fox MISSL performs slightly better than ours. As MISSL is initially proposed to solve the LCBIR problem, we infer that our method performs competitively in the LCBIR tasks.

Text Categorization

The text categorization data set contains seven sets, each containing 200 positive and 200 negative bags with about 3330 instances. We report the result of our method and MISSL in Table 2. 30 positive and 30 negative bags are labeled to perform semi-supervised learning. For simplicity, the parameters are tuned solely on TST1, and the best parameter is then applied for all other sets. Our method outperforms MISSL on all 7 data sets, with an increase of about eight percent on average.

Further, we test the performance under different numbers of both labeled and unlabeled bags. The results on TST1-3 (the others are omitted for simplicity) are shown in Figure 2. For the first row, we vary the number of labeled bags from 5 to 50 for both positive and negative bags, and use the other bags as unlabeled data; for the second row, we fix the number of labeled bags to 30 positive and 30 negative, and vary the number of unlabeled bags from 40 to 340. The learning curves showed the superiority of our method over MISSL in most of the cases. Also, a larger number of bags (both labeled and unlabeled) bring up two results: (1) they help to increase the AUC, and (2) they make the result of the algorithm more stable (*i.e.,* a smaller standard deviant). Note again that adding unlabeled bags improves the performance, which indicates that unlabeled data is useful for better classification in multiple instance learning.

Conclusion

In this paper, we proposed a novel algorithm for semi-supervised multiple instance learning from an instance-level view. We used the instance-level graph to describe the relationship between instances, and designed IL-SMIL similar to manifold regularization for semi-supervised multiple instance learning. We derived the loss function for positive and negative bags respectively, and employed an efficient CCCP algorithm to perform optimization. The spirit of our method roots from the standard semi-supervised learning theory. It is firmly supported both empirically and theoretically by previous literature, and can naturally be extended to induction. The experiments over different real-world applications have shown the superiority of our method.
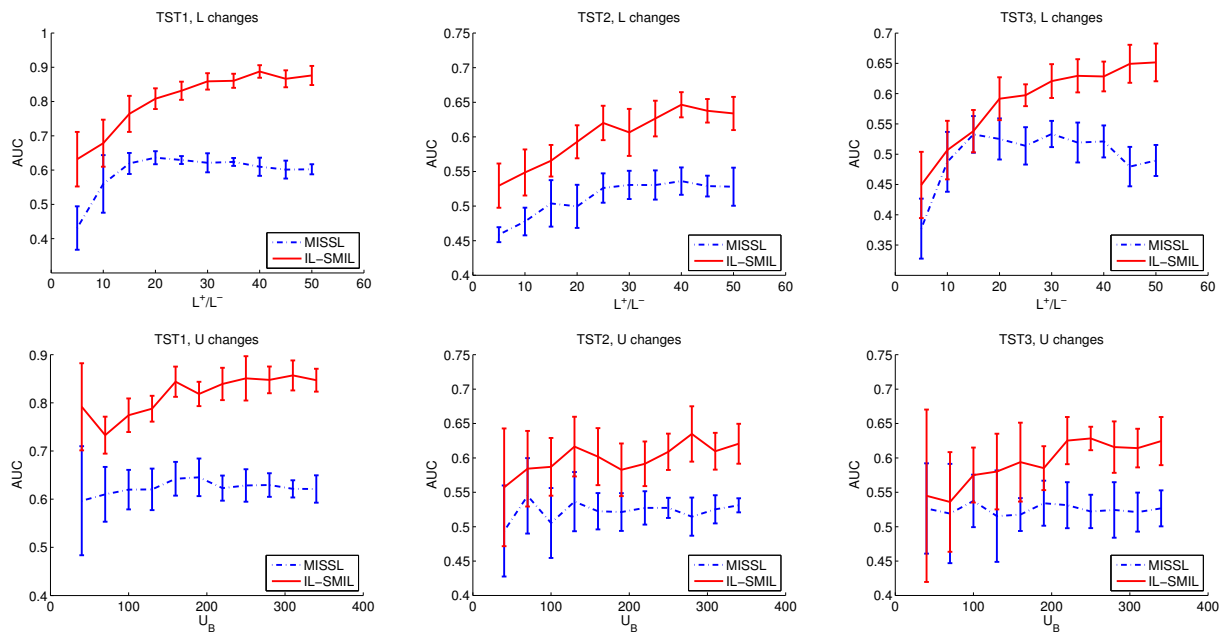
Figure 2: Learning curves on TST1-3 (with standard deviation shown as error bars).

Another potential advantage of our method is that it gives detailed information about the labeling of instances. On a bag-level graph this is not direct, as the prediction is only made on the bag level. This advantage enables us to look into the instance-level structure of the problem. We will focus on utilizing such information in our future work.

Acknowledgement

References

Andrews, S.; Tsochantaridis, I.; and Hofmann, T. 2002. Support vector machines for multiple-instance learning. *Advances in Neural Information Processing Systems*.

Asuncion, A., and Newman, D. 2007. UCI machine learning repository. *http://www.ics.uci.edu/˜mlearn/ MLRepository.html*.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2005. On manifold regularization. *Proc. International Workshop on Artificial Intelligence and Statistics*.

Chen, Y., and Wang, J. 2004. Image Categorization by Learning and Reasoning with Regions. *The Journal of Machine Learning Research* 5:913–939.

Chen, Y.; Bi, J.; and Wang, J. 2006. MILES: Multiple-Instance Learning via Embedded Instance Selection. *IEEE Trans Pattern Anal Mach Intell* 28(12):1931–1947.

Chung, F. 1997. *Spectral Graph Theory*. American Mathematical Society.

Dietterich, T.; Lathrop, R.; and Lozano-Perez, T. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence* 89(1):31–71.

Maron, O., and Lozano-Perez, T. 1998. A framework for multiple-instance learning. *Advances in Neural Information Processing Systems* 10:570–576.

Rahmani, R., and Goldman, S. 2006. MISSL: Multiple instance semi-supervised learning. *Proc. International Conference on Machine Learning* 705–712.

Smola, A.; Vishwanathan, S.; and Hofmann, T. 2005. Kernel methods for missing variables. *Proc. International Workshop on Artificial Intelligence and Statistics*.

Wang, J., and Zucker, J. 2000. Solving the multiple-instance problem: A lazy learning approach. *Proc. International Conference on Machine Learning* 1119–1125.

Yuille, A., and Rangarajan, A. 2003. The Concave-Convex Procedure. *Neural Computation* 15(4):915–936.

Zhang, Q., and Goldman, S. 2001. EM-DD: An improved multiple-instance learning technique. *Advances in Neural Information Processing Systems* 14:1073–1080.

Zhang, T., and Oles, F. J. 2000. A probability anaylsis on the value of unlabeled data for classification problems. *Machine Learning* 17:1191–1198.

Zhou, Z., and Xu, J. 2007. On the relation between multi-instance learning and semi-supervised learning. *Proc. International Conference on Machine Learning*.

Zhou, D.; Bousquet, O.; Lai, T.; Weston, J.; and Scholkopf, B. 2003. Learning with Local and Global Consistency. *Advances in Neural Information Processing Systems*.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. 2003. Semi-supervised learning using Gaussian fields and harmonic functions. *Proc. International Conference on Machine Learning*.