On the Relation Between Multi-Instance Learning and Semi-Supervised Learning

Zhi-Hua Zhou ZHOUZH@LAMDA.NJU.EDU.CN
Jun-Ming Xu XUJM@LAMDA.NJU.EDU.CN
National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

Abstract

Multi-instance learning and semi-supervised learning are different branches of machine learning. The former attempts to learn from a training set consists of labeled *bags* each containing many unlabeled instances; the latter tries to exploit abundant unlabeled instances when learning with a small number of labeled examples. In this paper, we establish a bridge between these two branches by showing that multi-instance learning can be viewed as a special case of semi-supervised learning. Based on this recognition, we propose the MissSVM algorithm which addresses multi-instance learning using a special semi-supervised support vector machine. Experiments show that solving multi-instance problems from the view of semi-supervised learning is feasible, and the MissSVM algorithm is competitive with state-of-the-art multi-instance learning algorithms.

1. Introduction

Multi-instance learning (Dietterich et al., 1997) and semi-supervised learning (Chapelle et al., 2006; Zhu, 2006) are different branches of machine learning. In multi-instance learning, the training set is composed of labeled *bags* each consists of many unlabeled instances, and the goal is to learn some concept from the training set for correctly labeling unseen bags. Here a bag is positively labeled if it contains at least one positive instance and negatively labeled otherwise. In semi-supervised learning, there are a small number of labeled training examples and abundant unlabeled instances, and the goal is to exploit these unlabeled instances to help improve the performance of supervised learning. Both of these two branches have attracted much attention during the past few years.

In this paper, we establish a bridge between multi-instance learning and semi-supervised learning. We show that multi-instance learning can be viewed as a special case of semi-supervised learning, that is, learning with labeled negative examples along with unlabeled instances enforced with *positive constraints*. The intuition is that although the instances' labels are not given in multi-instance learning, it is known that a negative bag does not contain any positive instance. Thus, we can regard all the instances in negative bags as labeled negative instances. On the other hand, since a positive bag may contain positive as well as negative instances, we can regard its instances as unlabeled ones enforced with a positive constraint that at least one of them is positive. Based on this recognition, we propose the MissSVM (Multi-Instance learning by Semi-Supervised Support Vector Machine) algorithm, which tackles multi-instance problems using semi-supervised learning techniques, in particular, a special semi-supervised SVM. Experiments show that the MissSVM algorithm is competitive with state-of-the-art multi-instance learning algorithms.

The rest of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 proposes the MissSVM algorithm. Section 4 presents the experimental results. Finally, Section 5 concludes.

2. Related Work

2.1. Multi-Instance Learning

Multi-instance learning was originated from Dietterich et al. (1997)'s research on drug activity prediction, since that it has been studied by a lot of researchers and many algorithms have been developed, such as Diverse Density (Maron & Lozano-Pérez, 1998) and EM-DD (Zhang & Goldman, 2002), the k-nearest neighbor

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

algorithm Citation-kNN (Wang & Zucker, 2000), decision tree algorithms RELIC (Ruffo, 2000) and ID3-MI (Chevaleyre & Zucker, 2001), rule learning algorithm RIPPER-MI (Chevaleyre & Zucker, 2001), SVM algorithms MI-SVM and mi-SVM (Andrews et al., 2003) and DD-SVM (Chen & Wang, 2004), ensemble algorithms MI-Ensemble (Zhou & Zhang, 2003) and MI-Boosting (Xu & Frank, 2004), logistic regression algorithm MI-LR (Ray & Craven, 2005), etc. Many of those algorithms were developed by adapting a single-instance supervised learning algorithm to multi-instance learning through shifting its focus from the discrimination on the instances to the discrimination on the bags (Zhou & Zhang, 2003).

Besides multi-instance classification, multi-instance regression has also been studied (Amar et al., 2001; Ray & Page, 2001). Weidmann et al. (2003) formulated *generalized multi-instance learning* through employing different assumptions of how the instances' classifications determine their bag's label. Another generalized multi-instance learning setting was defined by Scott et al. (2003). In this paper we focus on standard multi-instance learning (Dietterich et al., 1997).

Multi-instance learning techniques have already been applied to diverse applications such as image categorization (Maron & Ratan, 1998; Chen & Wang, 2004; Chen et al., 2006), computer security (Ruffo, 2000), Web mining (Zhou et al., 2005b), face detection (Viola et al., 2006), etc.

2.2. Semi-Supervised Learning

Semi-supervised learning (Chapelle et al., 2006; Zhu, 2006) deals with algorithms for exploiting unlabeled data to improve supervised learning performance. Many semi-supervised learning approaches have been developed. Some approaches use a generative model for the classifier and employ EM to model the label estimation or parameter estimation process (Miller & Uyar, 1997; Nigam et al., 2000); some approaches use the unlabeled data to regularize the learning process in various ways, e.g., defining a graph on the data set and then enforcing the label smoothness over the graph as a regularization term (Belkin et al., 2001; Zhou et al., 2005a; Zhu et al., 2003); some approaches train two learners and then let the learners to label unlabeled instances for each other (Blum & Mitchell, 1998; Goldman & Zhou, 2000; Zhou & Li, 2005).

Semi-supervised support vector machines have been studied by many researchers, which attempt to maximize the margin on both labeled and unlabeled data, by assigning unlabeled data to appropriate classes such that the resulting margin is the maximum. Earlier works include TSVM (Joachims, 1999), S^{3}VM (Bennett & Demiriz, 1999), V^{3}SVM and CV^{3}SVM (Fung & Mangasarian, 1999), etc. Multi-class semi-supervised SVM has also been developed (Xu & Schuurmans, 2005). Recently, Chapelle et al. (2007) showed that the popular semi-supervised SVM objective function is very well suited for semi-supervised learning, and indicated that more effort should be made on trying to efficiently find good local minima. Actually, much effort has already been devoted to this direction. In particular, Collobert et al. (2006) indicated that in many cases non-convexity could provide scalability advantages over convexity, and by exploiting CCCP (Smola et al., 2005), they reported a significant speed up of semi-supervised support vector machines.

In this paper we develop a special kind of semi-supervised support vector machine to tackle multi-instance problems. Note that this is different from *multi-instance semi-supervised learning* (Rahmani & Goldman, 2006) whose goal is to exploit abundant unlabeled bags to help improve the performance of learning with a small number of labeled bags.

3. The Proposed Method

3.1. Notations

Let $\mathcal{X}$ denote the instance space. Given a data set $\{(X_1, y_1), (X_2, y_2), \cdots, (X_m, y_m)\}$, where $X_i = \{\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \cdots, \boldsymbol{x}_{i,n_i}\} \subseteq \mathcal{X}$ ($i \in \{1, \cdots, m\}$) is a set of instances called a *bag* and $y_i \in \{-1, +1\}$ is a class label, the goal is to predict the label for an unseen bag. Here $\boldsymbol{x}_{ij} \in \mathcal{X}$ ($j \in \{1, \cdots, n_i\}$) is an instance $[x_{ij1}, x_{ij2}, \cdots, x_{ijd}]'$ where x_{ijk} is the value of $\boldsymbol{x}_{ij}$ at the kth ($k \in \{1, \cdots, d\}$) attribute, and n_i denotes the number of instances in X_i. X_i is a *positive bag* (thus $y_i = +1$) if there exists $g \in \{1, \cdots, n_i\}$, $\boldsymbol{x}_{ig}$ is positive. Yet the concrete value of the index g is not known.

Without loss of generality, assume that there are p positive bags and q negative bags, $0 < p, q < m$ and $p + q = m$, and the negative bags are ordered before the positive bags. That is, the training set is organized as $\{X_1^-, X_2^-, \cdots, X_q^-, X_{q+1}^+, \cdots, X_{q+p-1}^+, X_m^+\}$, where X_i^- and X_i^+ denote that X_i is a negative or positive bag, respectively. If we put the instances bag-by-bag into an instance set $\{\boldsymbol{x}_{11}, \cdots, \boldsymbol{x}_{1,n_1}, \boldsymbol{x}_{21}, \cdots, \boldsymbol{x}_{m1}, \cdots, \boldsymbol{x}_{m,n_m}\}$ and re-index the instances into $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T\}$ where $T = \sum_{i=1}^{m} n_i$, then it is evident that the first $T_L = \sum_{i=1}^{q} n_i$ instances are from negative bags while the remaining $T_U = \sum_{i=q+1}^{m} n_i$ instances are from positive bags, and the bag X_i's instances are $\{\boldsymbol{x}_{s_i}, \cdots, \boldsymbol{x}_{e_i}\}$ where $s_i = \sum_{l=1}^{i-1} n_l + 1$ and $e_i = \sum_{l=1}^{i} n_l = s_i + n_i - 1$.

3.2. A Reformulation of the Task

Now we show that how multi-instance learning can be viewed as a special semi-supervised learning task.

According to the definition of multi-instance learning (Dietterich et al., 1997), a positive bag contains at least one positive instance while a negative bag does not contain any positive instance. This implies that all the instances in negative bags are negative. As for every positive bag, we can regard it as a subset of unlabeled instances with a *positive constraint*, i.e. at least one instance in this subset is positive. Then, using the notations defined in Section 3.1, the multi-instance learning task can be reformulated as:

Definition 1 *Given a set of labeled negative examples* $\{(\boldsymbol{x}_1, -1), (\boldsymbol{x}_2, -1), \cdots, (\boldsymbol{x}_{T_L}, -1)\}$ *and a set of unlabeled instances* $\{\boldsymbol{x}_{T_L+1}, \cdots, \boldsymbol{x}_T\}$*, to learn a function* $F^s : \mathcal{X} \to \{-1, +1\}$ *subject to: For* $i = q+1, \cdots, m$*, at least one instance in* $\{\boldsymbol{x}_{s_i}, \cdots, \boldsymbol{x}_{e_i}\}$ *is positive.*

The task stated in Definition 1 is obvious a semi-supervised learning task. The function F^s is different from the desired multi-instance learning function F, but after obtaining F^s it is easy to derive the value of F for any unseen bag $X^* = \{\boldsymbol{x}_{*1}, \boldsymbol{x}_{*2}, \cdots, \boldsymbol{x}_{*,n_*}\}$ through the following rule: $F(X^*) = +1$ if there exists a $j \in \{1, 2, \cdots, n_*\}$ such that $F^s(\boldsymbol{x}_{*j}) = +1$ and $F(X^*) = -1$ otherwise. Thus, solving the original multi-instance learning task is equivalent to solving the semi-supervised learning task in Definition 1.

3.3. MissSVM

Let $\mathcal{H}$ be a Reproducing Kernel Hilbert Space (RKHS) of function $f : \mathcal{X} \to \mathbb{R}$. Denote the RKHS norm of $\mathcal{H}$ by $||f||_{\mathcal{H}}$. The optimization problem for popular semi-supervised support vector machine is:

$$\min_f \frac{1}{2}||f||_{\mathcal{H}}^2 + \lambda \sum_{t=1}^{T_L} H_1(y_t f(\boldsymbol{x}_t)) + \delta \sum_{t=T_L+1}^{T} D(f(\boldsymbol{x}_t)) \tag{1}$$

where $H_1(z) = \max\{0, 1-z\}$ is Hinge Loss. The loss function $D(z)$ is usually a non-convex hat shape function which can be defined and solved in different ways (Zhu, 2006). Here we adopt Bennett and Demiriz (1999)'s definition:

$$D(z) = \min\{H_1(z), H_1(-z)\} \tag{2}$$

For the task in Definition 1, if we do not consider the positive constraints, a function f^s will be generated after the training process of the above semi-supervised support vector machine, which maximizes the margin on both labeled and unlabeled data. Then, a bag X will be labeled as:

$$\text{sign}\left(\max_{\boldsymbol{x} \in X} f^s(\boldsymbol{x})\right). \tag{3}$$

Considering the positive constraints, the following term should be added to the optimization function:

$$\sum_{i=q+1}^{m} H_1\left(\max_{t=s_i,\cdots,e_i} f(\boldsymbol{x}_t)\right) \tag{4}$$

Thus, based on the above considerations, we can write the optimization function as the constrained minimization problem shown in Eq. 5.

$$\min_{f \in \mathcal{H}, \eta, \theta, \varepsilon, \xi} \frac{1}{2}||f||_{\mathcal{H}}^2 + \lambda \eta' \mathbf{1} + \gamma \theta' \mathbf{1} + \delta \min(\boldsymbol{\varepsilon}, \boldsymbol{\xi})' \mathbf{1} \tag{5}$$

$$\text{s.t.} \begin{cases} (-1) f(\boldsymbol{x}_t) + \eta_t \geqslant 1,\ \eta_t \geqslant 0,\ t = 1, 2, \cdots, T_L; \\ \max\limits_{t=s_i,\cdots,e_i} f(\boldsymbol{x}_t) + \theta_{i-q} \geqslant 1,\ \theta_{i-q} \geqslant 0, \\ \qquad\qquad i = q+1, \cdots, m; \\ f(\boldsymbol{x}_t) + \varepsilon_{t-T_L} \geqslant 1,\ \varepsilon_{t-T_L} \geqslant 0, \\ \qquad\qquad t = T_L+1, \cdots, T; \\ (-1) f(\boldsymbol{x}_t) + \xi_{t-T_L} \geqslant 1,\ \xi_{t-T_L} \geqslant 0, \\ \qquad\qquad t = T_L+1, \cdots, T. \end{cases}$$

where $\eta = [\eta_1, \cdots, \eta_{T_L}]'$ are slack variables for the errors on the instances of negative bags, $\theta = [\theta_1, \cdots, \theta_p]'$ are slack variables for the errors on the positive bags, $\varepsilon = [\varepsilon_1, \cdots, \varepsilon_{T_U}]'$ and $\xi = [\xi_1, \ldots, \xi_{T_U}]'$ are slack variables for the errors on the instances of positive bags, λ, γ and δ are user-defined parameters that trade off model complexity with the errors, and $\mathbf{1} = [1, 1, \ldots, 1]'$ is vectors of 1's.

To reduce the optimization problem from a possibly infinite-dimensional space to a finite-dimensional space, representer theorem (Schölkopf & Smola, 2002) can be used. The representer theorem requires that $\Omega(||f||_{\mathcal{H}}) : [0, \infty) \to \mathbb{R}$ is a strictly monotonically increasing function. As in SVM, using $\Omega(||f||_{\mathcal{H}}) = ||f||_{\mathcal{H}}$ satisfies this requirement. Therefore, $f(\boldsymbol{x})$ can be formed as Eq. 6 where all $\alpha_t \in \mathbb{R}$.

$$f(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t k(\boldsymbol{x}, \boldsymbol{x}_t) \tag{6}$$

With the order mentioned before, a $T \times T$ kernel matrix $\mathbf{K}$ can be defined on all instances in the training set. Denote the t-th column of $\mathbf{K}$ by $\mathbf{k}_t$, then

$$f(\boldsymbol{x}_t) = \mathbf{k}'_t \boldsymbol{\alpha} + b \tag{7}$$

Using the representer theorem, the optimization problem in Eq. 5 can be rewritten as Eq. 8.

$$\min_{\boldsymbol{\alpha}, \eta, \theta, \varepsilon, \xi, b} \frac{1}{2}\boldsymbol{\alpha}' \mathbf{K} \boldsymbol{\alpha} + \lambda \eta' \mathbf{1} + \gamma \theta' \mathbf{1} + \delta \min(\boldsymbol{\varepsilon}, \boldsymbol{\xi})' \mathbf{1} \tag{8}$$

$$\text{s.t.} \begin{cases} (-1)(\mathbf{k'}_t\boldsymbol{\alpha}+b)+\eta_t \geqslant 1,\ \eta_t \geqslant 0, \\ \qquad\qquad t=1,2,\cdots,T_L; \\ \max\limits_{t=s_i,\cdots,e_i}(\mathbf{k'}_t\boldsymbol{\alpha}+b)+\theta_{i-q} \geqslant 1,\ \theta_{i-q} \geqslant 0, \\ \qquad\qquad i=q+1,\cdots,m; \\ (\mathbf{k'}_t\boldsymbol{\alpha}+b)+\varepsilon_{t-T_L} \geqslant 1,\ \varepsilon_{t-T_L} \geqslant 0, \\ \qquad\qquad t=T_L+1,\cdots,T; \\ (-1)(\mathbf{k'}_t\boldsymbol{\alpha}+b)+\xi_{t-T_L} \geqslant 1,\ \xi_{t-T_L} \geqslant 0, \\ \qquad\qquad t=T_L+1,\cdots,T. \end{cases}$$

Note that the first three terms of the objective function are convex, and the last one is concave. Except the second constraint, all constraints are linear. The second constraint can be regarded as a difference of two convex functions. Therefore, the constrained concave-convex procedure (CCCP) which has well-studied convergence properties (Smola et al., 2005) can be used to solve this optimization problem.

Both *max* and *min* are non-smooth. To use CCCP, we have to replace their gradients by subgradients. Here we adopt Fung and Mangasarian (1999)'s definition of the subgradient of $\mathbf{min}(\varepsilon,\xi)'\mathbf{1}$:

$$\partial(\mathbf{min}(\varepsilon,\xi)'\mathbf{1}) = \sum_{r=1}^{T_U} \begin{cases} \binom{I_r}{0_{T_U}} & if \quad \varepsilon_r < \xi_r \\ \binom{I_r}{0_{T_U}} + \tau\binom{0_{T_U}}{I_r} & if \quad \varepsilon_r = \xi_r \\ \binom{0_{T_U}}{I_r} & if \quad \varepsilon_r > \xi_r \end{cases} \tag{9}$$

where $0_{T_U} = [0,0,\ldots,0]'$ is a T_U-dimensional zero vector, I_r is the r-th column of the $T_U \times T_U$ identify matrix I, and $\tau \in (0,1)$. We can simply set $\tau = 0.5$.

We adopt the subgradient of $\max\limits_{t=s_i,\cdots,e_i} \mathbf{k'}_t\boldsymbol{\alpha}$ used by Cheung and Kwok (2006):

$$\partial(\max_{t=s_i,\cdots,e_i} \mathbf{k'}_t\boldsymbol{\alpha}) = \sum\nolimits_{t=s_i}^{e_i} \beta_{it}\mathbf{k'}_t \tag{10}$$

where

$$\beta_{it} \begin{cases} = 0, & \text{if } \mathbf{k'}_t\boldsymbol{\alpha} \neq \max\limits_{r=s_i,\cdots,e_i} \mathbf{k'}_r\boldsymbol{\alpha}, \\ \geqslant 0, & \text{otherwise}, \end{cases} \tag{11}$$

and

$$\sum\nolimits_{t=s_i}^{e_i} \beta_{it} = 1. \tag{12}$$

For convenience, the subgradient can be written with:

$$\beta_{it} = \begin{cases} 0, & \text{if } \mathbf{k'}_t\boldsymbol{\alpha} \neq \max\limits_{r=s_i,\cdots,e_i} \mathbf{k'}_r\boldsymbol{\alpha}, \\ \frac{1}{n_a}, & \text{otherwise}, \end{cases} \tag{13}$$

where n_a is the number of $\boldsymbol{x}_t$ that maximizes $\mathbf{k'}_r\boldsymbol{\alpha}$. At the a-th iteration, $\max\limits_{t=s_i,\cdots,e_i} \mathbf{k'}_t\boldsymbol{\alpha}$ can be replaced in the constraint by:

$$\max_{t=s_i,\cdots,e_i}(\mathbf{k'}_t\boldsymbol{\alpha}^a) + \sum_{t=s_i}^{e_i} \beta_{it}^a \mathbf{k'}_t(\boldsymbol{\alpha}-\boldsymbol{\alpha}^a) = \sum_{t=s_i}^{e_i} \beta_{it}^a \mathbf{k'}_t\boldsymbol{\alpha} \tag{14}$$

Now, we can write the optimization problem for the a-th iteration of CCCP as Eq. 15.

$$\min_{\boldsymbol{\alpha},\eta,\theta,\varepsilon,\xi,b} \frac{1}{2}\boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha}+\lambda\eta'\mathbf{1}+\gamma\theta'\mathbf{1}+\delta\partial(min(\boldsymbol{\varepsilon}^a,\boldsymbol{\xi}^a)'1)'\begin{pmatrix}\boldsymbol{\varepsilon}\\ \boldsymbol{\xi}\end{pmatrix} \tag{15}$$

$$\text{s.t.} \begin{cases} (-1)(\mathbf{k'}_t\boldsymbol{\alpha}+b)+\eta_t \geqslant 1,\ \eta_t \geqslant 0, \\ \qquad\qquad t=1,2,\cdots,T_L; \\ \sum\limits_{t=s_i}^{e_i} \beta_{it}^a\mathbf{k'}_t\boldsymbol{\alpha}+b+\theta_{i-q} \geqslant 1,\ \theta_{i-q} \geqslant 0, \\ \qquad\qquad i=q+1,\cdots,m; \\ (\mathbf{k'}_t\boldsymbol{\alpha}+b)+\varepsilon_{t-T_L} \geqslant 1,\ \varepsilon_{t-T_L} \geqslant 0, \\ \qquad\qquad t=T_L+1,\cdots,T; \\ (-1)(\mathbf{k'}_t\boldsymbol{\alpha}+b)+\xi_{t-T_L} \geqslant 1,\ \xi_{t-T_L} \geqslant 0, \\ \qquad\qquad t=T_L+1,\cdots,T. \end{cases}$$

It is evident that Eq. 15 is a standard QP problem. Following CCCP, we can do the iteration until α converges to a solution.

4. Experiments

We evaluate the performance of MissSVM on three real-world tasks, i.e. drug activity prediction, image categorization, and Web index page recommendation.

4.1. Drug Activity Prediction

The *Musk* data is a real-world benchmark for multi-instance learning, which was generated in the research of drug activity prediction (Dietterich et al., 1997) and has been well studied in this area. Here a bag represents a molecule and the instances in the bag represents the alternative low-energy shapes of that molecule. Each shape is represented by a 166-dimensional feature vector. For each molecule, if at least one of its low-energy shapes could tightly bind to the target area of some larger molecules, the molecule is qualified to make a certain drug and it is regarded as a positive bag; otherwise it is a negative bag.

There are two data sets, i.e. *Musk1* and *Musk2*, both publicly available at the UCI machine learning repository (Blake et al., 1998). *Musk1* contains 47 positive bags and 45 negative bags, and the number of instances contained in each bag ranges from 2 to 40 (5.17 in average). *Musk2* contains 39 positive bags and 63 negative bags, and the number of instances contained in each bag ranges from 1 to 1,044 (64.49 in average).

Three parameters, i.e. λ, γ, and δ, need to be specified for MissSVM. Besides, the parameter of Guanssian kernel, γ_g, also needs to be set. Here we simply set $\gamma = \lambda \times \frac{\#NegativeInstances}{\#PostitiveBags} = \lambda T_L/p$ and $\delta = \lambda \times \frac{\#NegativeInstances}{\#UnlabeledInstances} = \lambda T_L/T_U$. Thus, only λ and γ_g need to tune. For *Musk1*, we chose λ among

$\{8, 10, 12, 15\}$ and γ_g from 0.5 to 1.0 with a step size 0.05 by cross validation on the training set; for *Musk2*, for simplicity, we fixed $\lambda = 10$ based on the result on *Musk1* and only tuned γ_g. The other experiments used a similar routine to set the parameters.

On *Musk1* we performed 10-fold cross validation; on *Musk2* we conducted hold-out tests by using 1/3 data for training while the remaining data for testing, and repeated the experiments for 30 runs with random training/test partitions. The results are tabulated in Table 1. The 95%-confidence intervals are $[84.3, 90.8]$ on *Musk1* and $[72.5, 87.5]$ on *Musk2*. For comparison, Table 1 also shows the performance of many multi-instance learning algorithms reported in literature, including MILES (Chen et al., 2006), MI-LR (Ray & Craven, 2005), MIBoosting (Xu & Frank, 2004), DD-SVM (Chen & Wang, 2004), mi-SVM and MI-SVM (Andrews et al., 2003), RIPPER-MI (Chevaleyre & Zucker, 2001), RELIC (Ruffo, 2000), Citation-kNN (Wang & Zucker, 2000), Diverse Density (Maron & Lozano-Pérez, 1998), MULTINST (Auer, 1997) and Iterated-discrim APR (Dietterich et al., 1997).

Table 1. Predictive accuracy (%) on the *Musk* data

Algorithm	*Musk1*	*Musk2*
MissSVM	87.6	80.0
MILES	86.3	87.7
MI-LR	86.7	87.0
MIBoosting	87.9	84.0
DD-SVM	85.8	91.3
mi-SVM	87.4	83.6
MI-SVM	77.9	84.3
RIPPER-MI	88.0	77.0
RELIC	83.7	87.3
Citation-kNN	92.4	86.3
Diverse Density	88.9	82.5
MULTINST	76.7	84.0
Iterated-discrim APR	92.4	89.2

Table 1 shows that on the *Musk* data MissSVM is competitive with state-of-the-art multi-instance learning algorithms. In particular, it is superior to seven among the twelve compared algorithms on *Musk1*. On *Musk2* its performance is not as good as on *Musk1*, possibly because that we have not tuned λ on this data and simply used the value chosen on *Musk1*. Nevertheless, Table 1 shows that multi-instance problems can be addressed from the view of semi-supervised learning.

4.2. Image Categorization

The COREL data set described in (Chen & Wang, 2004; Chen et al., 2006) was used in this experiment, which contains 2,000 JPEG images with sizes of 384×256 or 256×384. There are twenty image categories each containing 100 images. Each image is regarded as a bag, and the ROIs (Region of Interests) in that image are regarded as instances in the bag. Each ROI is described by a nine-dimensional feature vector. We used the processed data [1] such that all the bags and instances are as same as those used in (Chen & Wang, 2004; Chen et al., 2006). Table 2 summarizes the details of this data set.

Table 2. The image categories and the average number of instances per bag (*Inst/bag*) for each category

ID	Category name	Inst/bag
0	*African people and villages*	4.84
1	*Beach*	3.54
2	*Historical building*	3.10
3	*Buses*	7.59
4	*Dinosaurs*	2.00
5	*Elephant*	3.02
6	*Flowers*	4.46
7	*Horses*	3.89
8	*Mountains and glaciers*	3.38
9	*Food*	7.24
10	*Dogs*	3.80
11	*Lizards*	2.80
12	*Fashion models*	5.19
13	*Sunset scenes*	3.52
14	*Cars*	4.93
15	*Waterfalls*	2.56
16	*Antique furniture*	2.30
17	*Battle ships*	4.32
18	*Skiing*	3.34
19	*Desserts*	3.65

The experimental routine described in (Chen et al., 2006) was adopted here. In detail, the original data set was used as two data sets. The first one (i.e. *1000-Image*) used the first ten categorizes in Table 2 while the second (i.e. *2000-Image*) used all the categorizes. For each data set, images within each category were randomly partitioned in half. One was used for training and the other for testing. Each experiment was repeated for five times for five random splits, and the average results were reported.

One-against-one strategy is employed by MissSVM for this multi-class task. The overall accuracy as well as 95% confidence intervals are reported in Table 3. Besides the results of MissSVM, the table also presents the results of some existing multi-instance learning algorithms reported in literature, including MILES (Chen et al., 2006), DD-SVM (Chen & Wang, 2004), MI-SVM (Andrews et al., 2003; Chen & Wang, 2004) and $k$means-SVM (Csurka et al., 2004).

Table 3 shows that on this task MissSVM is competitive with the compared algorithms. The confusion matrix of MissSVM on *1000-Image* is shown in Figure 1,

[1] http://www.cs.olemiss.edu/∼ychen/ddsvm.html

	Cat. 0	Cat. 1	Cat. 2	Cat. 3	Cat. 4	Cat. 5	Cat. 6	Cat. 7	Cat. 8	Cat. 9
Cat. 0	**71.6**	4.0	3.2	0.4	0.0	10.0	1.2	0.8	2.4	6.4
Cat. 1	3.2	**50.0**	1.6	6.8	0.0	2.4	0.8	0.0	35.2	0.0
Cat. 2	5.2	10.0	**62.4**	5.2	0.0	3.6	0.4	0.0	11.6	1.6
Cat. 3	2.4	1.2	3.6	**87.2**	0.0	0.0	0.0	0.0	1.6	4.0
Cat. 4	0.0	0.0	0.0	0.8	**96.4**	1.6	0.0	0.0	0.4	0.8
Cat. 5	5.6	2.4	4.0	0.0	0.0	**79.2**	0.0	0.4	8.4	0.0
Cat. 6	4.0	0.0	1.2	0.4	0.0	0.0	**86.4**	0.0	1.6	6.4
Cat. 7	6.8	0.8	0.8	0.0	0.0	7.2	0.8	**81.6**	1.6	0.4
Cat. 8	0.0	13.6	1.6	4.4	0.0	0.8	0.0	0.0	**78.4**	1.2
Cat. 9	7.6	3.6	0.0	0.4	0.0	1.6	0.0	0.0	0.0	**86.8**

Figure 1. The confusion matrix of MissSVM on *1000-Image*

Table 3. Overall accuracy (%) on image categorization

Algorithm	*1000-Image*	*2000-Image*
MissSVM	78.0: [75.8,80.2]	65.2: [62.0,68.3]
MILES	82.6: [84.1,83.7]	68.7: [67.3,70.1]
DD-SVM	81.5: [78.5,84.5]	67.5: [66.1,68.9]
MI-SVM	74.7: [74.1,75.3]	54.6: [53.1,56.1]
*k*means-SVM	69.8: [67.9,71.7]	52.3: [51.6,52.9]

where each row lists the average percentages of images in a specific category classified to each of the 10 categories. Therefore, the numbers on the diagonal show the classification accuracy for each category and off-diagonal entries indicate classification errors. Figure 1 reveals that MissSVM works well on most categorizes. The largest errors are errors between Category 1 (i.e. *Beach*) and Category 8 (i.e. *Mountains and glaciers*): 35.2% *Beach* images were misclassified as *Mountains and glaciers* while 13.6% *Mountains and glaciers* images were misclassified as *Beach*. This phenomenon has appeared in previous research (Chen & Wang, 2004; Chen et al., 2006). As Chen and Wang (2004) stated, these high classification errors are due to the fact that many images of these two categories contain semantically related and visually similar regions such as those corresponding to mountain, river, lake and ocean. Overall, the results on the image categorization task shows again that multi-instance problems can be addressed from the view of semi-supervised learning.

4.3. Web Index Page Recommendation

The Web index page recommendation data sets described in (Zhou et al., 2005b) were used in this experiment. A *Web index page* is a Web page which contains plentiful information but only provides titles or brief summaries while leaving the detailed presentation to its linked pages. 113 Web index pages were collected and labeled by nine volunteers according to their interests, and therefore there are 9 data sets. The whole data is of 30.2MB after compression. Each Web index page is regarded as a bag while its linked pages are regarded as instances. The biggest bag contains 200 instances, while the smallest one contains only 4 instances. In average, each bag contains 30.29 (3,423/113) instances. Each instance is described by the 1st to 15th most frequent terms appearing in the corresponding linked page. TFIDF are used to represent the frequent term, and normalization is performed instance by instance. For each of the nine data sets, 75 bags were randomly selected for training while the remaining 38 bags were used for testing. We used the processed data [2] such that all the bags and instances are as same as those used in (Zhou et al., 2005b). The number of positive and negative bags in the data sets is tabulated in Table 4.

Table 4. The *Web index page recommendation* data sets

Data set	Training set		Test set	
	Pos.	Neg.	Pos.	Neg.
V1	17	58	4	34
V2	18	57	3	35
V3	14	61	7	31
V4	56	19	33	5
V5	62	13	27	11
V6	60	15	29	9
V7	39	36	16	22
V8	35	40	20	18
V9	37	38	18	20

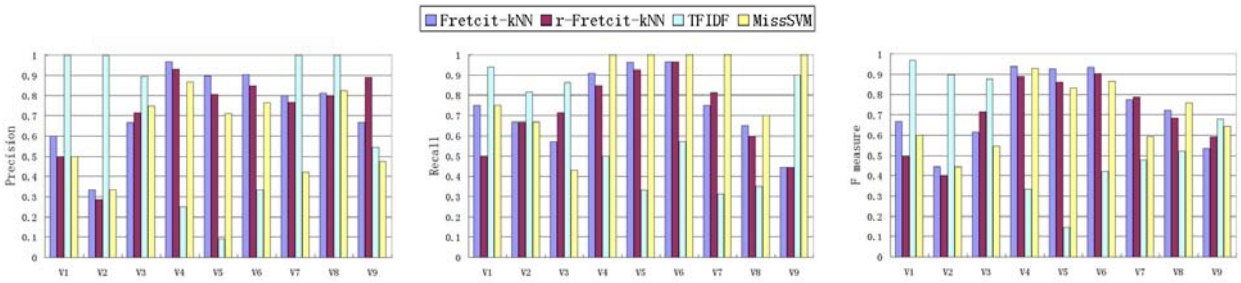
The performance of MissSVM measured by *precision*, *recall* and *F-measure* on these data sets are depicted in Figure. 2. The figure also includes the performance of Fretcit-*k*NN, r-Fretcit-*k*NN and TFIDF reported in literature (Zhou et al., 2005b). The results averaged across all the nine data sets are summarized in Table 5.

Table 5. Results averaged across the nine data sets

Algorithm	Precision	Recall	F-measure
MissSVM	0.627	0.838	0.690
Fretcit-*k*NN	0.739	0.741	0.728
r-Fretcit-*k*NN	0.727	0.720	0.704
TFIDF	0.679	0.620	0.591

Table 5 shows that MissSVM is competitive with the compared algorithms on the Web index page recom-

[2] http://cs.nju.edu.cn/people/zhouzh/zhouzh.files/publication/annex/milweb-datafile.htm



(a) Precision (b) Recall (c) F-measure

Figure 2. Precision, recall and F-measure on the nine *Web index page recommendation* data sets

mendation data sets. In particular, its recall is the best among the algorithms, and F-measure is only about 5% lower than Fretcit-kNN and 2% lower than r-Fretcit-kNN. Figure 2 reveals that on most data sets the F-measure of MissSVM is close to that of Fretcit-kNN and r-Fretcit-kNN except on $V7$ due to a poor precision. These observations verify again that multi-instance problems can be addressed from the view of semi-supervised learning.

5. Conclusion

Multi-instance learning and semi-supervised learning are two different branches of machine learning. In this paper, we establish a bridge between them by showing that multi-instance learning can be viewed as a special case of semi-supervised learning. Based on this recognition, we develop the MissSVM algorithm which tackles multi-instance problems using semi-supervised learning techniques. Experiments show that the MissSVM algorithm is competitive with many existing multi-instance learning algorithms.

The MissSVM algorithm can be easily extended to multi-instance regression (Amar et al., 2001; Ray & Page, 2001) by exploiting semi-supervised regression techniques. It is also possible to be extended to generalized multi-instance learning (Weidmann et al., 2003; Scott et al., 2003). Since MissSVM is inherently a semi-supervised learning algorithm, using it to tackle multi-instance semi-supervised learning (Rahmani & Goldman, 2006) is straightforward.

In addition to the convenience in representing some real-world objects such as the molecules (Dietterich et al., 1997), we think multi-instance learning relaxes the *i.i.d.* assumption made by traditional supervised learning. That is, in contrast to assuming that all the instances are identically and independently distributed, multi-instance learning only assumes that the bags are *i.i.d.* samples yet the instances in the bags need not to be so. For example, it is reasonable to assume *i.i.d.* molecules but not reasonable to assume that the shapes of the same molecule are identically and independently distributed. Unfortunately, most previous studies on multi-instance learning ignore this characteristic. Our result suggests that by assuming *i.i.d.* instances, multi-instance learning might become a special case of semi-supervised learning. So, it might be better to assume only *i.i.d.* bags in future research of multi-instance learning.

Acknowledgment

This work was supported by NSFC (60635030, 60325207) and FANEDD (200343).

References

Amar, R. A., Dooly, D. R., Goldman, S. A., & Zhang, Q. (2001). Multiple-instance learning of real-valued data. *ICML'01* (pp. 3–10). Williamston, MA.

Andrews, S., Tsochantaridis, I., & Hofmann, T. (2003). Support vector machines for multiple-instance learning. In *NIPS 15*, 561–568. Cambridge, MA: MIT Press.

Auer, P. (1997). On learning from multi-instance examples: Empirical evaluation of a theoretical approach. *ICML'97* (pp. 21–29). Nashville, TN.

Belkin, M., Matveeva, I., & Niyogi, P. (2001). Regularization and semi-supervised learning on large graphs. *COLT'01* (pp. 624–638). Banff, Canada.

Bennett, K. P., & Demiriz, A. (1999). Semi-supervised support vector machines. In *NIPS 11*, 368–374. Cambridge, MA: MIT Press.

Blake, C., Keogh, E., & Merz, C. J. (1998). UCI repository of machine learning databases. [http://www.ics.uci.edu/~mlearn/MLRepository.html].

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *COLT'98* (pp. 92–100). Madison, WI.

Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-supervised learning*. Cambridge, MA: MIT Press.

Chapelle, O., Sindhwani, V., & Keerthi, S. S. (2007). Branch and bound for semi-supervised support vector machines. In *NIPS 19*. Cambridge, MA: MIT Press.

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

Chen, Y., & Wang, J. Z. (2004). Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, *5*, 913–939.

Cheung, P.-M., & Kwok, J. T. (2006). A regularization framework for multiple-instance learning. *ICML'06* (pp. 193–200). Pittsburgh, PA.

Chevaleyre, Y., & Zucker, J.-D. (2001). A framework for learning rules from multiple instance data. *ECML'01* (pp. 49–60). Freiburg, Germany.

Collobert, R., Sinz, F., Weston, J., & Bottou, L. (2006). Trading convexity for scalability. *ICML'06* (pp. 201–208). Pittsburgh, PA.

Csurka, G., Bray, C., Dance, C., & Fan, L. (2004). Visual categorization with bags of keypoints. *ECCV'04 Workshop on Statistical Learning in Computer Vision* (pp. 59–74). Prague, Czech.

Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*, 31–71.

Fung, G., & Mangasarian, O. (1999). *Semi-supervised support vector machines for unlabeled data classification* (Technical Report 99-05). Data Mining Institute, University of Wisconsin at Madison, Madison, WI.

Goldman, S., & Zhou, Y. (2000). Enhancing supervised learning with unlabeled data. *ICML'00* (pp. 327–334). San Francisco, CA.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *ICML'99* (pp. 200–209). Bled, Slovenia.

Maron, O., & Lozano-Pérez, T. (1998). A framework for multiple-instance learning. In *NIPS 10*, 570–576. Cambridge, MA: MIT Press.

Maron, O., & Ratan, A. L. (1998). Multiple-instance learning for natural scene classification. *ICML'98* (pp. 341–349). Madison, MI.

Miller, D. J., & Uyar, H. S. (1997). A mixture of experts classifier with learning based on both labelled and unlabelled data. In *NIPS 9*, 571–577. Cambridge, MA: MIT Press.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*, 103–134.

Rahmani, R., & Goldman, S. A. (2006). MISSL: Multiple-instance semi-supervised learning. *ICML'06* (pp. 705–712). Pittsburgh, PA.

Ray, S., & Craven, M. (2005). Supervised versus multiple instance learning: An empirical comparison. *ICML'05* (pp. 697–704). Bonn, Germany.

Ray, S., & Page, D. (2001). Multiple instance regression. *ICML'01* (pp. 425–432). Williamstown, MA.

Ruffo, G. (2000). *Learning single and multiple instance decision trees for computer security applications*. Doctoral dissertation, Department of Computer Science, University of Turin, Torino, Italy.

Schölkopf, B., & Smola, A. J. (Eds.). (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Scott, S. D., Zhang, J., & Brown, J. (2003). *On generalized multiple-instance learning* (Technical Report UNL-CSE-2003-5). Department of Computer Science, University of Nebraska, Lincoln, NE.

Smola, A. J., Vishwanathan, S. V. N., & Hofmann, T. (2005). Kernel methods for missing variables. *AISTATS'05*. Savannah Hotel, Barbados.

Viola, P., Platt, J., & Zhang, C. (2006). Multiple instance boosting for object detection. In *NIPS 18*, 1419–1426. Cambridge, MA: MIT Press.

Wang, J., & Zucker, J.-D. (2000). Solving the multi-instance problem: A lazy learning approach. *ICML'00* (pp. 1119–1125). San Francisco, CA.

Weidmann, N., Frank, E., & Pfahringer, B. (2003). A two-level learning method for generalized multi-instance problem. *ECML'03* (pp. 468–479). Cavtat-Dubrovnik, Croatia.

Xu, L., & Schuurmans, D. (2005). Unsupervised and semi-supervised multi-class support vector machines. *AAAI'05* (pp. 904–910). Pittsburgh, PA.

Xu, X., & Frank, E. (2004). Logistic regression and boosting for labeled bags of instances. *PAKDD'04* (pp. 272–281). Sydney, Australia.

Zhang, Q., & Goldman, S. A. (2002). EM-DD: An improved multi-instance learning technique. In *NIPS 14*, 1073–1080. Cambridge, MA: MIT Press.

Zhou, D., Schölkopf, B., & Hofmann, T. (2005a). Semi-supervised learning on directed graphs. In *NIPS 17*, 1633–1640. Cambridge, MA: MIT Press.

Zhou, Z.-H., Jiang, K., & Li, M. (2005b). Multi-instance learning based web mining. *Applied Intelligence*, *22*, 135–147.

Zhou, Z.-H., & Li, M. (2005). Semi-supervised learning with co-training. *IJCAI'05* (pp. 908–913). Edinburgh, Scotland.

Zhou, Z.-H., & Zhang, M.-L. (2003). Ensembles of multi-instance learners. *ECML'03* (pp. 492–502). Cavtat-Dubrovnik, Croatia.

Zhu, X. (2006). *Semi-supervised learning literature survey* (Technical Report 1530). Department of Computer Sciences, University of Wisconsin at Madison.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *ICML'03* (pp. 912–919). Washington, DC.