Review of Multi-Instance Learning and Its applications

Jun Yang

juny@cs.cmu.edu

School of Computer Science

1. Introduction

Multiple Instance Learning (MIL) is proposed as a variation of supervised learning for problems with incomplete knowledge about labels of training examples. In supervised learning, every training instance is assigned with a discrete or real-valued label. In comparison, in MIL the labels are only assigned to *bags of instances*. In the binary case, a bag is labeled positive if *at least* one instance in that bag is positive, and the bag is labeled negative if *all* the instances in it are negative. There are no labels on the individual instances. The goal of MIL is to classify unseen bags or instances based on the labeled bags as the training data.

The study on MIL was first motivated by the problem of predicting the drug molecule activity level. After that, many MIL methods have been proposed, such as learning axis-parallel concepts [Dietterich *et al.*, 1997], diverse density [Maron and Lozano-Perez, 1998], extended Citation kNN [Wang and Zucker, 2000], etc. They have been applied to a wide spectrum of applications ranging from image concept learning and text categorization to stock market prediction. We review the several popular MIL methods and their applications in this document.

2. Multi-instance Learning Algorithms

2.1 Learning Axis-Parallel Concepts

Learning axis-parallel concepts [Dietterich *et al.*, 1997] is the first class of algorithms that were proposed to solve MIL problems. The idea is to find an axis-parallel hyper-rectangle (APR) in the feature space to represent the target concept. Intuitively, this APR should contain at least one instance from each positive bag and meanwhile exclude all the instances from negative bags. Dietterich et al. (1997) suggested three algorithms to find such a hyper-rectangle: a "standard" algorithm finds the smallest APR that bounds all the instances from positive bags; an "outside-in" algorithm constructs the smallest APR that bounds all the instances in positive bags and then shrink the APR to exclude false positives; an "inside-out" algorithm starts from a seed point and then grows a rectangle from it with the goal of finding the smallest APR that covers at least one instance per positive bag and no instances from negative bags. The three algorithms are evaluated on two real and one artificial data sets for drug activity prediction problem, and the "inside-out" algorithm is proved to be the most effective one.

2.2 Diverse Density (DD) and its EM version

Diverse Density (DD) was proposed in [Maron and Lozano-Perez, 1998] as a general framework for solving multi-instance learning problems. The main idea of DD approach is to find a concept point in the feature space that are close to at least one instance from every positive bag and meanwhile far away from instances in negative bags. The optimal concept point is defined as the one with the maximum diversity density, which is a measure of how many different positive bags have instances near the point, and how far the negative instances are away from that point.

A probabilistic derivation of Diverse Density is given below. We denote positive bags as B_i^+, and the *j*th instance in that bag as B_{ij}^+. Suppose each instance can be represented by a feature vector (or a point in the feature space), and we use B_{ijk}^+ to denote the value of the *k*th feature of instance B_{ij}^+. Likewise, B_i^- denotes a negative bag and B_{ij}^- is the *j*th instance in that bag. The true concept is a single point *t* defined by maximizing the diverse density defined as $DD(t) = P(t \mid B_1^+, \ldots, B_n^+, B_1^-, \ldots, B_m^-)$ over the feature space. Using Bayes rule and assuming uniform prior over the concept location, this is equivalent to maximizing the following likelihood:

$$\arg\max_t P(B_1^+, \ldots, B_n^+, B_1^-, \ldots, B_m^- \mid t)$$

By making additional assumption that the bags are conditionally independent given the concept point *t*, this decomposes to:

$$\arg\max_t \prod_i P(B_i^+ \mid t) \prod_i P(B_i^- \mid t)$$

Using Bayes rule once more with the uniform prior assumption, this is equivalent to:

$$\arg\max_t \prod_i P(t \mid B_i^+) \prod_i P(t \mid B_i^-)$$

which gives a general definition of the Diverse Density. Given the fact that boolean label (say, 1 and 0) of a bag is the result of "logical-OR" of the labels of its instances, $P(t \mid B_i)$ is instantiated using the noise-or model:

$$P(t \mid B_i^+) = 1 - \prod_j (1 - P(B_{ij}^+)) \text{ and } P(t \mid B_i^-) = \prod_j (1 - P(B_{ij}^-))$$

Finally, $P(t \mid B_{ij}^+)$ (or $P(t \mid B_{ij}^-)$) is estimated (though not necessarily) by a Gaussian-like distribution

$$P(t \mid B_{ij}^+) = \exp\left(-\left\|B_{ij}^+ - t\right\|^2\right) = \exp\left(-\sum_k w_k (B_{ijk}^+ - t_k)^2\right)$$

where w_k is a non-negative scaling factor that reflects the degree of relevance of different features. Without close-form solution to the above maximization problem, *gradient ascent* method is used to search the feature space for the concept point with (local) maximum DD. Usually the search is repeated using the instances from every positive bag as the starting points.

DD has been widely used for various MIL problems. It has been applied to drug activity estimation [Maron and Lozano-Perez, 1998], natural scene classification [Maron and Ratan, 1998], and image retrieval [Yang and Lozano-Perez, 2000] [Zhong *et al*, 2002]. Several extensions of the DD method have been proposed, one of which aims at learning more complicated concepts than a single point. For example, a 2-disjunctive concept can be learned by maximizing the following:

$$\arg\max_t \prod_i (1 - \prod_j (1 - P(t \vee s \mid B_{ij}^+)) \prod_j (1 - P(t \vee s \mid B_{ij}^-)))$$

where $P(t \vee s \mid B_{ij}^+)$ is estimated as $\max(P(t \mid B_{ij}^+), P(s \mid B_{ij}^+))$, and similarly for $P(t \vee s \mid B_{ij}^-)$. Besides, a EM version of the DD method [Zhang and Goldman, 2001] (see Section 2.3) and an extension of DD on real-vale domain [Amar *et al.,* 2001] have been proposed.

2.3 Expectation-Maximization version of Diverse Density (EM-DD)

In the MIL setting, the label of a bag is determined by the "most positive" instance in the bag, i.e., the one with the highest probability of being positive among all the instances in that bag. The difficulty of MIL comes from the ambiguity of not knowing which instance is the most likely one. In [Zhang and Goldman, 2001], the knowledge of which instance determines the label of the bag is modeled using a set of *hidden variables*, which are estimated using the Expectation Maximization style approach. This results in an algorithm called EM-DD, which combines this EM-style approach with the DD algorithm.

EM-DD starts with an initial guess of the concept point *t* (which can be obtained using original DD algorithm), and then repeatedly performs the following two steps: in E-step, the current hypothesis of concept *t* is used to pick the most likely instance from each bag given a generative model; in M-step, the a new concept *t'* is estimated by maximizing a transformed DD defined on the instances selected in the E-step using the gradient search. Then, the old concept *t* is replaced by the new concept *t'* and the two steps are repeated until the algorithm converges. This EM-DD algorithm implements a "hard" version of EM, since only one instance per bag is used for estimating the hypothesis. It can be also regarded as a special case of the K-means clustering algorithm, where only one cluster is considered.

By removing the noise-or (or a "softmax") part in the original DD algorithm, EM-DD turns a multi-instance problem into a single-instance one, and thus greatly reduces the complexity of the optimization function and the computational time. It is also believed to help avoid local maxima since it makes major changes on the hypothesis when it switches from one instance to another in a bag. In experiments on the drug activity problem, EM-DD outperforms other MIL algorithms like DD, Citation-kNN, and APR by a decent margin [Zhang and Goldman, 2001].

2.4 Citation kNN

The popular k Nearest Neighbor (k-NN) approach can be adapted for MIL problems if the distance between bags is defined. In [Wang and Zucker, 2000], the *minimum Hausdorff distance* was used as the bag-level distance metric, defined as the shortest distance between any two instances from each bag.

$$Dist(A,B) = \min_{\substack{1\le i\le n\\ 1\le j\le m}} (Dist(a_i,b_j) = \min_{a\in A}\min_{b\in B}\|a-b\|$$

where *A* and *B* denote two bags, and a_i and b_j are instances from each bag. Using this bag-level distance, we can predict the label of an unlabeled bag using the k-NN algorithm.

However, in a MIL setting, sometimes the majority label of the *K* nearest neighbors of an unlabeled bag is *not* the true label of that bag, mainly because the underlying prediction-generation scheme of *k*NN, *majority voting*, can be easily confused by the false positive instances in positive bags. The citation approach is used to overcome this weakness, which considers not only the bags as the nearest neighbors (known as references) of a bag *B*, but also the bags that that count *B* as their neighbors (known as citers) based on the minimum Hausdorff distance. Thus, *citation-kNN* predicts the label of a bag based on the labels of both the references and citers of that bag, which is empirically proved to be more robust than the *kNN* based on only references. Another alternative of the majority voting scheme is the Bayesian method, which computes the posterior probabilities of the label of an unknown bag based on labels of its neighbors. Experiments on the drug activity problem showed comparable performance of citation *kNN* and Bayesian *kNN* with the DD and APR algorithm. But note that unlike the DD method, *kNN* methods are unable to predict the labels of the instances.

2.5 Support Vector Machine for multi-instance learning

Andrews *et al.* (2002) proposed two approaches to modify Support Vector Machines, *mi-SVM* for instance-level classification and *MI-SVM* for bag-level classification. mi-SVM explicitly treats the label instance labels y_i as unobserved hidden variables subject to constraints defined by their bag labels Y_I . The goal is to maximize the usual instance margin jointly over the unknown instance labels and a linear or kernelized discriminant function, given below:

mi-SVM

$$\min_{\{y_i\}} \min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

$$s.t.\ \ \forall i:\ y_i\left(\langle w, x_i\rangle + b\right) \ge 1-\xi_i,\ \xi_i \ge 0,\ \ \sum_{i\in I}\frac{y_i+1}{2} \ge 1, \forall I\ s.t.\ Y_I = 1,\ \ and\ y_i = -1,\ \forall I\ s.t.\ Y_I = -1,$$

where the second part of the constraint enforce the relations between instance labels and bag labels. In comparison, *MI-SVM* aims at maximizing the bag margin, which is defined as the margin of the "most positive" instance in case of positive bags, or the margin of the "least negative" instance in case of negative bags, given as:

MI-SVM

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_I \xi_I$$

$$s.t.\ \ \forall I:\ Y_I \max_{i\in I}\left(\langle w, x_i\rangle + b\right) \ge 1-\xi_I,\ \xi_I \ge 0$$

Note that in *mi-SVM* the margin of every instance matters, and one has the freedom to set the instance label variables under the constraints of their bag labels to maximize the margin. In comparison, in *MI-SVM* only one instance per bag matters since it determines the margin of the bag. The former is suitable for tasks where users care about instance labels, while the latter is suitable for tasks where only the bag labels are concerned.

Both methods were implemented using mixed integer quadratic programming.

2.6 Multiple-decision tree

Zucker and Chevaleyre (2001) presented an extension of the C4.5 decision tree, named *multi-decision tree* (C4.5-M), to solve MIL problems. The growing of a decision tree is based on the *information gain* of a feature w.r.t. to *set* of instances, which is related to the *entropy* of the instances. They extended the concept of information gain and entropy to bags of instances in the MIL framework. Suppose S is a collection of instances which belong to $u(S)$ positive bags and $v(S)$ negative bags, F is the feature being considered as the splitting criterion, and S_v is the collection of instances whose value of feature F is v. The extended information gain and entropy are defined as:

$$Entropy_{multi}(S) = -\frac{u(S)}{u(S)+v(S)} \times \log_2\left(\frac{u(S)}{u(S)+v(S)}\right) - \frac{v(S)}{u(S)+v(S)} \times \log_2\left(\frac{v(S)}{u(S)+v(S)}\right)$$

$$InfoGain_{multi}(S, F) = Entropy_{multi}(S) - \sum_{v \in Values(F)} \frac{u(S_v)+v(S_v)}{u(S)+v(S)} \times Entropy_{multi}(S_v)$$

Note that it is the instances instead of the bags that are divided at each node of the tree, which implies that the instances of a bag may go to different branches of the tree. Moreover, the termination of tree growing is determined by the (im)purity of the instances measured by u and v. If a leaf is labeled positive if all the instances under it are from positive bags, otherwise it is labeled negative. To classify a bag, all its instances are passed through the multi-decision tree. If one positive leaf is reached by one of the instances, the bag is classified as positive, otherwise it is labeled negative.

2.7 Other approaches

Besides the aforementioned methods, a few other methods have been proposed for MIL. For example, Andrews and Hofmann (2003) proposed a MIL method based on disjunctive programming boosting, Zhou and Zhang (2002) adapted a popular neural network algorithm for MIL problems by employing a specific error function. In [Zhou and Zhang, 2003], Zhou revealed the relationships between MIL methods and their counterparts in supervised learning, such as DD versus Bayesian classifier, and hinted that many supervised learning methods can be adapted for MIL problems if their focus on discriminating instances is shifted to discriminating bags. They also suggested using ensemble methods such as bagging to combine multiple MIL learners.

3. Multi-instance Learning of Real-Value Data

Multi-instance learning was originally formulated for discrete outputs, especially for binary class labels. Recently there were efforts on developing MIL methods with real-value outputs, such as the multi-instance regression [Ray and Page, 2001] and real-value version of the *kNN* and DD methods [Amar *et al.,* 2001]. In a real-valued MIL setting, one needs to predict a real-valued label of a bag based on the training data consisting of bags labeled with real-valued output. Similarly, all the instances are not labeled.

Amar *et al.* (2001) proposed an extension of DD method for the real-value setting. The key modification is in estimating $P(t \mid B_i)$, which is now defined as:

$$P(t \mid B_i) = (1 - |l_i - Label(B_i \mid t)|)/Z$$

$$Label(B_i \mid t) = \max_j \left\{ \exp\left(-\sum_{d=1}^{n} (s_d (B_{ijd} - t_d)^2) \right) \right\}$$

where l_i is true label of bag B_i, which is a real value within [0,1] without loss of generosity, Z is the normalization constant, and $Label(B_i \mid t)$ is the label B_i would receive for the concept point *t*. Note that when $l_i \in \{0,1\}$, this reduces to a standard DD algorithm for binary label setting.

In the same paper, the authors suggested a straightforward extension of *kNN* or *Citation-kNN* for real-valued MIL. Instead of using the majority voting to decide the binary label, the average of the real-valued labels of the nearest neighbors (or references and citers, in case of Citation-kNN) of a bag in question is

computed as the real-valued label for that bag.

Multi-instance regression [Ray and Page, 2001] assumes that a linear model between the data and the real-valued output plus an additive Gaussian noise with zero mean. It also assumes that only one primary instance in each bag is responsible for the real-value label of the bag. Thus, the algorithm aims to find a hyper-plane $Y = Xa$ such that

$$a = \arg\min_a \sum_{i=1}^{n} L(y_i, X_{ip}, a)$$

where X_{ip} is the primary instance of bag X_i , L is a error function measuring the goodness of the hyperplane defined by parameters a w.r.t. the instances. Without knowing X_{ip} of each bag, the method further assumes the primary instance is the one "best fit" the current hyperplane. Under this assumption, the hyperplane can be defined as:

$$a = \arg\min_a \sum_{i=1}^{n} \min_j L(y_i, X_{ij}, a)$$

The L-error function can take many forms, such as square error $L(y_i, X_{ij}, a) = (y_i - X_{ij}a)^2$. Due to the inside minimization in above equation, there is no close-form solution to it, and searching for the optimal hyperlane was proved to be NP-hard. Here, an Expectation-Maximization-like method is used to find the primary instance of each bag under the current hyperplane (E-step), and the primary instances are used to fit the hyper-plane (M-step) in an interleaved manner.

4. Applications of Multi-Instance Learning

This section reviews the applications of MIL algorithms, including drug activity prediction, text categorization, image retrieval and classification.

4.1 Drug activity prediction

The very first MIL work [Dietterich *et al.*, 1997] was motivated by the problem of determining whether a drug molecule will bind strongly to a target protein. As the examples, some molecules that bind well (i.e., positive examples) and some that do not bind well (i.e., negative ones) are provided. A molecule may adopt a wide range of shapes or confrontations. A positive molecule has at least one shape that can bind well -- but we do not know which one. However, a negative molecule means none of its shapes can make the molecule bind well. Therefore, it is very natural to model each molecule as a bag and the shapes it can adopt as the instances in that bag. The features of an instance (shape) are the distances from an origin to different positions on the molecule surface at the corresponding shape.

Ditterich et al. (1997) provided two datasets for the drug activity prediction problem, the *Musk Data Set 1* and *Musk Data Set 2*. On these two datasets, they showed that MIL approaches significantly outperform normal supervised learning approaches which ignore the multi-instance nature of MIL problems. Later, these datasets have been extensively used as *de facto* benchmark in evaluating and comparing MIL methods [Maron and Lozano-Perez, 1998] [Zhang and Goldman, 2001][Andrews *et al.* 2002], although some argued that Axis-Parallel Concept methods are especially tuned for these datasets and thus the performance is exceptionally high.

4.2 Content-based image retrieval and classification

The key to the success of image retrieval and image classification is the ability of identifying the intended target object(s) in images. This is made more complicated by the fact that an image may contain multiple, possibly heterogeneous objects. Thus, the global description of a *whole* image is too coarse to achieve good classification and retrieval accuracy. Even if relevant images are provided, identifying which object(s) within the example images are relevant remains a hard problem in the supervised learning setting. However, this problem fits in the MIL setting well: each image can be treated as a bag of segments which are modeled as instances, and the concept point representing the target object can be learned through MIL algorithms.

Maron and Ratan (1998) partitioned natural scene pictures from the Corel Image Gallery into fixed-sized sub-images and applied DD algorithm to classify them into semantic classes. Yang and Lozano-Perez

(2000) used a similar approach for content-based image retrieval. Zhang *et al* (2002) compared both DD and EM-DD algorithms for image retrieval. Instead of partitioning images into fixed-sized regions, this work used k-means segmentation algorithms to generate more meaningful image regions. Unfortunately, there is no conclusive results showing that MIL methods is superior to traditional methods, though Yang and Lozano-Perez (2000) hinted that MIL methods are inclined to work better on "object images" rather than "natural scene images".

4.3 Text categorization

Similar to the argument made on images, a text document can consist of multiple passages that are of different topics, and thus descriptions at the document level might be too rough. Andrews *et al.*(2002) applied SVM-based MIL methods to the problem of text categorization, where each document is represented by overlapping passages consisting of 50 words in length. The work was evaluated on TREC9 document categorization sets, unfortunately without comparisons with traditional text categorization methods.

5. Concluding Remarks

Multi-instance learning has received moderate popularity after it was formally introduced in 1997. Many supervised learning methods have been adapted or extended for the MIL setting. MIL has been proved successful empirically to learning problems with label ambiguity, which exiting supervised or semi-supervised learning approaches are successful. Its continued success however depends on the significance and popularity of the applications where MIL stands out as the primary choice.

6. Reference

[Amar *et al.,* 2001] R. A. Amar, D. R. Dooly, S. A. Goldman, and Q. Zhang. Multiple-instance learning of real-valued data. Proc. of the 18th Int. Conf. on Machine Learning, pp.3-10, 2001.

[Andrews *et al.*, 2002] S. Andrews, I. Tsochantaridis, T. Hofmann. Support Vector Machines for Multiple-Instance Learning. *NIPS 2002*.

[Andrews and Hofmann, 2003] S. Andrews, T. Hofmann. Multiple Instance Learning via Disjunctive Programming Boosting, NIPS 2003.

[Dietterich *et al.*, 1997] T. G. Dietterich, R. H. Lathrop, T. Lozano-Perez. Solving the Multiple-Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence Journal,* 89, 1997.

[Maron and Lozano-Perez, 1998] Oded Maron , Tomás Lozano-Pérez, A framework for multiple-instance learning, *Proc. of the 1997 Conf. on Advances in Neural Information Processing Systems* 10, p.570-576, 1998.

[Maron and Ratan, 1998] O. Maron, A. L. Ratan. Multiple-Instance Learning for Natural Scene Classification. *Proc. 15th Int. Conf. on Machine Learning,* pp. 341-349, 1998.

[Ray and Page, 2001] S. Ray, D. Page. Multiple Instance Regression. *Proc. of 18th Int. Conf. on Machine Learning,* pp. 425-432, 2001.

[Wang and Zucker, 2000] J. Wang and J.-D. Zucker. Solving the multiple-instance problem: a lazy learning approach. Proc. 17th Int'l Conf. on Machine Learning, pp. 1119-1125, 2000.

[Xu and Croft, 1996] J. Xu and W. B. Croft. Query Expansion using Global and Local Document Analysis. SIGIR 1996.

[Yang and Lozano-Perez, 2000] C. Yang and T. Lozano-Perez. Image database retrieval with multiple-instance learning techniques. *Proc. of the 16th Int. Conf. on Data Engineering*, pp.233-243, 2000.

[Zhang and Goldman, 2001] Q. Zhang and S. A. Goldman. *EM-DD: An improved multiple-instance learning technique*. In Neural Information Processing Systems 14, 2001.

[Zhong *et al*, 2002] Q. Zhang, S. A. Goldman, W. Yu, J. E. Fritts. Content-Based Image Retrieval Using Multiple-Instance Learning. *Proc. of the19th Int. Conf. on Machine Learning,* July 2002.

[Zhou and Zhang, 2002] Z. H. Zhou, M. L. Zhang. Neural networks for multi-instance learning. In Proc. of the Int'l Conf. on Intelligent Information Technology, (2002) pp. 455-459.

[Zhou and Zhang, 2003] Z.-H. Zhou and M.-L. Zhang. Ensembles of multi-instance learners. In Proc of the 14th European Conf on Machine Learning, pages 492--501. Springer, 2003.

[Zucker and Chevaleyre, 2001] Y. Chevaleyre, J. D. Zucker, Solving multiple-instance and multiple-part learning problems with decision trees and rule sets. Application to the mutagenesis problem. In: Stroulia, E., Matwin, S. (eds.): *Lecture Notes in Artificial Intelligence*, Vol. 2056. Springer, Berlin (2001) 204-214.