ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

Labelling Multi-Instances Using Two Level Distributions Algorithm

R.Divyapriya[1], Dr.R.Manicka Chezhian[2]

Research Scholar, Computer Science (Aided), NGM College, Coimbatore, India [1]

Associate Professor, Computer Science (Aided), NGM College, Coimbatore, India [2]

Abstract: In multi-instance learning, the training set comprises labeled bags that are composed of unlabeled instances, and the task is to predict the labels of unseen bags. Multiple-instance learning is a variation on supervised learning, where the task is to learn a concept given positive and negative bags of instances. Each bag may contain many instances, but a bag is labelled positive even if only one of the instances in it falls within the concept. A bag is labelled negative only if all the instances in it are negative. This paper discusses the multi-instance problem using two-level distribution (TLD) algorithm.

Keywords: Multi-Instance Learning, Supervised Learning, Label, Two Level Distributions.

I. INTRODUCTION

Multiple-instance learning (MIL) is a generalization of supervised classification in which training class labels are associated with sets of patterns, or bags, instead of individual patterns. While every pattern may possess an associated true label, it is assumed that pattern labels are only indirectly accessible through labels attached to bags. The law of inheritance is such that a set receives a particular label, if at least one of the patterns in the set possesses the label. In the important case of binary classification, this implies that a bag is positive if at least one of its member patterns is a positive example. MIL differs from the general set-learning problem in that the set-level classifier is by design induced by a pattern-level classifier. Hence the key challenge in MIL is to cope with the ambiguity of not knowing which of the patterns in a positive bag are the actual positive examples and which ones are not[4].

The term multi-instance learning was coined by Dietterich et al. When they were investigating the problem of drug activity prediction. In multi-instance learning, the training set is composed of many bags each contains many instances. A bag is positively labeled if it contains at least one positive instance; otherwise it is labeled as a negative bag. The task is to learn some concept from the training set for correctly labeling unseen bags. Before applying learning algorithms to datasets, practitioners often globally discretize any numeric attributes. If the algorithm cannot handle numeric attributes directly, prior discretization is essential. Even if it can, prior discretization often accelerates induction, and may produce simpler and more accurate classifiers [7].

Different to supervised learning where all training instances are with known labels, in multi-instance learning the labels of the training instances are unknown; different to unsupervised learning where all training instances are without known labels, in multi-instance learning the labels of the training bags are known; different to reinforcement learning where the labels of the training instances are delayed, in multi-instance learning there is no any delay. Multiple-instance learning is a variation on supervised learning, where the task is to learn a concept given positive and negative bags of instances[8]. It has been shown that learning algorithms ignoring the characteristics of multi-instance problems, such as popular decision trees and neural networks, could not work well in this scenario.This document is a template. An electronic copy can be downloaded from the conference website. For questions on paper guidelines, please contact the conference publications committee as indicated on the conference website. Information about final paper submission is available from the conference website.

MI learning basically adopts the same setting as single-instance supervised learning

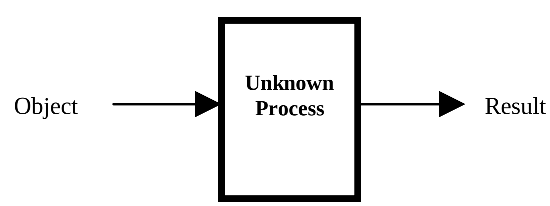

Fig 1 Single Instance Learning

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

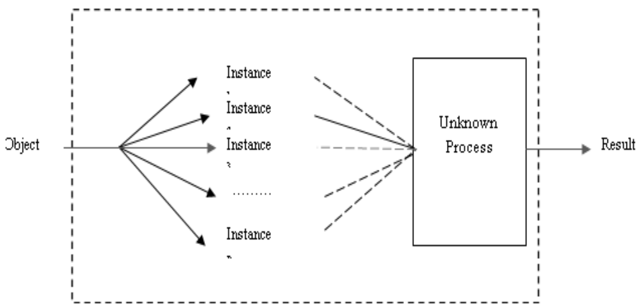

Fig 2 Multiple Instance Learning

The difference can best be depicted graphically as shown in Figure 1 [Dietterich et al., 1997]. In figure 1, the Object is an example described by some attributes, the Result is the class label and the Unknown Process is the relationship. Figure 1 depicts the case in normal supervised learning while in figure 2, there are multiple instances. We interpret the Unknown Process in Figure 2 as different from that in Figure 1 because the input is different. Note that the dashed and solid arrows representing the input of the process in figure 2 imply that only some of the input instances may be useful. Therefore while the Unknown Process in figure 1 is simply a classification problem, the Unknown Process in figure 2 is commonly viewed as a two-step process with a first step consisting of a classification problem and a second step that is a selection process based on the first step and some assumptions. Learning from structured data is becoming increasingly important. However, most prior work on kernel methods has focused on learning from attribute-Value data[10].

II. RELATED WORK

The first MI algorithm stems from the pioneering paper by Dietterich *et al.* [1997], which also introduced the aforementioned Musk datasets. The APR algorithms [Dietterich et al., 1997] modeled the MI problem as a two-step process: a classification process that is applied to every instance and then a selection process based on the MI assumption. A single Axis-Parallel hyper-Rectangle (APR) is used as the pattern to be found in the classification process. As a parametric approach,2 the objective of these methods is to find the parameters that, together with the MI assumption, can best explain the class labels of all the examples in the training data[2].Chevaleyre, Y. and Zucker, J.-D. [2000]. Solving multiple-instance and multiplepart learning problems with decision trees and decision rules**.** In recent work, Dietterich et al. (1997) have presented the problem of supervised multiple-instance learning and how to solve it by building axis-parallel rectangles. This problem is encountered in contexts where an object may have different possible alternative configurations, each of which is described by a vector. This paper introduces the multiple-part problem, which is related to the multiple-instance problem, and shows how it can be solved using the multiple-instance algorithms. These two so-called multiple problems could play a key role both in the development of efficient algorithms for learning the relations between the activity of a structured object and its structural properties and in relational learning. This paper analyzes and tries to clarify multiple-problem solving. It goes on to propose multiple-instance extensions of classical learning algorithms to solve multiple-problems by learning multiple-decision trees (Id3-Mi) and multiple-decision rules (Ripper- Mi). In particular, it suggests a new multiple-instance entropy function and a multiple-instance coverage function. Finally, it successfully applies the multiple-part framework on the well-known mutagenesis prediction problem [1]. multiple-instance learning concerns the problem of classifying a bag of instances, given bags that are labeled by a teacher as being overall positive or negative[9].

MI neural networks [Ramon and Raedt, 2000] closely adhered to the MI assumption. They adopt the same two-step framework used in the APR method [Dietterich et al., 1997] as described above. As a matter of fact, one may recognize that searching for parameters in the aforementioned two-step process is well suited for a two-level neural network architecture. Neural network is used to learn a pattern in the classification step, and a model-based instance selection method is applied in the second step. In the first step the family of patterns is not explicitly specified but implicitly defined according to complexity of the network constructed[3].

In the second step, like the most-likely-cause model in DD [Maron, 1998], the neural network picks up the instance with the highest output value in an example.8 Back propagation is used to search for the parameter values. Therefore it can be said that this method is based on the MI assumption. Indeed, the reported results obtained seemed to be very similar to those of the DD algorithm on the Musk datasets.

Blum, A. and Kalai, A. [1998] decribes a simple reduction from the problem of PAC-learning from multiple-instance examples to that of PAC-learning with one-sided random classification noise. All concept classes learnable with one-sided noise, which includes all concepts learnable in the usual 2-sided random noise model plus others such as the parity function, are learnable from multiple-instance examples[5]. Introduction One of the drawbacks of applying the supervised learning model is that it is not always possible for a teacher to provide labeled examples for training[10].

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

III. EXPERIMENT WORK

A. TLD Algorithm

First let us consider normal single-instance learning. The essence of the group conditional methods is to derive distributional properties within each group (and the group priors) so that we can decide the frequency of a point in the instance space for each group. We classify a point according to the most frequently appearing group (class) label. We follow the same line of thinking in the multi-instance (MI) case. Dependent on different groups, we first derive distributional properties for each bag, that is, on the instance level. Because even within one class, the distributional properties are different from bag to bag, we need a second-level distribution to relate the instance-level distribution to one another. We refer to the second level as the "bag level" distribution. That is why we call this approach the two-level distribution (TLD) approach. The entire document should be in Times New Roman or Times font. Type 3 fonts must not be used. Other font types may be used if needed for special purposes.

First, we introduce some notation. We denote the jth bags as bj for brevity. Formally, if bj has nj instances Y denotes the class variable. Then, given a class label Y = y (in two-class case y = 0; 1) and a bag bj , we have the distribution Pr(bjjY) for each class, which is parameterized with a fixed bag-level parameter Æy (hence we simply write down Pr(bjjY) as Pr(bjjÆy)). We estimate Æ using the maximum likelihood method:

$$\delta^{y}_{MLE} = argmaxL = argmax \prod_{j} Pr\left(b_j|\delta^{y}\right)$$

Here L is the likelihood function ,Qj Pr(bjjÆy). Now, the instances in bag bj are not directly related to Æ, as we discussed before. Instead, the instances xjk are governed by an instance-level distribution parameterized by a parameter vector, that in turn is governed by a distribution parameterized by Æ. Since is a random variable, we integrate it out in L. Mathematically,

$$L = \prod_{j} Pr\ \left(b_j|\delta^{y}\right)$$

$$= \prod_{j} \int Pr\ \left(b_j\theta|\delta^{y}\right) d\theta$$

$$= \prod_{j} \int Pr\ \left(b_j|\theta,\delta^{y}\right) P\left(\theta|\delta^{y}\right) d\theta$$

and assuming conditional independence,

$$= \prod_{j} \int Pr\left(b_j|\theta\right) Pr\left(\theta|\delta^{y}\right) d\theta$$

The complexity of the calculus would have stopped us here had we not assumed independence between attributes.

TABLE I
Error rate estimates from 10 runs of stratified 10-fold CV and LOO evaluation.

Methods	Musk 1		Musk 2	
	LOO	10CV	LOO	0CV
SVM with Mini max Kernel	7.6	8.4	13.7	13.7
RELIC	-	16.3	-	12.7
TLD Simple +EC	14.13	16.96	9.80	15.88

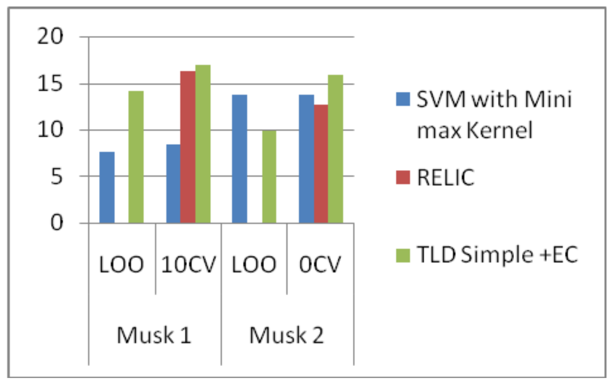

Figure 3 Error rate performances

IV. CONCLUSION

Multiple-instance learning is a variation on supervised learning, where the task is to learn a concept given positive and negative bags of instances. In this paper, multi-instance learning and TLD is used to solve the different kinds of problem and find the relationship between multi-instance learners.

REFERENCES

1. Chevaleyre,Y. and Zucker, J.-D, "Solving multiple-instance and multiple part learning problems with decision trees and decision rules",2000.
2. T.G.Dietterich,R.H.Lathrop, and T.Lozano-Perez,"Solving the multiple-instance problem with axis-parallel rectangles",Artificial Intelligence, volume 89,No.1-2,pp31-71,1997.
3. Ramon, J. and Raedt, L. D. [2000]. "Multi instance neural networks. In Attribute-Value and Relational Learning: Crossing the Boundaries", Workshop at the

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

Seventeenth International Conference on Machine Learning.

4. Stuart Andrews, Ioannis Tsochantaridis,"Support vector Machines for Multiple-Instance Learning.
5. Blum, A. and Kalai, A. "A note on learning from multiple-instance examples", Machine Learning, Volume 30(1), PP:23–30,1998.
6. Gartner, T., Flach, P., Kowalczyk, A. and Smola,"A Multi-instance kernels", In Proceedings of the Nineteenth International Conference on Machine Learning , San Francisco, pp.179–186,2004.
7. Frank, E. and Witten,I. "Making better use of global discretization". In Proceedings of the Sixteenth International Conference on Machine Learning, pp. 15–123, San Francisco, 1999.
8. Zhang, Q. and Goldman, S. EM-DD: "An improved multiple-instance learning technique", In Proceedings of the 2001 Neural Information Processing Systems (NIPS) Conference, pp. 1073–1080. Cambridge, 2002.
9. Wang, J. and Zucker Solving the multiple-instance problem: a lazy learning approach. In Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco pp. 1119–1134, 2000.
10. Maron, O. and Lozano-P´erez,"A framework for multiple-instance learning", In Advances in Neural Information Processing Systems, pp. 570–576, 1998.