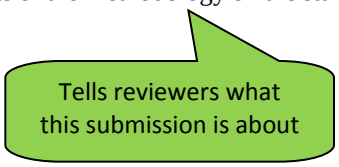
Supervised Learning in Absence of Accurate Class Labels: a Multi Instance Learning Approach

Name
Company
Contact Info

ABSTRACT

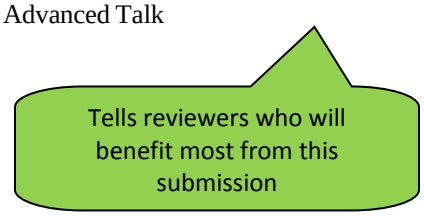
Traditionally supervised learning algorithms are built using labeled training data. Accurate labels are essential to guide the classifier towards an optimal separation between the classes. However, there are several real world scenarios where the class labels at an instance level may be unavailable or imprecise or difficult to obtain, or in situations where the problem is naturally posed as one of classifying instance groups. To tackle these challenges, we apply Multi Instance Learning (MIL) algorithms where labels are available at a bag level rather than at an instance level. We motivate the need for MIL algorithms and describe an ensemble based method, wherein the members of the ensemble are lazy learning classifiers using the Citation Nearest Neighbor method. Diversity among the ensemble methods is achieved by optimizing their parameters using a multi-objective optimization method, with the objective being to maximize positive class accuracy and minimize false positive rate. We demonstrate results of the methodology on the standard Musk 1 dataset.



.

AUDIENCE

Areas: Data Mining, Machine Learning
Advanced Talk



INTRODUCTION

Supervised learning algorithms usually have a set of input samples and corresponding labels associated with that data. The goal of building a classifier is then to find a suitable boundary that can predict correct labels on test or unseen data. A lot of research has been carried out to build robust supervised learning algorithms that can battle the challenges of nonlinear separations, class imbalances etc

However, the implicit assumption is that there exists a set of labels for the training data. This assumption may sometimes be expensive or not practical in the real world. In this paper, we would like to draw your attention towards a set of algorithms where labels are not available at an instance level but rather at a coarser level – "bag" level. A bag is nothing but a collection of instances or individual data points. A bag is labeled positive if it contains at least one positive instance (which may or may not be specifically identified), and negative otherwise. This class of problems is known as multi-instance learning (MIL) problems. This setting is applicable where data samples are not labeled because either it is difficult or time consuming for the expert or an expert chooses to label data at a coarser level instead of precise labeling for each data element. For example for medical image analysis, a radiologist may choose to label samples at an image level rather than marking a tumor region at a pixel level. Here the image can be thought of as a bag and the pixels can be thought of as instances.

In this paper, we describe a generic algorithm for building an ensemble-based method for multi instance learning. Here the members of the ensemble are lazy learning classifiers learnt using the Citation Nearest Neighbor method [1]. Diversity among the ensemble members is achieved by optimizing their parameters using a multi-objective optimization algorithm, with the objectives being to maximize positive class accuracy and minimize false positive rate.

METHODOLOGY

Consider a set of instance bags {B1,B2, …, BN }, where each bag Bi contains instances {X1i,….., XNii } and is labeled Y i ∈ {-1, +1} – we shall refer to them as positive and negative bags. The specific instance-level labels yji for Xji may be unknown, except that Y i is set to 1 if at least one of Yji is 1, and -1 otherwise. The task of the proposed algorithm is to

predict the true label Y new for an unseen bag B new. Let the prediction be denoted by $\hat{Y}^{new}$.

The broad steps followed by the proposed classifier are as follows:

1. Use an ensemble of multi instance classifiers that use the Citation Nearest Neighbor technique [1]. Let the classifiers be denoted by $C_{(j)}$, $j = 1, \ldots, J$, and the predictions from these classifiers for B^{new} be denoted by $\hat{Y}^{new}_{(j)} = C_{(j)}(B^{new})$. Each classifier uses a different set of parameters, so that diversity among the ensemble is maintained.
2. Combine the predictions $\hat{Y}^{new}_{(1)} \ldots, \hat{Y}^{new}_{(J)}$ using a normal classifier F and return the final prediction, i.e.,

$$\hat{Y}^{new} = F(\hat{Y}^{new}_{(1)}, \ldots, \hat{Y}^{new}_{(J)}).$$

Build ensemble of classifiers

We accomplish this by using a multi-objective search heuristic such as NGSA-II [4] to find the optimal CNN parameters. The search algorithm then finds the best set of parameters that optimize the following objectives:

1. Maximize the likelihood of classifying a positive instance bag correctly
2. Maximize the likelihood of classifying a negative instance bag correctly

These two objectives are estimated using a leave-one-out method. Note that these two objectives may be in conflict in any problem where perfect separability between the classes is not achievable at the given level of solution complexity. Therefore, the multi-objective search algorithm will throw up a set of candidate solutions, each of which optimizes these two objectives at varying degrees of relative importance.

Theoretically, the best possible set is known as a Pareto frontier of solutions. Any solution in the Pareto frontier cannot be considered superior to another in the frontier (i.e. if it improves on one objective, it loses on another simultaneously), but can be considered superior to all other solutions available. (Note that in this case, when we use the word solution, we refer to a parameter set for the CNN algorithm, and by performance, we refer to the ability to identify positive and negative bags correctly, as measured using the leave-one-out method.)

In practice, a multi-objective optimizer such as NGSA-II will try and arrive at a good approximation to the Pareto frontier, and will output a diverse set of solutions (i.e., parameter combinations for CNN) that optimize the two objectives at varying degrees of relative importance. These solutions constitute the ensemble we wished to construct.

Combining the CNN Classifiers

As described earlier, we construct an ensemble of CNN models in order to capture diverse views of the problem to be solved. However, the task lies before us to combine these views. The simplest method of combination would be to let all the models vote on a test instance bag, and let the majority decide the label. However, it is possible that the optimal method of combination of these diverse views (as represented by the CNN models in the ensemble) calls for a greater degree of complexity than a voting scheme.

Therefore, we propose the use of the stacked generalization method [2], wherein we build a second level classifier *F*, which will combine the predictions of the various CNN models in order to return the final prediction. *F* can be any two-class classifier such as a support vector machine, random forest etc.

OUTCOMES/RESULTS

We demonstrate the utility of our proposed method on the Musk 1 dataset taken from the UCI Machine Learning repository. This dataset describes a set of 92 molecules of which 47 are judged by human experts to be musks and the remaining 45 molecules are judged to be non-musks. The goal is to learn to predict whether new molecules will be musks or non-musks. However, the 166 features that describe these molecules depend upon the exact shape, or conformation, of the molecule. Because bonds can rotate, a single molecule can adopt many different shapes. To generate this data set, the low-energy conformations of the molecules were generated and then filtered to remove highly similar conformations. This left 476 conformations. Then, a feature vector was extracted that describes each conformation.

This many-to-one relationship between feature vectors and molecules lends itself naturally to a multiple instance problem. When learning a classifier for this data, the classifier should classify a molecule as musk if any of its conformations is classified as musk. A molecule should be classified as non-musk if none of its conformations is classified as a musk[3].

CNN models

The solution parameters to be optimized are the CNN model parameters, as well as the feature subset used to compute distance between instances. Since this is a large-scale multi-

objective optimization problem with objectives where the

gradient is ill-defined, we use a direct search method such as a multi-objective genetic algorithm to solve it. Specifically, we use the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to optimize the CKNN parameters and feature subset.

The fitness functions (Class +1 and Class 1 accuracy) are calculated for each solution (i.e., CKNN parameters and feature subset) by considering the average performance on cross-validation samples obtained using the leave-one-out method. This method has been shown to give good estimates of generalization ability [5], and would therefore help us in arriving at the best possible model.

Since NSGA-II is a multi-objective optimization method, its purpose is to generate a Pareto frontier of solutions (CKNN models), namely those which represent the best possible trade-off between the various objectives (Class +1 and Class 1 accuracy). Table 1 gives a summary of the results.

Class 0 accuracy	Class 1 accuracy	# Models
100%	91.49%	12
95.56%	95.74%	42
93.33%	97.87%	16
84.44%	100%	30

Stacked ensemble

We find that the results of the CNN algorithm do not yet approach the performance level desired by us. We therefore consider using an ensemble approach, whereby we combine the predictions of the various CNN models arrived at in the final generation of the NSGA-II run. Since these models approximate the Pareto frontier, it is possible that their combination would allow us to come up with a hybrid model that does even better on both objectives. Also, we wish to keep unrestricted, the method of combination of the CKNN predictions; therefore, we use the stacked generalization approach.

We therefore model the second level learning problem as one of mapping the predictions from the last generation of CKNN models to the desired sequence labels. We choose a Support Vector Machine classifier with a Radial Basis function kernel in order to combine the predictions. In order to optimize the and C parameters of the SVM model, as well as pick the optimal subset of CKNN models whose predictions are to be combined, we again use the NSGA-II algorithm.

Experimental results

Since NSGA-II generates a Pareto frontier of solutions, a sample of three solutions of the stacked ensemble model is given in Table 2. These results suggest that the stacking layer improves the trade-off between accuracy on the two classes.

Class 0 accuracy	Class 1 accuracy	Models
100%	93.61%	76
97.78%	100%	24

CONCLUSION

We have described a multi instance learning based methodology to tackle the problem of label deficiency. One obvious area of further work is to test it on a diverse set of problems and benchmark it against methods, as a way of validating the effectiveness of the proposed method.

PARTICIPATION STATEMENT

I will attend the conference if the paper is accepted.

REFERENCES

[1] J. Wang and J-D. Zucker. Solving the multiple instance problem - a lazy learning approach. In 17th International Conference on Machine Learning, 2000.
[2] D. H. Wolpert. Stacked generalization. Neural Networks, 5:241–259, 1992.
[3] K. Bache and M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.
[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation, 6(2), 2002.

Content was previously limited to 3 pages but has been expanded to 4

BIO

Presenter Bio

Bios should not be generic but must pertain to the session.