A Probabilistic Approach to Multiple-Instance Learning

Silu Zhang(✉), Yixin Chen, and Dawn Wilkins

Department of Computer and Information Science,
The University of Mississippi, University, MS 38677, USA
{ychen,dwilkins}@cs.olemiss.edu, szhang6@go.olemiss.edu

Abstract. This paper introduced a probabilistic approach to the multiple-instance learning (MIL) problem with two Bayes classification algorithms. The first algorithm, named Instance-Vote, provides a simple approach for posterior probability estimation. The second algorithm, Embedded Kernel Density Estimation (EKDE), enables data visualization during classification. Both algorithms were evaluated using MUSK benchmark data sets and the results are highly competitive with existing methods.

Keywords: Multiple-instance learning · Non-linear dimensionality reduction · Data visualization

1 Introduction

Machine Learning approaches have been widely applied in drug activity prediction [2,12], i.e., predicting whether an unknown drug will bind to a target (protein) based on existing knowledge of drugs-protein interactions. The multiple-instance learning (MIL) problem arises when a drug has more than one conformation and several of those can bind to the target. However, only the labels of drugs in the training set are given: a drug is positive if at least one of its conformations binds to the target, and negative otherwise. The label of each conformation is unknown. The task is to predict the label of an unseen drug (i.e., bag) given its conformations (i.e., instances). MIL is also applied in gene function prediction at the isoform level [5]. In this context, a gene is considered as a bag, which consists of multiple isoforms, referred as instances.

Of existing MIL algorithms, one class is based upon learning the labels of instances and then labelling the bag using instance label information. The assumption typically used is that a bag is positive if it has at least one positive instance and negative if all of its instances are negative [1,4]. A different assumption is that all instances contribute equally and independently to a bag's label, and the bag label was generated by combining the instance-level probability estimates [10,11,13]. There are also many methods that convert the MIL problem to a supervised learning problem using feature mapping [3]. However,

Z. Cai et al. (Eds.): ISBRA 2017, LNBI 10330, pp. 331–336, 2017.
DOI: 10.1007/978-3-319-59575-7_30

feature mapping usually results in increased dimensionality, and a commonly used approach to overcome this problem is feature selection.

In this paper, we develop two Bayes classifiers for MIL. The first approach, named Instance-Vote, attaches the bag label to its instances for all bags in the training set. For any new bag, we simply use a k-NN classifier to predict the label for each instance in the bag, followed by estimating the probability of the bag being positive via the percentage of positive instances in the bag. The second algorithm, named as EKDE (Embedded Kernel Density Estimation) converts the MIL to a supervised learning problem by mapping each bag into an instance-defined space. Instead of feature selection, a non-linear dimensionality reduction method, t-SNE (t-Distributed Stochastic Neighbor Embedding), is then used to reduce the dimension to 1 or 2. The advantage of this approach is the capability of data visualization. The class conditional probability densities are then estimated in this low dimensional space by kernel density estimation (KDE). The classification is based on the posterior probability according to Bayes' theorem.

2 Methodology

2.1 MIL via Instance-Vote

A New Interpolation of Instance Label. The main challenge of MIL is that the label of instances are unknown. The classical MIL assumption treats a bag as negative if none of its instances is positive. Here we relax the assumption by allowing negative bags to contain positive instances. In order to predict instance labels, we assign bag label to all its instances in the training set. We then use k-NN classifier to predict instance labels. This above process of generating instance-based training data clearly introduces noise into instance labels. However, the noise can be accounted for by the following voting model and the choice of threshold parameter.

Voting for Bag Label. To classify a bag, all its instances cast a vote based on the instance label. We assume that the posterior probability of a bag being positive (or negative) is a monotonically non-decreasing function of the probability of a randomly chosen instance from the bag being positive (or negative), i.e.,

$$\Pr(y = +|B) = f(\Pr(x_i \in +|B))$$

where y is the label of bag B, x_i is a randomly chosen instance from the bag, f is an unknown monotonically non-decreasing function. The maximum likelihood estimate of $\Pr(x_i \in +|B)$ is obtained as $\frac{m^+}{m}$, where m^+ is the number of positive instances in the bag and m is the total number of instances in the bag. We use a simple Bayes decision rule for classification, i.e.,

$$y = \begin{cases} + & \text{if } \Pr(x_i \in +|B) > \theta, \\ - & \text{otherwise,} \end{cases}$$

where θ is the threshold parameter.

2.2 Embedded Kernel Density Estimation

In this approach, we convert the MIL problem to supervised learning via feature mapping. We aim to find the probability distributions of the two classes using KDE and then apply the Bayes decision rule. However, KDE does not perform well for high dimensional data, since data points are too sparse in high dimensional space. The solution is to learn an embedding of the data and apply KDE in the low dimensional latent space ($d = 1$ or 2). Therefore, we name this approach as Embedded Kernel Density Estimation (EKDE).

Feature Mapping. We adopt the same method described in [3] considering its good performance. Each bag is represented by all the instances in the training set via a similarity measurement. The similarity of a bag B_i and an instance x^k is defined as:

$$s(B_i, x^k) = \max_j \exp(-\frac{\left\|x_{ij} - x^k\right\|^2}{\sigma^2}),$$

where x_{ij} is the j'th instance in bag B_i with $j = 1, \ldots, n_i$, n_i is the number of instances in bag B_i, and σ is a predefined scaling factor. Then bag B_i can be represented as: $[s(B_i, x^1), s(B_i, x^2), \ldots (B_i, x^n)]$, where n is the total number of instances in the training set, i.e., $\sum_{i=1}^{l} n_i = n$, where l is the total number of bags in the training set. The dimension after feature mapping is now n, which can be considerably large. Therefore, dimensionality reduction is desired.

Dimensionality Reduction and Visualization. Among all existing dimensionality reduction techniques, t-SNE was chosen due to its prominent performance [8]. Specifically, we chose the parametric t-SNE since it provides a mapping function from the original space to the low dimensional space [7]. The dimension of latent space was set to 1 or 2 such that KDE can be reliably implemented and visualization of the data can also be achieved. Although not required by classification, visualization is beneficial for data analysis.

Probability Density Estimation and Classification. According to Bayes' theorem, given a bag represented as x, the posterior probabilities can be computed as

$$\Pr(y = +|x) = \frac{p(x|y = +)\Pr(y = +)}{p(x)}, \Pr(y = -|x) = \frac{p(x|y = -)\Pr(y = -)}{p(x)},$$

where y is the bag label. Assuming bags being i.i.d., the maximum likelihood estimates of $\Pr(y = +)$ and $\Pr(y = -)$ are $\Pr(y = +) = \frac{l^+}{l}, \Pr(y = -) = \frac{l^-}{l}$, where $l^+(l^-)$ is the number of positive(negative) bags in training set. The class conditional densities $p(x|y = +)$ and $p(x|y = -)$ can be estimated by KDE using training data after dimensionality reduction. $p(x)$ is a constant respect to y. The classifier can make predictions on the bag label y by setting a threshold θ for the odd ratio (OR):

$$y = \begin{cases} + & \text{if OR} > \theta, \\ - & \text{otherwise,} \end{cases}$$

where $\text{OR} = \frac{\Pr(y=+|x)}{\Pr(y=-|x)}$.

3 Experimental Results

3.1 Data Sets

The benchmark datasets MUSK1 and MUSK2 are used in our study. In these two datasets, each molecule (bag) has more than one conformation (instance). The label of the molecule is "musk" (positive) if any of its conformations is a musk or "non-musk" if none of its conformations is a musk.

3.2 Experimental Setup

For the Instance-Vote algorithm, different values of k were tested for k-NN classifier instance classification. For the EKDE algorithm, the setup of feature mapping is same as [3]. We used the implementation of parametric t-SNE that is publicly available at [9]. A Gaussian kernel was used in KDE and the optimal bandwidth was determined by 10-fold cross validation.

3.3 Results

Both algorithms were tested using 10-fold cross-validation at the bag level. We use area under ROC (receiver operating characteristics) curve (AUC) for evaluation since it is a preferred measurement over accuracy [6]. Each experiment was performed 10 times and the average AUC was used for comparison with other algorithms.

In the testing of Instance-Vote algorithm, $k = 3$ gives the optimal cross-validation result for both data sets. The AUC obtained is shown in Table 1. The result is surprisingly good considering the simpleness of this algorithm. This may suggest that noise introduced during labelling instances in the training set is not significant. From the chemistry point of view, it is reasonable that many conformations of a musk molecule can preserve the musk property.

We next present the results of the EKDE algorithm. After feature mapping, the data dimensions are 476 (MUSK1) and 6598 (MUSK2), as determined by the total number of instances in the training sets. The dimension is then reduced to 1 or 2 by applying parametric t-SNE. Due to the limit of space, we only show the visualization results in 2D (Fig. 1). The two classes are separated well for both data sets with minor overlapping in MUSK2, thanks to the superiority of t-SNE on preserving the local structure. The AUC results are shown in Table 1.

For comparison, we also include the results of various existing MIL algorithms that use AUC as the measure for evaluation (Table 1). Among all of the listed method, Instance-Vote is the simplest and has comparable results with the others. The EKDE algorithm outperforms others on MUSK1 and is comparable with those on MUSK2.

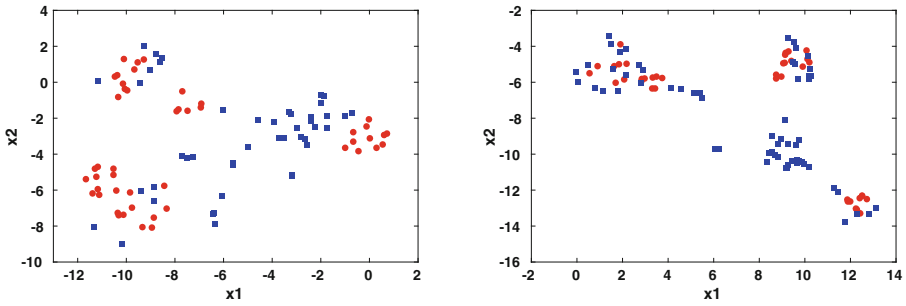

Fig. 1. Visualization of MUSK1 (left) and MUSK2 (right) data sets in 2D. Positive and negative bags are presented as red circles and blue squares, respectively. (Color figure online)

Table 1. AUC obtained by the proposed algorithms and other methods on the MUSK data set (All listed algorithms were evaluated by 10-fold cross-validation.).

Algorithms	MUSK1	MUSK2
Instance-Vote	0.921	0.856
EKDE ($d = 1$)	**0.954**	0.859
EKDE ($d = 2$)	0.941	0.865
MI RVM [11]	0.942	**0.987**
RVM [11]	0.951	0.985
MI Boost [11,13]	0.899	0.964
MI LR [10,11]	0.846	0.795
DD(1) [10]	0.895	0.903
DD(3) [10]	0.883	0.850
DD(5) [10]	0.861	0.838

4 Conclusions

In this paper, we introduced two Bayes algorithms to solve the multiple-instance problem. The Instance-Vote algorithm acquires the label of each instance in the training set from its associated bag and predict an unseen bag label base on the percentage of predicted positive instances in the bag. The EKDE algorithm performs KDE after feature mapping in the embedded low dimensional space with the help of parametric t-SNE. In this approach, both classification and data visualization can be achieved. We have shown that both algorithms are competitive with other MIL algorithms on MUSK benchmark data sets.

Acknowledgements. This work was supported by the Department of Computer and Information Science, University of Mississippi.

References

1. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Advances in Neural Information Processing Systems, pp. 577–584 (2003)
2. Burbidge, R., Trotter, M., Buxton, B., Holden, S.: Drug design by machine learning: support vector machines for pharmaceutical data analysis. Comput. Chem. **26**(1), 5–14 (2001)
3. Chen, Y., Bi, J., Wang, J.Z.: Miles: multiple-instance learning via embedded instance selection. IEEE Trans. Pattern Anal. Mach. Intell. **28**(12), 1931–1947 (2006)
4. Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artif. Intell. **89**(1), 31–71 (1997)
5. Eksi, R., Li, H.D., Menon, R., Wen, Y., Omenn, G.S., Kretzler, M., Guan, Y.: Systematically differentiating functions for alternatively spliced isoforms through integrating RNA-seq data. PLoS Comput. Biol. **9**(11), e1003314 (2013)
6. Huang, J., Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. IEEE Trans. knowl. Data Eng. **17**(3), 299–310 (2005)
7. van der Maaten, L.: Learning a parametric embedding by preserving local structure. RBM **500**(500), 26 (2009)
8. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(Nov), 2579–2605 (2008)
9. Matten, L.: t-SNE. https://lvdmaaten.github.io/tsne/
10. Ray, S., Craven, M.: Supervised versus multiple instance learning: an empirical comparison. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 697–704. ACM (2005)
11. Raykar, V.C., Krishnapuram, B., Bi, J., Dundar, M., Rao, R.B.: Bayesian multiple instance learning: automatic feature selection and inductive transfer. In: Proceedings of the 25th International Conference on Machine Learning, pp. 808–815. ACM (2008)
12. Warmuth, M.K., Liao, J., Rätsch, G., Mathieson, M., Putta, S., Lemmen, C.: Active learning with support vector machines in the drug discovery process. J. Chem. Inf. Comput. Sci. **43**(2), 667–673 (2003)
13. Xu, X., Frank, E.: Logistic regression and boosting for labeled bags of instances. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS, vol. 3056, pp. 272–281. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24775-3_35