Project Report for:

Mutiple Instance Learning for Drug Activity Prediction

Ridvan Eksi, Raj Tejas Suryaprakash and Aria Ghasemian Sahebi

December 16, 2011

1 Introduction

In a standard supervised learning model, it is assumed that we are provided with a set of training instances and the complete set of corresponding labels. The objective of such a model is to classify any newly observed instance. In certain applications, it is not possible to observe the labels for all instances within the training set.

Multiple instance learning provides a new approach to classification when only partial information about the labels are provided. In this model, rather than a set of instances and the corresponding labels, we are provided with a number of *bags* each of which consists of several instances. A bag is labeled positive if there exists a positive instance within the bag and it is labeled negative if all the instances within the bag are negative. We are provided with the bag labels rather than instance labels and the objective is to classify newly observed instances or bags.

This problem was first studied for drug activity prediction. Each molecule may be either *active* or *inactive*, and in both cases, it can exist in one of many possible *conformations*. A molecule conformation can be represented by a vector of features (typically of size around 150) which represent the topological characteristics of the molecule. If an active molecule exists in a conformation that is conducive for forming a bond, it forms an active drug, or else, the resulting compound is passive. The challenges here are

- To predict if a new compound (bag) is an active drug
- To identify the conformations of molecules which are active

The figure below illustrates a set of molecules partitioned into three separate bags:

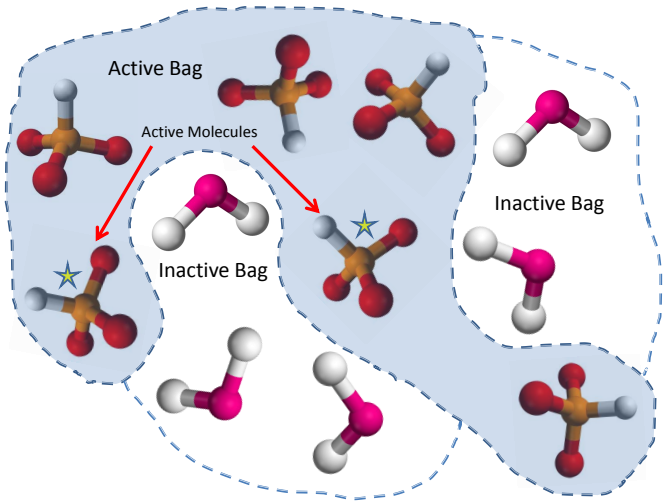


Figure 1: Drug Activity Prediction: The shaded area forms a positive (active) bag since there exist positive instances (molecule conformations) within this area. The two white areas are negative (inactive) since they dont contain any active molecule conformations.

In the example of this figure there are three bags separated by dashed borders. The shaded bag consists of two active molecule conformations and hence is labeled as active (bag label is 1) while the other two consist of only inactive molecule conformations and are labeled inactive (bag label is 0). Experimentally we are only able to observe drug formation due to active bags but we have no information about which conformations (feature vectors) led to the formation of the active drug.

The problem of drug activity prediction has great applications in medical sciences and has been an area of great interest for biomedical research scientists. In this project we focused on multiple instance learning for drug activity prediction. We studies several papers on the subject and implemented several existing methods (some of which were available online). We have also proposed and implemented several modifications and developed new approaches to the problem.

In Section 2, we state the problem in more details and introduce our notation. We then present some known results and related works in Section 3 and in Section 4 we try to give an interpretation of the problem which relates it to standard classification problems. In Section 5 we present our original contributions and modifications to the existing algorithms. We provide numerical results and a comparison of existing and new approaches in Section 6 and finally we conclude in Section 7.

2 Problem Statement

We may cast the above problem as a binary MIL problem by associating molecules with bags, and their conformations with individual instances. Formally, Consider a set of input patterns $x_1, x_2, \ldots, x_n$ and the corresponding binary labels $y_1, y_2, \cdots, y_n$. Also, for an integer k, consider subsets $I_1, I_2, \cdots, I_k$ of the index set $\{1, 2, \cdots, n\}$ with the property that $I_1 \cup I_2 \cup \cdots \cup I_k = \{1, 2, \cdots, n\}$. For $i = 1, \cdots, k$, define the i^{th} bag B_i as $B_i = \{x_j | j \in I_i\}$; In other words, bag i contains all the instances indexed by the index set I_i. Also for $i = 1, \cdots, k$, define the label Y_i of the bag B_i as $Y_i = \max_{j \in I_i} y_j$. This definition

is equivalent to:

$$Y_i = \begin{cases} 1 \text{ if } \exists j \in I_i : y_j = 1 \\ 0 \text{ if } \forall j \in I_i : y_j = 0 \end{cases} = \begin{cases} 1 \text{ if } \exists x_j \in B_i \text{ with label } y_j = 1 \\ 0 \text{ if } \forall x_j \in B_i \text{ label } y_j = 0 \end{cases}$$

In the drug activity problem, each molecule is represented by an index $i = 1, \cdots, k$ and the bag B_i represents the set of all conformation of that particular molecule. The label Y_i of B_i indicates the status of the molecule; i.e. $Y_i = 1$ if there exists an active conformation of the molecule and $Y_i = 0$ otherwise. Also, for $j = 1, \cdots, n$, the instance x_j represents a conformation of a molecule and it's label y_j represents its status; i.e. $y_j = 1$ if that particular conformation of the molecule is active and $y_j = 0$ otherwise.

In standard binary classification problems, for $j = 1, \cdots, n$, the label y_j of the instance x_j in the training set is observed and the objective is to classify a set of newly observed instances as accurately as possible. In the multiple instance learning problem however, the complete information about instance labels is not available and only bag labels are observed. Formally, we are given a set of bags $B_1, \cdots, B_k$ and their labels Y_i and the objective is to classify newly observed bags or instances.

3 Related Work

We surveyed three different algorithms from the literature that attempt to solve the MIL problem. One method [2] is specific to the data of our interest, which is MIL in drug activity prediction. Two algorithms, based on k Nearest Neighbors (kNN) [4] and Support Vector Machines (SVMs) [1] are for general MIL problems and do not make any additional assumptions regarding the datasets. We will briefly summarize the approach taken by all three papers here.

3.1 Axis Parallel Rectangles Method

This is an algorithm proposed by [2]. This method tries to indentify the feature vectors within a bag that result in a positive label for the bag. By visualizing the dataset (the same dataset used in this project), the authors conclude that, given a feature vector x_i, the feature vector is active ($y_i = 1$) with high probability, if each element the feature vector is within a range of values. Geometrically, this constraint, $x^j_{min} \leq x^j_i \leq x^j_{max}$ for some constants, x^j_{min} and x^j_{max}, can be visualized as setting a hyper-rectangular bound on feature vectors. The algorithm then identifies the positive feature vectors as follows: beginning at the outer most boundary of the hyper-rectange (i.e. considering the entire range of values of each element in a feature vector), the algorithm progressively sets bounds for each dimension of the feature vector, such that all the negative feature vectors are discarded. This selects many instances of positive feature vectors (which lie within the selected rectangle or range of values), and rejects many of the negative feature vectors. Further, the authors prove that bounding the elements of the feature vector, also called as defining an axis parallel rectangle (APR), which covers at least once instance of every positive molecule contains the optimum bound in terms of selection of positive to rejection of negative feature vectors. The main aim of the paper is to identify the feature vectors which cause the assigned label to bags, i.e. to estimate the function g in section 2. The two ways this is done are

- To build an outer APR that includes all positive and some negative features. This APR is then trimmed along each dimension to obtain a rectangle around only positive features, by rejecting the positive and negative features at the edge of the APR with some weighted cost per positive feature rejected.
- To select any one positive feature and build an APR from inside out.

3.2 Modified kNN

This algorithm is proposed by [4]. This algorithm tries to classify new bags rather than identify individual feature labels, i.e. it tries to estimate the function f is section 2. They use an extension of the kNN principle from class, and they now consider "k nearest bags". Accordingly, they also modify the distance measure for the appropriate the distance for bags. The distance between two bags is defined in terms of the Hausdorff distance - given two sets of points $A = \{a_1, a_2, \ldots, a_m\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, Hausdorff distance is defined as

$$H(A, B) = max\{h(A, B), h(B, A)\}$$

where

$$h(A, B) = \max_{a \in A} \min_{b \in B} ||a - b||$$

A k-level instead of max- Hausdorff distance is used to make the algorithm robust to noise.

3.3 Citation kNN

The 'citation k- nearest neighbors' (ckNN) algorithm was proposed by [4]. It is an extension of the k nearest neighbors algorithm to solve the multiple instance learning problem (MIL). The ckNN algorithm extends the kNN algorithm for MIL problems, described in 3.2. The distance between two bags is measured in terms of the Hausdorff distance. The extension to kNN algorithm is the concept of 'citations' and 'references'. For instance, research papers usually refer to previously published work, known as 'reference'. When the same paper is referenced by future work, it becomes a 'citer' for the future work. Following a similar strategy, this algorithm posits that classification of a new bag B depends upon the k- nearest bags of B (references), and also the bags which consider B as a neighbor (citations). Consider the k nearest references, denoted R, and the c-nearest citations, denoted by C, of a test bag. Of these, let R_p references and C_p citations have positive labels, while R_n references and C_n citations have negative labels. $(R_p + R_n = R, C_p + C_n = C)$. The classification rule is,

$$Y_i = \begin{cases} 1 \text{ if } R_p + C_p > R_n + C_n \\ 0 \text{ else} \end{cases} \tag{1}$$

3.4 SVM Method

This is proposed by [1]. One approach is to treat the pattern labels as unobserved integer variables, subjected to constraints defined by the (positive) bag labels. The goal then is to maximize the usual pattern margin, or soft-margin, jointly over hidden label variables and

a linear (or kernelized) discriminant function. The second approach generalizes the notion of a margin to bags and aims at maximizing the bag margin directly. For convenience in notation, in this part we assumes binary labels are represented by $\{-1, 1\}$ rather than $\{0, 1\}$. The primal form of objective function is

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

$$\text{s.t. } \forall i : y_i(< w, x_i > +b) \geq 1 - \xi_i, \xi_i \geq 0, y_i \in \{-1, 1\}, \text{ and equation (1) hold}$$

In the class problems, all x_i and y_i were given, but here y_i belonging to positive bags are treated as unknown variables. In the other formulation, the functional margin of a bag with respect to a hyperplane is defined by

$$\Gamma_i = Y_i \max_{j \in I_i}(< w, x_j > +b)$$

Using this "bag margin", the SVM optimization problem becomes

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

$$\text{s.t. } \forall i : y_i \max_{j \in I_i}(< w, x_j > +b) \geq 1 - \xi_i, \xi_i \geq 0$$

For negative bags, if $Y_i = -1$, the constraint is read as $- < w, x_i > -b \geq 1 - \xi_i, \forall j \in I_i$. For positive bags, a selector variable $s(i) \in I_i$ is introduced which denotes the pattern selected as positive "witness" in B_i. This results in constraints $< w, x_{s(i)} > +b \geq 1 - \xi_i$ and every bag B_i is effectively represented by a single member pattern $x_{s(i)}$.

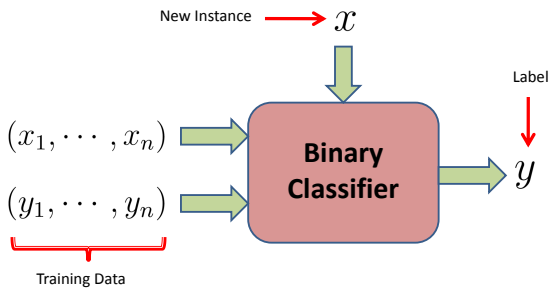
4 A Mathematical Discussion

It is evident that multiple instance learning is a generalization of a binary classification problem. In fact, when the size of the bags are all equal to one, the multiple instance learning simplifies to a standard binary classification problem. On the other hand, we claim that from a theoretical point of view, the complete characterization of the solution of the multiple instance learning problem can be derived from the solution of the standard classification problem. We provide a somewhat hand-wavy argument below.

Consider the following generalization of the binary classification problem: Instances $(x_1, \cdots, x_n)$ are observed and for $i = 1, \cdots, n$, the instance x_i is associated with an unobserved label y_i. Let k be an integer and for $j = 1, \cdots, k$ let R_j be an arbitrary set. For $j = 1, \cdots, k$, define the function $f_j : \{0, 1\}^n \to R_j$ which takes instance labels $(y_1, \cdots, y_n)$ to an element r_j of R_j. We have partial information about the instance labels through functions $f_1, f_2, \cdots, f_k$ of $(y_1, \cdots, y_n)$. In other words, the inputs to our classification algorithm are $x_1, \cdots, x_n$ and $r_1, \cdots, r_k$ while the deterministic functions $f_1, \cdots, f_k$ are also known. We assume no further knowledge about instance labels. The objective is to label newly observed instances.

This problem can be brought into a standard classification framework through the following argument: Let $\mathcal{Y}(r_1, \cdots, r_n) \subseteq \{0, 1\}^n$ be the set of all instance labels $(y_1, \cdots, y_n) \in \{0, 1\}^n$ for which $f_j(y_1, \cdots, y_n) = r_j$ for $j = 1, \cdots, k$. If we assume an element $(y_1, \cdots, y_n)$ of this set as the realized but unobserved vector of instance labels, the problem simplifies

to a standard classification problem and can be solved using standard binary classification methods. Since other than the constraints imposed by functions $f_1, \cdots, f_k$, we have no further knowledge about instance labels, all of the elements of $\mathcal{Y}(r_1, \cdots, r_n)$ are equi-likely the realized instance labels. This argument suggests that in order to classify a newly observed instance x, we need to do the binary classification assuming each of $(y_1, \cdots, y_n) \in \mathcal{Y}(r_1, \cdots, r_n)$ as the realized vector of instance labels and then take the majority vote among all the $|\mathcal{Y}(r_1, \cdots, r_n)|$ labels obtained where $|\cdot|$ denotes the cardinality or size of its argument. The figure below shows a block diagram of a standard binary classifier:



For the case of multiple instance learning, the second input of the classifier is "uniformly distributed" over $\mathcal{Y}(r_1, \cdots, r_n)$ and hence the output is also uniformly distributed over the outputs of $|\mathcal{Y}(r_1, \cdots, r_n)|$ standard binary classifiers.
Since the number of all possible instance labels ($|\mathcal{Y}(r_1, \cdots, r_n)|$) can be exponentially large, one way to avoid a high computational cost is to run binary classification algorithms for only a small number of times for random $(y_1, \cdots, y_n)$'s chosen uniformly from the set $\mathcal{Y}(r_1, \cdots, r_n)$.
Although this argument is not very rigorous, our numerical results show that this method has a decent performance compared to other existing algorithms.

5 Contribution

We have explored several algorithms and tried several modifications and new ideas; some of which had a good performance and some didn't. We have listed them all in this section.

5.1 An Alternative Initialization for the SVM method

The SVM method uses heuristics to search the optimum assignment for instance labels. The initial assignment is simply to assume that all the instances within positive bags are positive and all the instances within negative bags are negative. By this initialization, there is no guarantee that the algorithm will converge to the global minimum, It fact with a high chance it will get stuck in a *local* minimum. One possible solution to this problem is to initialize the instance labels using a more intelligent way; namely to use a variant of kNN approach for the initialization. We believe that this method will converge

to the global minimum more often than the existing method and we are yet to confirm it using the dataset. In a different approach, If we assume that the instances follow a fixed probability density, we can have a better guess for the instance labels based on probabilistic arguments.

5.2 Splitting Negative Bags

Since all the instances contained in a negative bag are negative, there is no loss of information if we split a negative bag to smaller negative bags (This operation is invertible; i.e. we can again combine the two bags and get the original training data). We used this observation to modify some of the existing algorithms. It turns out that this method can improve the accuracy of some of the methods significantly.

5.3 Randomized Binary Classification

This method is described in Section 4. We implemented this approach using two standard classification approaches: SVM and kNN. We ran each of these classification algorithms 50 times using instance labels randomly chosen from the set of possible labels. We refer to these algorithms as "randomized kNN" and "randomized SVM".

6 Numerical Results

6.1 Evaluation Plan and Data

The data is provided by Ditterich et al. for the drug activity prediction problem in two separate versions and is available online at the Machine Learning Repository website:

- Link 1: http://archive.ics.uci.edu/ml/datasets/Musk+(Version+1)
- Link 2: http://archive.ics.uci.edu/ml/datasets/Musk+(Version+2)

The difference in these two datasets is that in the first one, a higher fraction of bags are labeled as positive than the second dataset.

6.2 Results and Comparison

6.3 An Alternative Initialization for the SVM method

For the SVM method of Section 5.1 we tried several different initializations for labels of instances in positive bags. We observed that SVM classifier is highly sensitive to the percentage of negative labeled instances in positive bags. It is stated in [3] that instance based SVM method is heavily biased towards problems with very little ambiguity (i.e. problems with small number of negative labeled instances in positive bags). After a certain threshold for the ratio of negative instances to positive instances in a bag, SVM does not perform well. Up to this threshold, different initializations yield to the same performance as the original initialization (all positive labels for instances in positive bags). From these findings, we conclude that instance based SVM method is inapplicable for a general class of datasets, since it is biased towards problems that have small percentage of

negative instances in positive bags. Secondly, even if the dataset of interest follows that assumption, different initializations does not yield to better performance with respect to initialization where all instances in positive bags are labeled as positive.

6.4 Splitting Negative bags

The plot 6.5 shows the effect of splitting negative bags on performance accuracy. From the graph, we conclude that splitting all the negative bags into individual bags has the advantage of improving the performance. We justify this by noting that, since we are operating at bag levels, splitting the negative bags effectively increases our training set by giving us more bags for which we know the label (0, since we are certain that all features had 0 label as they belonged to negative bags). While an algorithm working at individual feature level may not have this advantage (since the number of features remains the same), an algorithm working at the bag level has the advantage of seeing an increase in the number of bags.

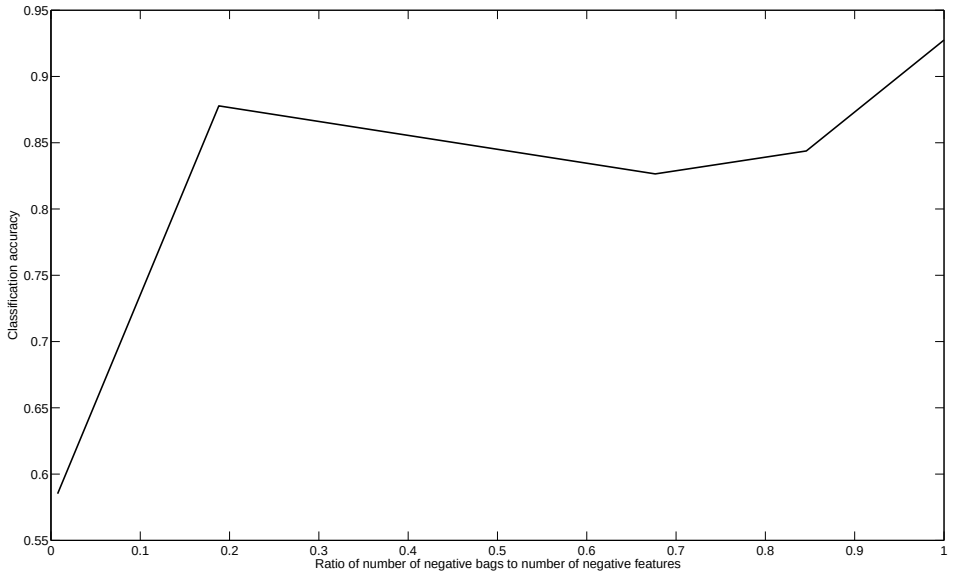


Figure 2: This Figure shows the change in performance of citation kNN as a result of combining and splitting negative bags.

6.5 Randomized Binary Classification

The plot [TODO ref] shows the effect of splitting negative bags on performance accuracy. From the graph, we conclude that splitting all the negative bags into individual bags has the advantage of improving the performance. We justify this by noting that, since we are operating at bag levels, splitting the negative bags effectively increases our training set by giving us more bags for which we know the label (0, since we are certain that all features had 0 label as they belonged to negative bags). While an algorithm working at individual feature level may not have this advantage (since the number of features remains the same), an algorithm working at the bag level has the advantage of seeing an increase in the number of bags.

In all of the kNN based algorithms used in this project, the optimum value for k was found to be 2. The figure below shows the performance of the split citation kNN as a function of k for $k = 1, 2, 3$.

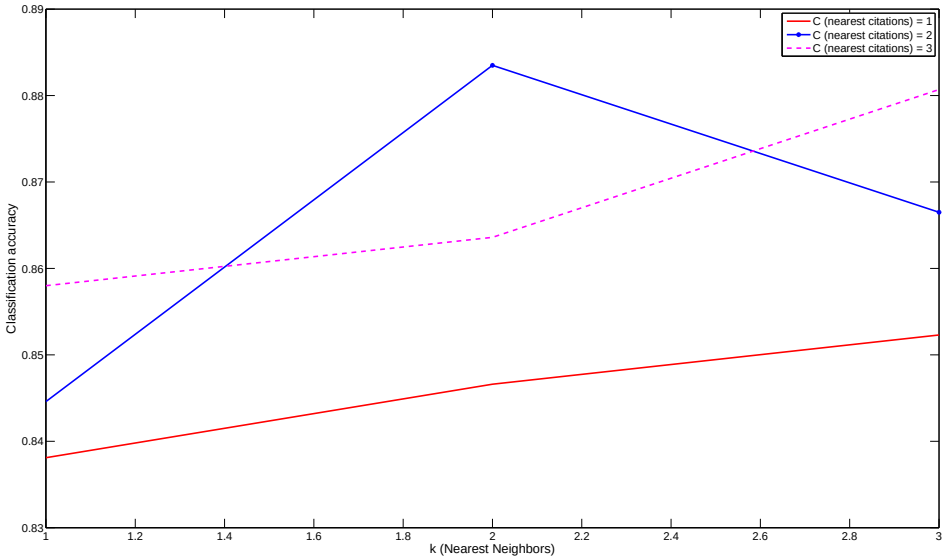


Figure 3: Setting optimum value for the number of references and citations in citation kNN.

The table below lists test error rates for existing and new algorithms performed on Musk-1.

Method	Test Accuracy
APR	91.30%
k Nearest Bags	85.20%
Citation kNN	91.30%
SVM	83.00%
Split kNN	90.23%
Split Citation kNN	92.13%
Randomized kNN	83.26%
Randomized SVM	78.76%
Randomized Split kNN	90.02%

7 Conclusion

We proposed a modification to the existing kNN algorithms for multiple instance learning, and successfully improved the accuracy of classification. We also tried several methods to initialize the SVM algorithm for MIL in a more methodical manner, but we were not successful in improving the performance of SVM.

8 Future Work

As suggestions for future work, we can exploit the high correlation among the elements of the feature vectors. Feature reduction techniques may be helpful for the multiple instance learning problem. Another aspect we would explore is to employ neural networks techniques by considering the individual feature labels as hidden nodes of the network.

9 Description of Individual Efforts

All 3 members of the team contributed to the formulation of the problem and disucussing the methods by which we could improve upon the existing machine learning techinques. Ridvan tried the different approaches to initialize the SVM in a more methodical manner using APR methods and clustering. Aria implemented the modified kNN algorithm, found the optimal k- paramters, and studied the effect of split bags and random initialization on performance. Raj Tejas implemented the citation kNN algorithm for both the original dataset and split bags, studied the optimal parameters to set for the citation kNN algorithm, and the effect of split bags using citation kNN.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. *NIPS*, 2002.

[2] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence Journal*, 1997.

[3] Peter V. Gehler and Olivier Chapelle. Deterministic annealing for multiple-instance learning. *AISTATS*, 2007.

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