An Open Multiple Instance Learning Framework and Its Application in Drug Activity Prediction Problems

Xin Huang[1], Shu-Ching Chen[1] and Mei-Ling Shyu[2]
[1]*Distributed Multimedia Information System Laboratory*
School of Computer Science, Florida International University, Miami, FL 33199
[2]*Department of Electrical and Computer Engineering, University of Miami*
Coral Gables, FL 33124

Abstract

In this paper, a powerful open Multiple Instance Learning (MIL) framework is proposed. Such an open framework is powerful since different sub-methods can be plugged into the framework to generate different specific Multiple Instance Learning algorithms. In our proposed framework, the Multiple Instance Learning problem is first converted to an unconstrained optimization problem by the Minimum Square Error (MSE) criterion, and then the framework can be constructed with an open form of hypothesis and gradient search method. The proposed Multiple Instance Learning framework is applied to the drug activity problems in bioinformatics applications. Specifically, experiments are conducted on the Musk-1 dataset to predict the binding activity of drug molecules. In the experiments, an algorithm with the exponential hypothesis model and the Quasi-Newton method is embedded into our proposed framework. We compare our proposed framework with other existing algorithms and the experimental results show that our proposed framework yields a good accuracy of classification, which demonstrates the feasibility and effectiveness of our framework.

Keyword: Multiple Instance Learning, Bioinformatics, Neural Networks, Machine Learning.

1. Introduction

The Multiple Instance Learning problem is getting more attention recently in the field of machine learning and has been applied in drug discovery. Other applications of Multiple Instance Learning include stock market prediction and natural scene classification. In standard supervised learning, each object in the training examples is labeled and the problem is to learn a hypothesis that can predict the labels of the unseen objects accurately. However, in the scenario of Multiple Instance Learning, the labels of individual instances in the training data are not available; instead the labeled unit is a set of instances (bag). In other words, a training example is a labeled bag. The goal of learning is to obtain a hypothesis from the training examples that generate labels to the unseen bags.

The Multiple Instance Learning technique is originally used in the context of drug activity prediction. In this domain, the input object is a molecule and the observed result is the measurement of the degree to which the molecule binds to a targeted "binding site." A good drug molecule will bind very tightly to the desired binding site, while a poor drug molecule won't [3]. A molecule has a lot of alternative conformations and only one or few of the different conformations of each molecule (bag) are actually bound to the binding site and produce the observed result; while the others typically have no effect on the binding. Unfortunately, the binding activity of a specific molecule conformation can not be directly observed. Actually, only the binding activity of a molecule can be observed. Therefore, the binding activity prediction problem is a multiple instance learning problem. In this sense, each bag is a molecule and the instances of a bag (molecule) are the alternative conformations of the molecule. The label of a bag (molecule) is a measurement of the degree to which the molecule binds to a targeted "binding site." The goal of learning is to predict the degree to which the molecule binds to a targeted "binding site."

In this paper, an open Multiple Instance Learning framework is proposed. The framework is open, which means different sub-methods can be embedded into the framework to generate different specific Multiple Instance Learning algorithms. In our proposed framework, the Multiple Instance Learning problem is first transformed to an unconstrained optimization problem and further converted to the standard supervised learning problem. After those transformations, a learning framework can be constructed. Performance comparison is performed on Musk-1 dataset in the domain of drug activity prediction,

which compares our proposed framework with other existing Multiple Instance Learning algorithms using the accuracy of classification as the performance metrics. The experimental results demonstrate the feasibility and effectiveness of our proposed framework.

This paper is organized as follows. Section 2 briefly introduces the related work in Multiple Instance Learning. Section 3 describes the details of the proposed Multiple Instance Learning framework. The experimental results are presented and analyzed in Section 4. Section 5 gives the conclusion.

2. Related work

Lots of algorithms in Multiple Instance Learning have been proposed in the past few years. Dietterich et al. [3] represented the target concept by an axis-parallel rectangle (APR) in the n-dimensional feature space and presented Multiple Instance Learning algorithms for learning axis-parallel rectangles (APR). In [2], the MULTIINST algorithm for Multiple Instance Learning that is also an APR based method was proposed. In [5], the authors introduced the concept of Diversity Density and applied a two-step gradient ascent with multiple starting points to find the maximum Diversity Density. [7] used the investigated Multiple Instance Regression. Their regression algorithm assumed that each bag has a representative instance and treated it as a missing value and then the EM (Expectation-Maximization) method was used to learn the representative instances and do the regression simultaneously.

Wang et al. [8] explored the lazy learning approaches in Multiple Instance Learning. They developed two kNN-based algorithms: Citation-kNN and Bayesian-kNN. In [9], the authors tried to solve the Multiple Instance Learning problem with decision trees and decision rules. Ramon et al. [6] applied the Neural Network technique on Multiple Instance Learning and proposed the Multiple Instance Neural Network. Andrews et al. [1] utilized the Support Vector Machine in Multiple Instance Learning.

3. The proposed Multiple Instance Learning framework

3.1. Problem definition

In classical Multiple Instance Learning, the label of each bag is either 1 (Positive) or 0 (Negative). A bag is labeled Positive if the bag has one or more Positive instances and is labeled Negative if and only if all its instances are Negative. The Multiple Instance Learning problem is to learn a hypothesis h mapping from a bag to a label (either Positive or Negative). The classical Multiple Instance Learning problem can be defined as follows:

Definition 1. Given the instance space α, the bag space $\beta = 2^{\alpha}$, the label space $\gamma = \{1\,(\text{Positive}), 0\,(\text{Negative})\}$, a set of training examples $T = \langle B, L \rangle$ where $B = \{B_i \mid B_i \in \beta, i = 1 \cdots n\}$ is a set of n bag and $L = \{L_i \mid L_i \in \gamma, i = 1 \cdots n\}$ is the set of their associated labels with L_i being the label of B_i, the problem of Multiple Instance Learning is to generate a hypothesis $h : \beta \to \gamma = \{0.1\}$ which can predict the labels of unknown bags accurately.

In our proposed Multiple Instance Learning framework, the label space is transformed from a discrete space $\gamma = \{1\,(\text{Positive}), 0\,(\text{Negative})\}$ to a continuous space $\gamma' = [0,1]$ and the label of a bag actually indicates the degree to which the bag is Positive, instead of just Positive or Negative. The label "1" means the bag is Positive one hundred percent and the label "0" indicates that the bag is impossible to be Positive. After this transformation, the goal of the learner changes to generate a hypothesis $h_B : \beta \to \gamma' = [0,1]$ from the training examples. Given an unknown bag, the learned hypothesis h_B predicts the degree to which the unknown bag is Positive. When the predicted label is greater than 50%, we can consider that bag is Positive; otherwise the bag is predicted to be Negative.

Actually, each instance in a particular bag has a label in the closed interval $[0,1]$, which represents the extent to which the instance is Positive. Given the labels of all the instances in a bag, the label of the bag can be represented by the maximum of the labels of all its instances. In other words, $L_i = \underset{j}{MAX}\{l_{ij}\}$ where the label L_i is the label of bag B_i and l_{ij} is the label of the j^{th} instance in B_i. Let $h_I : \alpha \to \gamma' = [0,1]$ denote the hypothesis that predicts the label of an instance. The relationship between hypotheses h_B and h_I can be depicted in Figure 1. In Figure 1, each bag B_i has m_i instances and I_{ij} represents the j^{th} instance of B_i. $l_{ij} = h_I(I_{ij})$ is the label of instance I_{ij}. The label of bag B_i is

$$L_i = h_B(B_i) = \underset{j}{MAX}\{l_{ij}\} = \underset{j}{MAX}\{h_I(I_{ij})\} \qquad (1)$$

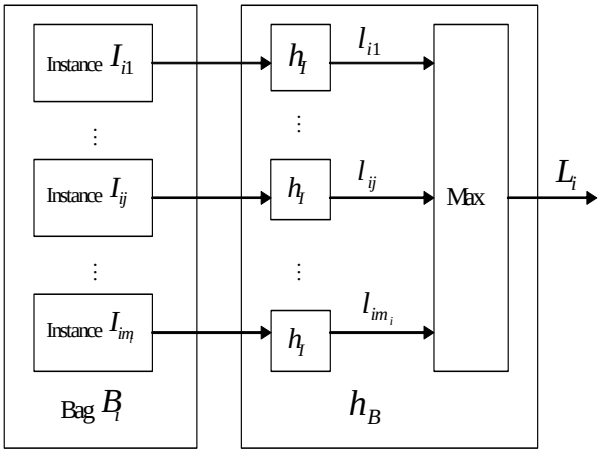

Figure 1. Relationship between h_B and h_I

3.2. Transforming MIL problem to the unconstrained optimization problem

In our proposed Multiple Instance Learning framework, the Multiple Instance Learning problem is transformed into the unconstrained optimization problem using the Minimum Square Error (MSE) criterion. In other words, the goal of the learner is to generate a hypothesis h_B from the given training examples $T = \langle B, L \rangle$ to minimize

$$SE = \sum_{i=1}^{n} \left(L_i - h_B(B_i)\right)^2 = \sum_{i=1}^{n} \left(L_i - \underset{j}{MAX}\left\{h_I\left(I_{ij}\right)\right\}\right)^2 \quad (2)$$

Suppose the hypothesis h_I has M parameters that are $\theta = \{\theta_k\}, (k = 1,2,\cdots,M)$. The Multiple Instance Learning problem in our proposed framework finally is transformed to the following unconstrained optimization problem:

$$\hat{\theta} = \arg \underset{\theta}{Min} \left\{ \sum_{i=1}^{n} \left(L_i - \underset{j}{MAX}\left\{h_I\left(I_{ij}\right)\right\}\right)^2 \right\} \quad (3)$$

One class of the unconstrained optimization methods is the gradient search method such as steepest descent method, Newton method, Quasi-Newton method and Back-propagation (BP) learning method in the Multilayer Feed-Forward Neural Network. To apply those gradient-based methods, the differentiation of the target optimization function needs to be calculated. In our Multiple Instance Learning framework, we need to calculate the differentiation of the target optimization function SE. In order to do that, the differentiation of the *MAX* function needs to be calculated first. The following section discusses how to differentiate the *MAX* function.

3.3. Differentiation of the *MAX* Function

As mentioned in [4], the differentiation of the *MAX* function results in a 'pointer' that specifies the source of the maximum. Let

$$y = MAX(x_1, x_2, \cdots x_n) = \sum_{i=1}^{n} x_i \prod_{j \neq i} U(x_i - x_j) \quad (4)$$

where $U(\cdot)$ is a unit step function, i.e.,

$$U(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

The differentiation of the *MAX* function can be written as:

$$\frac{\partial y}{\partial x} = \prod_{j \neq i} U(x_i - x_j) = \begin{cases} 1 & \text{if } x_i \text{ is a maximum} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

3.4. Differentiation of the Target Optimization Function

Equation (5) provides a way to differentiate the *MAX* function. In order to use the gradient-based search method to solve Equation (3), we need to further calculate the differentiation of the function $E = \left(L_i - \underset{j}{MAX}\left\{h_I\left(I_{ij}\right)\right\}\right)^2$ on the parameters $\theta = \{\theta_k\}$ of hypothesis h_I. The first partial derivative is as follows:

$$\begin{aligned} \frac{\partial E}{\partial \theta_k} &= \frac{\partial \left(L_i - \underset{j}{MAX}\left\{h_I\left(I_{ij}\right)\right\}\right)^2}{\partial \theta_k} \\ &= 2\left(\underset{j}{MAX}\left\{h_I\left(I_{ij}\right)\right\} - L_i \right) \times \frac{\partial \underset{j}{MAX}\left\{h_I\left(I_{ij}\right)\right\}}{\partial \theta_k} \\ &= 2\left(\underset{j}{MAX}\left\{h_I\left(I_{ij}\right)\right\} - L_i \right) \\ &\quad \times \sum_{j=1}^{m_i} \left(\frac{\partial \underset{j}{MAX}\left\{h_I\left(I_{ij}\right)\right\}}{\partial h_I\left(I_{ij}\right)} \times \frac{\partial \left\{h_I\left(I_{ij}\right)\right\}}{\partial \theta_k} \right) \end{aligned} \quad (6)$$

Suppose the s_i^{th} instance of bag B_i has the maximum value, i.e., $h_I(l_{is_i}) = MAX_j\{h_I(l_{ij})\}$. According to Equation (5), Equation (6) can be rewritten as:

$$\begin{aligned}\frac{\partial E}{\partial \theta_k} &= 2\left(h_I\left(I_{is_i}\right) - L_i\right) \\ &\quad \times \sum_{j=1}^{m_i}\left(\frac{\partial MAX_j\left\{h_I\left(I_{ij}\right)\right\}}{\partial h_I\left(l_{ij}\right)} \times \frac{\partial\left\{h_I\left(I_{ij}\right)\right\}}{\partial \theta_k}\right) \\ &= 2\left(h_I\left(I_{is_i}\right) - L_i\right) \times \frac{\partial\left\{h_I\left(I_{is_i}\right)\right\}}{\partial \theta_k} \\ &= \frac{\partial\left(L_i - h_I\left(I_{is_i}\right)\right)^2}{\partial \theta_k}\end{aligned} \tag{7}$$

Furthermore, the z^{th} derivative of the target optimization function E can be written as

$$\begin{aligned}\frac{\partial^z E}{\partial {\theta_k}^z} &= \frac{\partial^z\left(L_i - MAX_j\left\{h_I\left(I_{ij}\right)\right\}\right)^2}{\partial {\theta_k}^z} \\ &= \frac{\partial^z\left(L_i - h_I\left(I_{is_i}\right)\right)^2}{\partial {\theta_k}^z}\end{aligned} \tag{8}$$

and the mixed partial derivation of function E can be written as

$$\begin{aligned}\frac{\partial^{(\sum z_k)} E}{\prod_k \partial {\theta_k}^{z_k}} &= \frac{\partial^{\sum z_k}\left(L_i - MAX_j\left\{h_I\left(I_{ij}\right)\right\}\right)^2}{\prod_k \partial {\theta_k}^{z_k}} \\ &= \frac{\partial^{(\sum z_k)}\left(L_i - h_I\left(I_{is_i}\right)\right)^2}{\prod_k \partial {\theta_k}^{z_k}}\end{aligned} \tag{9}$$

With Equations (7), (8) and (9) to differentiate function E, the target optimization function SE can be easily calculated.

3.5. MIL to Standard Supervised Learning

First let's consider the following standard supervised learning problem. Given the instance space α, the label space $\gamma' = [0,1]$, a set of training examples $T = \langle O, L\rangle$ which includes a set of n objects $O = \{O_i \mid O_i \in \alpha, i = 1\ldots n\}$ and their associated labels $L = \{L_i \mid L_i \in \{0\ (Negative),\ 1\ (Positive)\}, i = 1\ldots n\}$ where L_i is the label of object O_i, the goal of the learner to generate a hypothesis $h_o : \alpha \to \gamma' = [0,1]$ which can predict the labels of unknown objects accurately.

Similar to the analysis on Multiple Instance Learning problem in Section 3.1 and Section 3.2, the traditional supervised learning problem can also be converted to an unconstrained optimization problem using MSE criterion as shown in Equation (10).

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{n}\left(L_i - h_o(O_i)\right)^2 \tag{10}$$

The partial derivative and mixed partial derivative of the function $E_o = \left(L_i - h_o(O_i)\right)^2$ are shown in Equations (11) and (12), respectively.

$$\frac{\partial^z E_o}{\partial {\theta_k}^z} = \frac{\partial^z\left(L_i - h_o(O_i)\right)^2}{\partial {\theta_k}^z} \tag{11}$$

$$\frac{\partial^{(\sum z_k)} E_o}{\prod_k \partial {\theta_k}^{z_k}} = \frac{\partial^{(\sum z_k)}\left(L_i - h_o(O_i)\right)^2}{\prod_k \partial {\theta_k}^{z_k}} \tag{12}$$

Notice that Equation (8) has the same format as Equation (11), and Equation (9) has the same format as Equation (12) except that I_{is_i} in Equations (8) and (9) represent the instances with the maximum label in bag B_i and respectively in Multiple Instance Learning; while O_i in Equations (11) and (12) represent the objects in standard supervised learning. This similarity provides us an easy way to transform Multiple Instance Learning to the traditional supervised learning.

For the Multiple Instance Learning problem with training examples $T = \langle B, L\rangle$ which includes a set n bags $B = \{B_i \mid B_i,\ i = 1...n\}$ and their associated labels $L = \{L_i \mid L_i, i = 1...n\}$ where L_i is the label of bag B_i, the steps of transformation are as follows:

1) For each bag $B_i\ (i = 1, \cdots, n)$ in the training set, calculate the label of each instance I_{ij} belonging to it.

2) Select the instance with the maximum label in each bag B_i. Let I_{is_i} denote the instance with the maximum label in bag B_i.

3) Construct a set of objects $O=\{O_i\}(i=1,\cdots,n)$ using all the instances I_{is}, where $O_i = I_{is_i}$.

4) For each object O_i, construct a label Lo_i that is actually the label of bag B_i. The set of labels is $L_O=\{L_{O_i}\}(i=1,\cdots,n)$.

5) The Multiple Instance Learning problem with the set of training examples $T=\langle B,L\rangle$ is converted to the traditional supervised learning problem with the set of training examples $T'=\langle O,L_O\rangle$.

After this transformation, the gradient-based search methods used in the standard supervise learning including the steepest descent method, Newton method, etc. can be applied to Multiple Instance Learning directly.

Despite the above transformation from Multiple Instance Learning to the standard supervised learning, there still exists a major difference between Multiple Instance Learning and standard supervised learning. In the standard supervised learning, the training examples are static and usually do not change during the learning procedure. However, in the transformed version of Multiple Instance Learning, the training examples may change during the learning procedure. The reason is that the instance with the maximum label in each bag may change with the update of the approximated hypothesis h_I during the learning procedure and therefore the training examples constructed along with the aforementioned transformation may change during the learning procedure. In spite of such a dynamic characteristic of the training examples, the fundamental learning method remains the same. Table 1 gives our proposed Multiple Instance Learning algorithm.

Once the parameters θ are learned, the hypothesis h_I is determined and thus the hypothesis h_B can be generated by Equation (1). The labels of the unknown bags can be generated by the hypothesis h_B. Obviously, the convergence of our Multiple Instance Learning framework depends on what kind of gradient-based search method is applied at Step 4. Actually, it has the same convergence property as when the gradient-based search method is applied.

MIL (B,L)

Input: $B=\{B_i\}(i=1,\cdots,n)$ is the set of bags in the training set and $B_i=\{I_{ij}\}(j=1,\cdots,m_i)$ where I_{ij} is the j^{th} instance of bag B_i and m_i is the number of instances in B_i.

$L=\{L_i\}(i=1,\cdots,n)$ is the set of labels where L_i is the label of bag B_i.

Output: $\theta=\{\theta_k\}(k=1,\cdots M)$ is the set of parameters of hypothesis h_I to be learned where M is the number of parameters.

Procedure:

1. Set the initial values to parameters θ_k in θ.

2. If the stop criterion has not been met, go to step 3; else return the parameter set θ of hypothesis h_I.
   / The stop criterion can be based on error or the number of iterations. */*

3. Transform Multiple Instance Learning to traditional supervised learning using the method described in this section. The result of this transformation is a training set $T'=\langle O,Lo\rangle$ where $O=\{O_i\}(i=1,\cdots,n)$ is a set of objects and $Lo=\{Lo_i\}(i=1,\cdots,n)$ is a set of corresponding labels of object set O.
   / In each iteration, the training example T' may be different since the parameter set γ is updated during the iteration, and the instance with the maximum label in each bag may also change. */*

4. Apply the gradient-based search method in traditional supervised learning to update the parameters in θ.
   / The search method applied is not restricted to a particular one. Instead, it is optional as long as it is a gradient-based search method such as the steepest descent method, Newton method, and BP algorithm in the Multilayer Feed-Forward Neural Network */*

5. Go to Step 2.

Table 1: The proposed MIL algorithm

4. Experiments and results

In our proposed Multiple Instance framework, the form of a hypothesis to be learned and the gradient search method are open, which means many forms of the hypotheses and search methods can be plugged into the framework. In other words, such an open framework is more powerful since one can select any hypothesis and search method suitable for a particular application. In the conducted experiments, an algorithm that embeds the exponential hypothesis model and the Quasi-Newton method into our proposed framework is used. This algorithm assumes that there is a target concept point in

the feature space of the instances. The probability of a point (instance) being Positive is the exponential function of its distance to the target concept point [5]. The Quasi-Newton method is used to search the target concept point and the optimum parameters.

Experiments are conducted on the Musk-1 dataset from the UCI machine learning repository (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/musk/), which has been used as the benchmark for evaluating Multiple Instance Learning algorithms. The experimental results show that our proposed framework achieves the same accuracy as that of algorithm Iterated-Discrim APR and higher accuracy than the other existing algorithms.

4.1. Experiment Setup

The exponential hypothesis model and the Quasi-Newton method are used in the algorithm for our proposed Multiple Instance Learning framework. Assume the dimension of the feature space of the instances is n and thus each instance corresponds to a point in the n-dimensional feature space. This algorithm models the probability of an instance I_i being Positive as, $p(I_i) = h_I(I_i) = \exp\left(\sum_{j=1}^{n} \omega_j (f_{ij} - t_j)^2\right)$ where f_{ij} is the j^{th} feature of instance I_i, $t = (t_1, t_2, \cdots, t_j, \cdots, t_n)$ is the target concept point in the feature space which is Positive with 100% probability, and $\omega = (\omega_1, \omega_2, \cdots, \omega_j, \cdots, \omega_n)$ are the scale factors indicating the importance of the different dimensions. In other words, this algorithm assumes the probability of an instance being Positive is exponentially proportional to its distance to the target concept point t. In addition, this algorithm uses the Quasi-Newton method as the gradient search method. The parameters θ of the hypothesis model are the target concept point $t = (t_1, t_2, \cdots, t_j, \cdots, t_n)$ and the scale factors $\omega = (\omega_1, \omega_2, \cdots, \omega_j, \cdots, \omega_n)$. The dataset, the initial values for the target concept point and the scale factors in the exponential hypothesis model, and the termination condition of Quasi-Newton method used in the experiments are set up as follows.

- **Setup of the dataset.**

The MUSK- I dataset is used in the experiments. The MUSK- I 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. Each molecule is a bag and has a number of instances that is an alternative conformation of a molecule. Each instance has 166 features. The goal is to learn to predict whether new molecules will be musks or non-musks.

- **Setup of the initial values of t.**

First, the K-Nearest Neighbor algorithm is used to cluster the instances in the training examples. Then the cluster with the greatest ratio of $\frac{\#\ of\ \ Positive\ Ins\tan ces}{\#\ of\ \ Negative\ Ins\tan ces}$ is selected. The centroid of all the Positive instances in the selected cluster is calculated, and the initial target concept point t is set as this centroid.

- **Setup of the initial values of ω.**

Each ω_j is initialized with a random value between 0 and 1.

- **Setup of the termination condition of the Quasi-Newton method.**

The termination condition is set to be $\left|E^{(k)} - E^{(k-1)}\right| < \eta \times E^{(k-1)}$, where $E^{(k)}$ denotes the value of the optimization target function E at the k^{th} iteration and η is a small constant. In our experiments, η is set to 0.005.

4.2. Experimental Results and Analysis

We compare the performance of our algorithm with other existing algorithms such as five APR-based algorithms [3], Diversity density [5] and Multiple Instance Neural Network [6] using the Musk-1 dataset, and the 10-fold cross-validation is used to estimate the accuracy of classification. Table 2 shows the accuracy of the various algorithms, where the accuracy is the average accuracy across 10 runs using 10-fold cross validation.

Algorithm	Accuracy (%)
Iterated-Discrim APR[3]	92.4
GFS elim-kde APR [3]	91.3
GFS elim-count APR [3]	90.2
GFS all-positive APR [3]	83.7
All-positive APR [3]	80.4
Multiple Instance Neural Network [6]	88.0
Diversity density [5]	88.9
Our algorithm	92.4

Table 2: Performance comparison on the Musk-1 dataset

As can be seen from Table 2, our algorithm achieves 92.4% accuracy of classification and outperforms all the algorithms listed in Table 2 except the Iterated-Discrim APR algorithm which also obtained the 92.4% accuracy.

The most advantage of our work is that it actually proposes an open framework of multiple instance learning. The form of the learned hypothesis and the specific optimization method are not fixed. Many forms of the hypotheses and optimization methods can be plugged into the framework. Compared with the other algorithms in the literature, where the form of hypothesis and learning method are fixed, our framework is flexible and more powerful. The reason is that for different specific applications, different kinds of hypotheses and learning methods may produce the best results. In fact, it is not possible to find one approach that can generally best suit all the applications. For example, we tried to use a three-layer feedforward neural network as the hypothesis and the back-propagation algorithm as the leaning method, and plugged them into our multiple instance learning network, which achieved 91.3% accuracy of classification on Musk-1 dataset. This is worse than the result we discussed previously by applying the exponential hypothesis model and the Quasi-Newton method into the framework. This means the latter are more suitable for dataset Musk-1. On the other hand, we also made experiments on an artificial dataset using those two algorithms in our framework and we found out that the three-layer feedforward neural network and back-propagation algorithm performed better on that dataset. Hence, our proposed Multiple Instance Learning framework provides the capability of selecting different forms of hypothesis and learning algorithm for different applications. Therefore, it is more flexible and powerful than other multiple instance learning algorithms in the literature.

5. Conclusions

Different with other work in Multiple Instance Learning, this paper presented an open Multiple Instance Learning framework. Our proposed framework is more powerful since different forms of hypotheses and gradient search methods for optimization can be plugged into the framework easily to generate a specific Multiple Instance leaning algorithm. The proposed learning framework converts the Multiple Instance Learning problem to an unconstrained optimization problem by the Minimum Square Error (MSE) criterion. Experiments were conducted to compare the accuracy of classification on the Musk-1 dataset for the bioinformatics application. In the experiments, an algorithm that embeds the exponential hypothesis model and the Quasi-Newton method into our proposed Multiple Instance Learning framework was used. The experimental results justify the feasibility and demonstrate good performance in term of the prediction accuracy.

6. Acknowledgements

This work is was supported in part by NSF CDA-9711582 and NSF EIA-0220562

7. References

[1] S. Andrews, T. Hofmann, and I. Tsochantaridis, "Multiple Instance Learning with Generalized Support Vector Machines," *Proc. of the 18th National Conference on Artificial Intelligence and 14th Conference on Innovative Applications of Artificial Intelligence*, AAAI Press, Edmonton, Alberta, Canada, 2002, pp. 943-944.

[2] P. Auer, "On Learning From Multi-instance Examples: Empirical Evaluation of a Theoretical Approach," *Proc. of the 14th International Conference on Machine Learning*, Morgan Kaufman, San Francisco, CA, 1997, pp. 21-29..

[3] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Perez, "Solving the Multiple-Instance Problem with Axis-Parallel Rectangles," *Aritificial Intelligence Journal*, 89, 1997, pp. 31-71.

[4] R.J. Marks II, S. Oh, P. Arabshahi, T.P. Caudell, J.J. Choi, and B.G. Song, "Steepest Descent Adaptation of Min-Max Fuzzy If-Then Rules," *Proc. of IEEE/INNS International Conference on Neural Networks*, Beijing, China, 1992. pp. 471-477.

[5] O. Maron and T. Lozano-Pérez, "A framework for Multiple-Instance learning," *In advances in Neural Information Processing Systems*, 10, MIT press, 1998.

[6] J. Ramon and L. De Raedt, "Multi-Instance Neural Networks," *Proc. of the ICML 2000 Workshop on Attribute-value and Relational Learning*, 2000.

[7] S. Ray and D. Page, "Multiple-Instance Regression," *Proc. of the 18th International Conference on Machine Learning*, Morgan Kaufman, San Francisco, CA, 2001, pp. 425-432.

[8] J. Wang and J.-D. Zucker, "Solving the Multiple-Instance Learning Problem: A Lazy Learning Approach," *Proc. of the 17th International Conference on Machine Learning*, Morgan Kaufman, San Francisco, CA, 2000, pp. 1119-1125.

[9] J.-D. Zucker and Y. Chevaleyre, "Solving Multiple-instance and Multiple-part Learning Problems with Decision Trees and Decision Rules," Application to the Mutagenesis Problem. *In Technical Report, LIP6*, Univ. of Paris 6, 2000.