Neural Processing Letters **19:** 1–10, 2004.

Improve Multi-Instance Neural Networks through Feature Selection

MIN-LING ZHANG and ZHI-HUA ZHOU*
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China. Tel.: +86-25-359-3163; Fax: +86-25-330-0710; e-mail: zml@ai.nju.edu.cn, zhouzh@nju.edu.cn

Abstract. Multi-instance learning is regarded as a new learning framework where the training examples are *bags* composed of instances without labels, and the task is to predict the labels of unseen bags through analyzing the training bags with known labels. Recently, a multi-instance neural network BP-MIP was proposed. In this paper, BP-MIP is improved through adopting two different feature selection techniques, i.e. feature scaling with Diverse Density and feature reduction with principal component analysis. In detail, before feature vectors are fed to a BP-MIP neural network, they are scaled by the feature weights found by running Diverse Density on the training data, or projected by a linear transformation matrix formed by principal component analysis. Experiments show that these feature selection mechanisms can significantly improve the performance of BP-MIP.

Key words. backpropagation, feature selection, machine learning, multi-instance learning, neural networks

Abbreviations. DD – Diverse density; PCA – Principal component analysis

1. Introduction

In the middle of 1990's, Dietterich et al. [7] investigated the problem of drug activity prediction. The goal is to endow learning systems with the ability of predicting that whether a new molecule could be used to make some drug, through analyzing a collection of known molecules. The qualification of the molecule to make some drug is determined by some of its shapes with low energy. However, at present biochemists only know that whether a known molecule is qualified to make some drug instead of knowing that which of its alternative shapes is responsible for the qualification. In order to solve this problem, Dietterich et al. [7] initialized the notion of *multi-instance learning*.

In multi-instance learning, the training set is composed of many *bags* each containing many instances. The bags are labeled in the way that if a bag contains at least one positive instance then it is labeled as a positive bag. Otherwise it is labeled as a negative bag. The task is to learn some concept from the training bags for correctly labeling unseen bags. The difficulty of multi-instance learning lies in that unlike standard supervised learning where all the training instances are labeled, the labels of the individual instances are unknown in multi-instance learning.

*Corresponding author.

Dietterich et al. [7] showed that learning methods ignoring the characteristics of multi-instance problem could not work well in this scenario. Due to its unique characteristics and extensive applicability, multi-instance learning has been regarded as a new learning framework different to *supervised learning*, *unsupervised learning*, and *reinforcement learning* [11].

When the notion of multi-instance learning was coined, Dietterich et al. [7] indicated that a particular interesting issue in this area is to design multi-instance modifications for neural networks. Recently, this open problem has been addressed by a multi-instance neural network named BP-MIP [21], which extended the popular BP [17] algorithm with a global error function defined at the level of bags instead of at the level of instances. Experiments [21] show that the performance of BP-MIP is comparable to many existing multi-instance learning algorithms. However, it is not so good as several algorithms with inbuilt feature selection mechanisms, such as iterated-discrim APR [7] and Diverse Density [12]. Therefore, an interesting issue to be explored is that whether the performance of BP-MIP could be significantly improved with the help of feature selection. In this paper, two variants of BP-MIP, i.e. BP-MIP-DD incorporating feature scaling with Diverse Density and BP-MIP-PCA incorporating feature reduction with principal component analysis (PCA) [9], are presented. Experiments on the drug activity prediction data, which is the only real-world benchmark test data for multi-instance learning at present, show that both methods significantly improve the performance of BP-MIP.

The rest of this paper is organized as follows. Section 2 reviews previous works on multi-instance learning. Section 3 briefly introduces BP-MIP and then presents BP-MIP-DD and BP-MIP-PCA. Section 4 reports experimental results on the drug activity prediction data. Finally, Section 5 concludes and indicates several issues for future work.

2. Previous Works Review

Long and Tan [10] described a high-order polynomial-time theoretical algorithm and show that if the instances in the bags are independently drawn from product distribution, then the APR is PAC-learnable. Auer et al. [3] showed that if the instances in the bags are not independent then APR learning under the multi-instance learning framework is NP-hard. Moreover, they also presented a theoretical algorithm without requiring product distribution and with reduced time and sample complexity than that of Long and Tan's algorithm. Later, this theoretical algorithm was transformed into a practical algorithm named MULTINST [2]. Blum and Kalai [5] gave a reduction from the problem of PAC-learning under the multi-instance learning framework to PAC-learning with one-sided or two-sided random classification noise, and presented a theoretical algorithm with smaller sample complexity than that of Auer et al.'s algorithm. Unfortunately, all the above theoretical analyses made the restrictive assumption that each bag contains the *same* number of *independent* instances, which is usually not the case in practice.

A representative practical multi-instance learning algorithm is Diverse Density proposed by Maron and Lozano-Pérez [12], which has been applied to several applications including learning a simple description of a person from a series of images [12], stock prediction [12], natural scene classification [13], and content-based image retrieval [19]. There are also many other practical algorithms. Wang and Zucker [18] extended k-nearest neighbor algorithm for multi-instance learning through adopting Hausdorff distance, and provided two multi-instance learning algorithm called Bayesian-kNN and Citation-kNN. Ruffo [16] presented a multi-instance version of C4.5 decision tree named Relic and applied it to data mining. Chevaleyre and Zucker [6] derived ID3-MI and RIPPER-MI, which are multi-instance version of decision tree algorithm ID3 and rule learning algorithm RIPPER. Zhou and Zhang [21] developed a multi-instance neural network named BP-MIP through employing a new error function capturing the nature of multi-instance learning. Zhang and Goldman [20] proposed EM-DD, which combines the EM and Diverse Density algorithms. Zhou and Zhang [22] applied ensemble learning paradigms to multi-instance learning and obtained the best result up to now on a benchmark test.

Recently, some researchers begin to investigate multi-instance regression tasks with real-valued outputs. Ray and Page [15] showed that the general formulation of multi-instance regression is NP-hard, and proposed an EM-based multi-instance regression algorithm. Dooly et al. [8] further proved that learning from real-valued multi-instance examples is as hard as learning DNF. Amar et al. [1] extended Diverse Density and Citation-kNN for multi-instance regression and designed a method for artificially generating data sets for multi-instance regression.

3. Feature Selection for BP-MIP

Suppose the training set is composed of N bags, i.e. $\{B_1, B_2, \ldots, B_N\}$ the ith bag contains M_i instances, i.e. $\{B_{i1}, B_{i2}, \ldots, B_{iM_i}\}$, each instance is a p-dimensional feature vector, e.g. the jth instance of the ith bag is $[B_{ij1}, B_{ijv}, \ldots, B_{ijp}]^T$. The training set is further divided into n positive bags $B_1^+, B_2^+, \ldots, B_n^+$ and m negative bags $B_1^-, B_2^-, \ldots, B_m^-$ where $n+m=N$. The desired output of a positive training bag is 1, while that of a negative training bag is 0.

3.1. BP-MIP

Suppose a neural network with p input units and one output unit is used to learn from the training set. Since the goal of multi-instance learning is to predict the labels of unseen bags, the global error of the network on the training set is defined as:

$$E = \sum_{i=1}^{N} E_i = \sum_{i=1}^{N} \frac{1}{2}\left(\max_{1 \leqslant j \leqslant M_i} o_{ij} - d_i \right)^2 \tag{1}$$

where E_i is the error of the network on B_i, o_{ij} is the actual output of the network on B_{ij} and d_i is the desired output of B_i.

Note that the error function shown as Equation (1) is defined at the level of training bags instead of at the level of training instances. Thus, the nature of multi-instance learning, i.e. a positive bag contains at least one positive instance while a negative bag contains no positive instances, is appropriately addressed.

With the defined error function, the BP algorithm [17] is easy to be updated for multi-instance learning. In detail, in each training epoch of BP-MIP, the training bags are fed to the network one by one. For bag B_i, if it has been correctly predicted then no weight in the network is changed. Otherwise the weights are modified according to the error on the instance whose corresponding actual output is the maximal in B_i. After that, B_{i+1} is fed to the network and the training process is iterated until the global error E decreases to a threshold or the number of training epochs increases to a threshold.

It is worth noting that although the predictive accuracy of BP-MIP on the drug activity prediction data is better than many multi-instance learning algorithms, it is not so good as several algorithms such as iterated-discrim APR [7] and Diverse Density [12] that have inbuilt feature selection mechanisms. Therefore an intuitive way to boost the performance of BP-MIP is to resort to feature selection techniques.

3.2. BP-MIP-DD

Diverse Density [12] is a famous multi-instance learning algorithm. The diverse density of a point in the feature space is the measure of how many different positive bags have instances near that point and how far the negative instances are from that point. Strictly speaking, for any point $\boldsymbol{t}$ in the feature space, the probability of $\boldsymbol{t}$ being the target point (i.e. the diverse density of $\boldsymbol{t}$), given all the positive and negative bags, is $\Pr(\boldsymbol{t} \mid B_1^+, \ldots, B_n^+, B_1^-, \ldots, B_m^-)$. So the target point that maximizes this probability is

$$\underset{\boldsymbol{t}}{\operatorname{argmax}} \Pr(\boldsymbol{t} \mid B_1^+, \ldots, B_n^+, B_1^-, \ldots, B_m^-) \tag{2}$$

Using Bayes' rule, assuming an uninformative prior over the concept location $\Pr(\boldsymbol{t})$ and conditional independence of the bags given the target concept $\boldsymbol{t}$, the above equals

$$\underset{\boldsymbol{t}}{\operatorname{argmax}} \prod_i \Pr(\boldsymbol{t} \mid B_i^+) \prod_i \Pr(\boldsymbol{t} \mid B_i^-) \tag{3}$$

This is a formal definition of maximizing Diverse Density. Maron and Lozano-Pérez [12] used the 'noisy-or' model [14] such that

$$\Pr(\boldsymbol{t} \mid B_i^+) = 1 - \prod_j (1 - \Pr(B_{ij}^+ = \boldsymbol{t})) \tag{4}$$

$$\Pr(\boldsymbol{t} \mid B_i^-) = \prod_j (1 - \Pr(B_{ij}^- = \boldsymbol{t})) \tag{5}$$

and made the following assumption:

$$\Pr(B_{ij} = \boldsymbol{t}) = \exp(-\|B_{ij} - \boldsymbol{t}\|^2) \tag{6}$$

where $\|B_{ij} - \boldsymbol{t}\|$ is the distance between the two vectors. It is worth mentioning that not all dimensions are equally important, so they defined the distance to be a weighted Euclidean distance:

$$\|B_{ij} - \boldsymbol{t}\|^2 = \sum_{k=1}^{p} w_k^2 (B_{ijk} - \boldsymbol{t}_k)^2 \tag{7}$$

where B_{ijk} is the kth dimension in the vector B_{ij} and $|w_k|$ is the corresponding non-negative weight ($|x|$ denotes the absolute value of x). Multiple gradient ascents (starting from every instance in all positive bags) are performed to search for the point with maximal diverse density as well as the best feature weights corresponding to that point.

BP-MIP-DD utilizes the weights found by Diverse Density to improve the performance of BP-MIP. In detail, before feature vectors are fed to a BP-MIP neural network for training or testing, they are scaled by the weights found by running Diverse Density on the training data. In fact, the intention of BP-MIP-DD is to rescale the feature space to stress important features with larger weights and suppress insignificant features with smaller weights.

3.3. BP-MIP-PCA

Experiments on the drug activity prediction data show that BP-MIP-DD achieves better results through feature scaling (as reported in Section 4), which indicates that some features may have little responsibility in characterizing the original data. Therefore, eliminating these redundant features while remaining the relevant ones may probably lead to better performance.

PCA [9] is the most popular method for the reduction of irrelevant features, which is usually employed to discover the intrinsic dimensionality of a data set based on the eigenvalues of a covariance matrix $\mathbf{R}$ computed from the data. The M eigenvectors corresponding to the M largest eigenvalues of $\mathbf{R}$ define a linear transformation matrix $\mathbf{T}$, which projects the N-dimensional space into an M-dimensional space in which the features are uncorrelated. Note that M, i.e. the dimensionality of the transformed data, is generally much smaller than N, i.e. the dimensionality of the original data.

BP-MIP-PCA incorporates PCA into BP-MIP to perform feature reduction. Specifically, feature vectors are projected by the linear transformation matrix $\mathbf{T}$ before they are fed to a BP-MIP neural network. Experiments on the drug activity prediction data show that BP-MIP-PCA can significantly increase the performance of BP-MIP (as reported in Section 4). This observation supports Dietterich et al.'s [7] claim that the number of relevant features of the drug activity prediction data will probably be substantially less than 166.

4. Experiments

4.1. EXPERIMENTAL SETUP

The *Musk* data is the only real-world benchmark test data for multi-instance learning at present. There are two data sets, both of which are publicly available

Table I. Some characteristics of the *Musk* data.

Data set	*Musk1*	*Musk2*
Dimensionality	166	166
Number of bags	92	102
Number of positive bags	47	39
Number of negative bags	45	63
Number of instances	476	6,598
Average number of instances per bag	5.17	64.69
Maximal number of instances in a bag	40	1,044
Minimal number of instances in a bag	2	1

from UCI Machine Learning Repository [4]. Characteristics of those two data sets are summarized in Table I.

Leave-one-out test is performed on the *Musk* data. In detail, for *N* bags, one bag is used to test while the others are used to train a neural network in a loop of *N* iterations. In each iteration, BP-MIP, BP-MIP-DD, and BP-MIP-PCA are trained according to the rules described in Section 3 respectively. The iterations are repeated in the way that each bag in the data set has been used as the test bag once. At the end of the loop, the final predictive accuracy is calculated as the total number of correctly labeled test bags divided by *N*.

Since the computational cost of Diverse Density is even higher than that of training a BP-MIP neural network, in the experiments Diverse Density is run only once on the whole data set instead of in each iteration for BP-MIP-DD. For fair comparison, PCA is also run only once so that the linear transformation matrix **T** used in each iterations of BP-MIP-PCA is the same, i.e. **T** is derived from all the instances in all *N* bags.

Feedforward neural networks with one output unit, one hidden layer with 80 units, and 166 input units each corresponds to a dimension of the 166-dimensional feature vectors, are trained with the BP-MIP and BP-MIP-DD algorithms. The learning rate is set to 0.05, while the number of training epochs varies from 50 to 1,000 with an interval of 50.

Configurations of BP-MIP-PCA are the same as that of BP-MIP and BP-MIP-DD except the number of input units and the number of hidden units. In order to determine the number of input units, i.e. the number of remained features after feature reduction, PCA is incorporated into another multi-instance learning algorithm named Citation-*k*NN [18]. Figure 1 and Figure 2 show the predictive accuracy of Citation-*k*NN on *Musk1* and *Musk2* with leave-one-out test respectively, where the horizontal axis indicates the number of remained features.

Figure 1 shows that Citation-*k*NN achieves the highest predictive accuracy 93.48% on *Musk1* when 41 or 42 features are remained. This indicates the intrinsic dimensionality of the *Musk1* data set may be around 40. As for *Musk2*, Figure 2 shows that Citation-*k*NN reaches the highest predictive accuracy 86.27% when

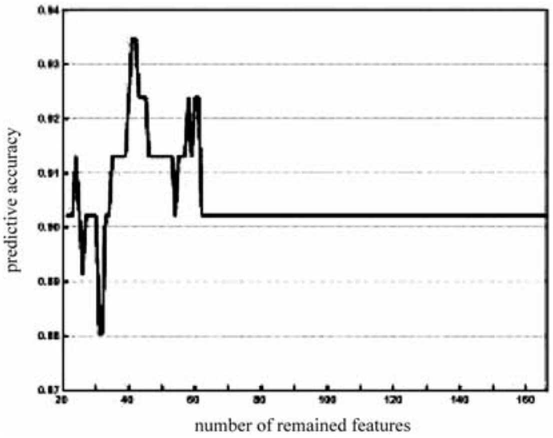

Figure 1. Predictive accuracy of Citation-*k*NN combined with PCA on the *Musk1* data set.

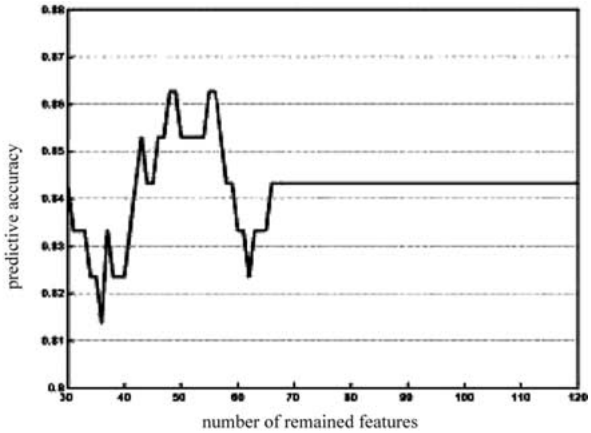

Figure 2. Predictive accuracy of Citation-*k*NN combined with PCA on the *Musk2* data set.

remaining 48, 49, 55 or 56 features. According to the above facts, the number of input units and the number of hidden units of BP-MIP-PCA are set to 41 and 40 respectively for *Musk1*, while 48 and 40 respectively for *Musk2*.

4.2. RESULTS

The predictive accuracy curves of BP-MIP, BP-MIP-DD, and BP-MIP-PCA on *Musk1* and *Musk2* are shown in Figure 3 and Figure 4 respectively. The horizontal axis indicates the number of training epochs.

For *Musk1* (as shown in Figure 3), the best performance of BP-MIP-PCA is 88.0%, which is better than 85.9% and 83.7%, i.e. the best performance achieved by BP-MIP-DD and BP-MIP respectively. Furthermore, BP-MIP-PCA and

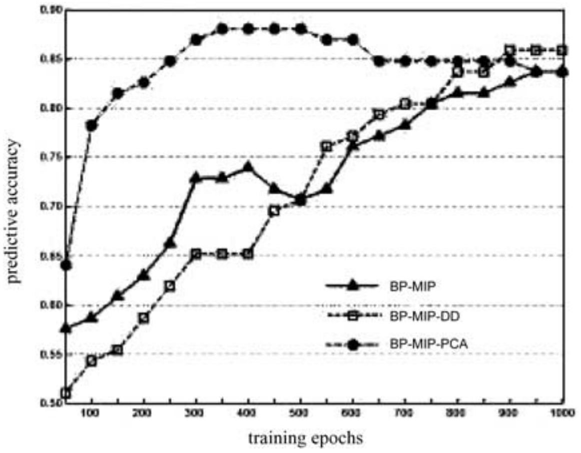

Figure 3. Predictive accuracy on *Musk1.*

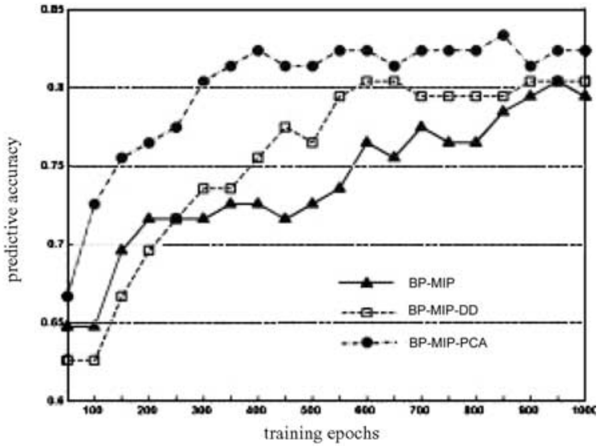

Figure 4. Predictive accuracy on *Musk2.*

BP-MIP-DD significantly and persistently perform better than BP-MIP after 50 and 500 training epochs correspondingly.

For *Musk1*, on the other hand, BP-MIP-PCA is better than BP-MIP-DD until the number of training epochs increases to 900. It is expected that if BP-MIP-DD is trained with more epochs, its predictive accuracy can be further improved. In addition, although the architecture and parameters of BP-MIP-PCA have not been finely tuned, its best performance on *Musk1*, i.e. 88.0% between training epochs 350 and 500, is comparable to 88.9%, i.e. the result achieved by Diverse Density [12].

For *Musk2* (as shown in Figure 4), as what has been exhibited on *Musk1*, the best performance of BP-MIP-PCA, i.e. 83.33%, is also better than 80.39%, i.e. the best performance identically achieved by BP-MIP-DD and BP-MIP. Furthermore, BP-MIP-PCA and BP-MIP-DD significantly and persistently perform better than BP-MIP after 50 and 250 training epochs correspondingly.

For *Musk2*, on the other hand, BP-MIP-DD reaches the highest predictive accuracy 80.39% after 600 training epochs, while BP-MIP rarely reach the same highest predictive accuracy 350 training epochs later. In addition, the best performance of BP-MIP-PCA, i.e. 83.33% on training epochs 850, is better than 82.50%, i.e. the result achieved by Diverse Density [12], even though the architecture and parameters of BP-MIP-PCA have not been finely tuned.

5. Conclusion

BP-MIP is a multi-instance neural network derived from the popular BP algorithm through employing a new error function capturing the nature of multi-instance learning. In this paper, two variants of BP-MIP, i.e. BP-MIP-DD and BP-MIP-PCA are developed. BP-MIP-DD scales the features with Diverse Density; and BP-MIP-PCA eliminates features with principal component analysis. Experiments on the drug activity prediction data show that the performance of BP-MIP can be significantly improved with both methods. This indicates that incorporating feature selection mechanisms may be a promising way to boost current multi-instance learning algorithms.

It is worth noting that the experimental results reported in this paper is rather preliminary because due to the time limitation, the architecture and parameters of the neural networks have not been finely tuned. It is obvious that investigating better configurations of the neural networks is an important issue to be explored in the near future. Furthermore, investigating other feature selection methods to further improve BP-MIP is another interesting issue for future work.

Acknowledgements

The comments and suggestions from the anonymous reviewers greatly improved this paper. This work was supported by the National Outstanding Youth Foundation of China under the Grant No. 60325207, the National Natural Science Foundation of China under the Grant No. 60105004, and the Natural Science Foundation of Jiangsu Province under the Grant No. BK2001406.

References

1. Amar, R. A., Dooly, D. R., Goldman, D. R. and Zhang, Q.: Multiple-instance learning of real-valued data, In: *Proceedings of the 18th International Conference on Machine Learning*, pp. 3–10, Williamstown, MA, 2001.
2. Auer, P.: On learning from multi-instance examples: empirical evaluation of a theoretical approach, In: *Proceedings of the 14th International Conference on Machine Learning*, pp. 21–29, Nashville, TN, 1997.
3. Auer, P., Long, P. M. and Srinivasan, A.: Approximating hyper-rectangles learning and pseudo-random sets, *J. Comput. Syst. Sci.* **57**(3) (1998), 376–388.

4. Blake, C., Keogh, E. and Merz, C. J.: UCI repository of machine learning databases, Department of Information and Computer Science, University of California, Irvine, CA, 1998. [http://www.ics.uci.edu/~mlearn/MLRepository.html]
5. Blum, A. and Kalai, A.: A note on learning from multiple-instance examples, *Mach. Learn.* **30**(1) (1998), 23–29.
6. Chevaleyre, Y. and Zucker, J.-D.: Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. Application to the mutagenesis problem, In: E. Stroulia and S. Matwin (eds), *Lecture Notes in Artificial Intelligence*, *Vol. 2056*, Berlin, Springer, pp. 204–214, 2001.
7. Dietterich, T. G., Lathrop, R. H. and Lozano-Pérez, T.: Solving the multiple-instance problem with axis-parallel rectangles, *Artif. Intell.* **89**(1–2) (1997), 31–71.
8. Dooly, D. R., Goldman, S. A. and Kwek, S. S.: Real-valued multiple-instance learning with queries, In: N. Abe, R. Khardon and R. Zeugmann (eds), *Lecture Notes in Artificial Intelligence*, *Vol. 2225*, Berlin, Springer, pp. 167–180, 2001.
9. Jollife, I. T.: *Principal Component Analysis*, Springer-Verlag, New York, 1986.
10. Long, P. M. and Tan, L.: PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples, *Mach. Learn.* **30**(1) (1998), 7–21.
11. Maron, O.: Learning from Ambiguity. Ph D thesis, Department of Electrical Engineering and Computer Science, MIT, June 1998.
12. Maron, O. and Lozano-Pérez, T.: A framework for multiple-instance learning, In: M. I. Jordan and M. J. Kearns (eds), *Advances in Neural Information Processing Systems 10*, Cambridge, MA, MIT Press, pp. 570–576, 1998.
13. Maron, O. and Ratan, A. L.: Multiple-instance learning for natural scene classification, In: *Proceedings of the 15th International Conference on Machine Learning*, pp. 341–349, Madison, WI, 1998.
14. Pearl, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA, 1988.
15. Ray, S. and Page, D.: Multiple instance regression, In: *Proceedings of the 18th International Conference on Machine Learning*, pp. 425–432, Williamstown, MA, 2001.
16. Ruffo, G.: Learning Single and Multiple Instance Decision Tree for Computer Security Applications. Ph D thesis, Department of Computer Science, University of Turin, Torino, Italy, 2000.
17. Rumelhart, D. E., Hinton, G. E. and Williams, R. J.: Learning internal representations by error propagation, In: D. E. Rumelhart and J. L. McClelland (eds), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, *Vol. 1*, pp. 318–362, Cambridge, MA, MIT Press, 1986.
18. Wang, J. and Zucker, J.-D.: Solving the multiple-instance problem: a lazy learning approach, In: *Proceedings of the 17th International Conference on Machine Learning*, pp. 1119–1125, San Francisco, CA, 2000.
19. Yang, C. and Lozano-Pérez, T.: Image database retrieval with multiple-instance learning techniques, In: *Proceedings of the 16th International Conference on Data Engineering*, pp. 233–243, San Diego, CA, 2000.
20. Zhang, Q. and Goldman, S. A.: EM-DD: an improved multi-instance learning technique, In: T. G. Dietterich, S. Becker and Z. Ghahramani (eds), *Advances in Neural Information Processing Systems 14*, Cambridge, MA, MIT Press, pp. 1073–1080, 2002.
21. Zhou, Z.-H. and Zhang, M.-L.: Neural networks for multi-instance learning, Technical Report, AI Lab, Computer Science & Technology Department, Nanjing University, China, Aug. 2002.
22. Zhou, Z.-H. and Zhang, M.-L.: Ensembles of multi-instance learners. In: N. Lavrac, D. Gamberger, H. Blockeel and L. Todorovski, (eds), *Lecture Notes in Artificial Intelligence, Vol. 2837*, Berlin, Springer, pp. 492–502, 2003.