Bio-Medical Materials and Engineering 26 (2015) S1969–S1976
DOI 10.3233/BME-151500
IOS Press

A label distance maximum-based classifier for multi-label learning

Xiaoli Liu[a,b], Hang Bao[a], Dazhe Zhao[a,b] and Peng Cao[a,b,*]
[a]*Medical Image Computing Laboratory of Ministry of Education, Northeastern University, 110819, Shenyang, China*
[b]*College of Information Science and Engineering, Northeastern University, 110819, Shenyang, China*

Abstract. Multi-label classification is useful in many bioinformatics tasks such as gene function prediction and protein site localization. This paper presents an improved neural network algorithm, Max Label Distance Back Propagation Algorithm for Multi-Label Classification. The method was formulated by modifying the total error function of the standard BP by adding a penalty term, which was realized by maximizing the distance between the positive and negative labels. Extensive experiments were conducted to compare this method against state-of-the-art multi-label methods on three popular bioinformatic benchmark datasets. The results illustrated that this proposed method is more effective for bioinformatic multi-label classification compared to commonly used techniques.

Keywords: Multi-label classification, neural networks, max label distance, self-adaptive learning rate

1. Introduction

Bioinformatics analysis poses classification challenges because one gene can be associated with several functional classes [1]. This is a typical multi-label classification problem. Multi-label classification is concerned with learning from a set of instances that is associated with a set of labels [2]. Traditional single-label classification generally assigns instances to a single category, where each instance is associated with a single label. Multi-label classification is a complex and challenging task in machine learning [3]. Many researchers have proposed various methods for multi-label learning [2, 4-10], but their effectiveness has not been satisfactory for bioinformatics classification.

This paper presents an improved neural network algorithm, Max Label Distance Back Propagation Algorithm (named MaxLDBP) for Multi-Label Classification. The method was formulated by modifying the total objective function of the standard Back Propagation (BP) by adding a penalty term, which was realized by maximizing the distance between the positive and negative labels. Hence, the total loss of the proposed method was comprised by standard error and the maximum label distance. Also, an adaptive learning rate was used to improve efficiency.

The rest of the paper is organized as follows: Section 2 reviews a standard back propagation neural network. Section 3 details the method. Section 4 presents experimental results and comparisons.

* Address for Corresponding: Peng Cao, College of Information Science and Engineering, Northeastern University, 110819, Shenyang, China. Tel.: (86 24) 8366 5418; Fax: (86 24) 8366 3446; E-mail: caopeng@ise.neu.edu.cn.

Finally, conclusions are summarized in Section 5.

2. Standard back propagation neural network

Consider a three-layered BP of which the *l-th* layer contains N_l units, l=1,…,M. The output of the *j-th* unit at the *l-th* layer is:

$$y_j^l = f\left(net_j^l + \theta_j^l\right) \tag{1}$$

where θ_j^l is the bias for the *j-th* unit, and $f\left(net_j^l\right)$ is the activation function. net_j is the input of the activation function for the *j-th* unit:

$$net_j^l = \sum_{i=1}^{N_{l-1}} w_{ij}^{l-1,l} y_i^{l-1} \tag{2}$$

where $w_{ij}^{l-1,l}$ is the weight from the *i-th* unit at the *(l-1)-th* layer to the *j-th* unit at the the *l-th* layer.

E^s denotes the training error of the Standard BP, as follows:

$$E^S = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{N_M}\left(y_{j,p} - t_{j,p}\right)^2 \tag{3}$$

where p is a pattern over input-output pairs and P is the amount of the training data. The $y_{j,p}$ and $t_{j,p}$ are the network and the target output vectors at the *j-th* output layer unit for the pattern p.

3. Proposed method

3.1. Error function

Many activation functions can be used in BP. The tangent activation function $f(x) = \left(e^x - e^{-x}\right)/\left(e^x + e^{-x}\right)$ was used due to its fast convergence. So, the label distance maximum error function E^L is defined as follows:

$$E^L = \sum_{p=1,k\in Y_p,k\in\overline{Y}_p}^{P}\sum_{j=1}^{N_M}\exp\left(-\frac{1}{2}\left(\min\left(y_{j,k}\right) - \max\left(y_{j,l}\right)\right)\right) \tag{4}$$

where Y_p denotes the labels for x_p.

Hence, the total error of the proposed method is comprised by the standard error, which is introduced in Section 2 and the label distance maximum error:

$$E = E^S + \lambda E^L = \sum_{p=1}^{P}\sum_{j=1}^{N_M}\left(E_P^S + \lambda E_P^L\right) = \sum_{p=1}^{P}\sum_{j=1}^{N_M}\left(\frac{1}{2}\left(y_j - t_j\right)^2 + \lambda\exp\left(-\frac{1}{2}\left(\min\left(y_{j,k}\right) - \max\left(y_{j,l}\right)\right)\right)\right) \tag{5}$$

where λ is the regularization parameter that controls the tradeoff between the training error and label distance maximum error.

This can be written as:

$$\frac{\partial E_P}{\partial w_{ij}} = \frac{\partial E_P^S}{\partial w_{ij}} + \lambda\frac{\partial E_P^L}{\partial w_{ij}} = \left(\frac{\partial E_P^S}{\partial net_j} + \lambda\frac{\partial E_P^L}{\partial net_j}\right)\frac{\partial net_j}{\partial w_{ij}} \tag{6}$$

and by Eq. (2), $\partial net_j / \partial w_{ij} = y_i$ is obtained; when $\delta_j = -\left(\partial E_p^S / \partial net_j + \partial E_p^L / \partial net_j\right)$ is defined by Eq. (1), Eq. (7) is drawn:

$$\delta_j = -\left(\frac{\partial E_p^S}{\partial y_j} + \lambda\frac{\partial E_p^L}{\partial y_j}\right)\frac{\partial y_j}{\partial net_j} = -\left(\frac{\partial E_p^S}{\partial y_j} + \lambda\frac{\partial E_p^L}{\partial y_j}\right)f'\left(net_j + \theta_j\right) \tag{7}$$

considering E^S, we get $\partial E_p^S / \partial y_j = y_j - t_j$. Considering E^L, we get:

$$\frac{\partial E_p^L}{\partial y_j} = \frac{\partial\left(\exp\left(-\frac{1}{2}\left(\min\left(y_{j,k}\right) - \max\left(y_{j,l}\right)\right)\right)\right)}{\partial y_j} = \begin{cases} -\frac{1}{2}\exp\left(-\frac{1}{2}\left(\min\left(y_j\right) - \max\left(y_l\right)\right)\right) & j \in Y_p \\ -\frac{1}{2}\exp\left(-\frac{1}{2}\left(\min\left(y_k\right) - \max\left(y_j\right)\right)\right) & j \in \overline{Y_p} \end{cases} \tag{8}$$

since $f'\left(net_j + \theta_j\right) = \left(1 + c_j\right)\left(1 - c_j\right)$, we get:

$$\delta_j = \begin{cases} \left(\left(t_j - y_j\right) + \frac{1}{2}\lambda\exp\left(-\frac{1}{2}\left(\min\left(y_j\right) - \max\left(y_l\right)\right)\right)\right)\left(1 + c_j\right)\left(1 - c_j\right) & j \in Y_p \\ \left(\left(t_j - y_j\right) - \frac{1}{2}\lambda\exp\left(-\frac{1}{2}\left(\min\left(y_k\right) - \max\left(y_j\right)\right)\right)\right)\left(1 + c_j\right)\left(1 - c_j\right) & j \in \overline{Y_p} \end{cases} \tag{9}$$

The delta rule used in the standard BP was used to update the weights. According to the gradient descent strategy, the weight and bias are changed as follows:

$$\Delta w_{ij} = -\eta\frac{\partial E_p}{\partial w_{ij}} = \eta\delta_j y_i,\ \Delta\theta_j = \eta\delta_j \tag{10}$$

where η is the learning rate, introduced in the next section.

3.2. Learning rate

Vogl, et al. proposed an adaptive learning-rate back-propagation with adaptive momentum (ABP) [11]. He modified the weight after each epoch over all the patterns as follows:

$$\Delta w_{ij}(K+1)=\eta\times\delta_j y_i+mc\times\Delta w_{ij}(K) \tag{11}$$

where K represents the iteration number rather than the presentation number and mc is the momentum factor. The learning rate is determined by whether the total error of all patterns decreases the performance after one iteration or not. If an update successively reduces the total error, η is increased by multiplying a factor $a>1$ for the next iteration. If the total error exceeds the previous value by a certain percentage, η is decreased by multiplying a factor $b<1$. The learning rate is constrained between 0.05 and 0.3.

A modified adaptive learning rate method was used based on ABP, considering the learning rate but not the momentum. The net was updated when the total error reduced successively. The method can be denoted by the following equation:

Algorithm 1
Max label distance back propagation algorithm (MaxLDBP) for multi-label classification

Algorithm MaxLDBP

Initialization weight matrix W, ε, total error E, η, $epoch$,a, b, φ,λ
while $E > \varepsilon$ **&** $K < epochs$ **do**
do $K = K+1$; $E = 0$;
for $p = 1$ **to** P **do**
for the *j*-th unit at the *l*-th layer do
$\Delta w_{ij}=-\eta\dfrac{\partial E_p}{\partial w_{ij}}=\eta\delta_j y_i$;
$w_{ij}=w_{ij}+\Delta w_{ij}$;
$E=E^S+\lambda E^L$;
if $E(K+1)<E(K),\eta(K+1)=a\eta(K),net=net(K+1)$;
else if $E(K+1)\geq\varphi E(K),\eta(K+1)=b\eta(K)$;
else if $E(K+1)=E(K),\eta(K+1)=b\eta(K)$;

$$\eta(K+1)=\begin{cases} a\eta(K), net=net(K+1) & E(K+1)<E(K) \\ b\eta(K) & E(K+1)\geq\varphi E(K) \\ b\eta(K) & E(K+1)=E(K) \\ \eta(K) & \text{otherwise} \end{cases} \tag{12}$$

where φ is the ratio between the update total error and the previous value. The parameters used in ABP and the proposed method are a=1.05, b=0.7, and φ=1.04. η starts with 0.01. Formally, Algorithm 1 describes the proposed method. The overall computational cost of the proposed algorithm is $O(M{\cdot}P{\cdot}E)$, where M is the total amount of the architectural parameters (weights and biases) of the network, P is the amount of training instances, and E is the total amount of training epochs.

Table 1

The property of the training and test datasets (Cardinality is the average number of labels per instance)

Datasets	Attributes	Classes	Training	Test	Cardinality
Yeast	103	14	1211	1196	4.24
Human	440	14	1864	1244	1.19
Plant	440	12	588	390	1.08

4. Experiments

This section provides an empirical evaluation of the proposed method derived from experimental analysis on three popular bioinformatics benchmark datasets. Table 1 presents the main property of the training and test datasets employed in the experiments. Dataset Yeast1 was sampled from biological data; it predicted Yeast Saccharomyces cerevisiae on gene function [12]. Each gene was comprised with multi-label functions, so it could be used as a multi-label dataset. Datasets Human and Plant2 predicted the subcellular locations of proteins according to their sequence. It has been observed that some multiplex proteins can be assigned to multiple locations sites simultaneously.

MaxLDBP was empirically assessed against the state-of-the-art methods for multi-label classification, such as SBP (Standard BP) [13], ABP (Adaptive Learning-rate BP with Adaptive Momentum) [11], the algorithm level for multi-label learning: BPMLL (Backpropagation for Multi-Label Learning) [2], and the strategy level for multi-label learning: BR [4], RakEL [7], CC (Classifier Chain) [14], and ECC (Ensemble of Classifier Chain) [14]. For SBP, ABP, and MaxLDBP, the number of units in the hidden layer was fixed at 10; the number of training epochs was set at 1000. For BPMLL, the parameters defined in [2] were used, which were set to be 20% of the number of input units and 200, respectively. For the strategy level algorithms, a neural network with the same configuration was chosen as the base classifier, and was implemented by a MULAN software package [15]. In addition, a L2 regularization term of all network weights was added to the global error function for each comparable method to avoid overfitting.

Tables 2-4 shows the results of the proposed method and other multi-label classification algorithms on the Yeast, Human, and Plant datasets, where the values in bold indicate the best result obtained by the corresponding method. Table 5 shows the number of wins, losses, and ties for MaxLDBP compared to the other methods across 3 datasets and six evaluation metrics, thereby composing 18 competitions. MaxLDBP achieved the best overall performance for 3 datasets in six evaluation metrics. The proposed method beat the other methods on all datasets in One-Error, Hamming Loss,

Table 2

Experimental results of each multi-label classification algorithms on the Yeast dataset (the evaluation metrics can be referenced in [3])

Methods	One-Error	Ranking Loss	Average Precision	Hamming Loss	F1	AUC
SBP	0.5280	0.2458	0.6459	0.2440	0.5470	0.7524
ABP	0.2419	0.1781	0.7545	0.2039	0.6324	0.8291
BPMLL	0.2368	**0.1749**	0.7506	0.2087	**0.6479**	0.8264
BR	0.3993	0.3097	0.6216	0.2454	0.5635	0.6889
CC	0.3562	0.3238	0.6295	0.2682	0.5499	0.6732
ECC	0.2532	0.1805	0.7476	0.2070	0.6256	0.8270
RakEL	0.2921	0.2135	0.7155	0.2257	0.6033	0.8224
MaxLDBP	**0.2351**	0.1757	**0.7567**	**0.2017**	0.6315	**0.8303**

Table 3

Experimental results of each multi-label classification algorithms on the Human dataset

Methods	One-Error	Ranking Loss	Average Precision	Hamming Loss	F1	AUC
SBP	0.7347	0.2106	0.4955	0.1096	0.2160	0.7914
ABP	0.6288	0.2316	0.5507	0.0906	0.2914	0.8031
BPMLL	0.7280	0.4000	0.3987	0.0862	0.2094	0.5981
BR	0.7115	0.4163	0.4165	0.1214	0.2526	0.5720
CC	0.6929	0.3935	0.4325	0.1179	**0.2960**	0.6026
ECC	**0.6063**	**0.1808**	**0.5693**	0.0851	0.2468	**0.8220**
RakEL	0.6352	0.2311	0.5365	0.0998	0.2296	0.7736
MaxLDBP	0.6283	0.1928	0.5519	**0.0835**	0.1482	0.8065

Table 4

Experimental results of each multi-label classification algorithms on the Plant dataset

Methods	One-Error	Ranking Loss	Average Precision	Hamming Loss	F1	AUC
SBP	0.6938	0.3108	0.4824	0.0896	0.1062	0.6968
ABP	0.6831	0.2766	0.4949	0.1026	**0.2715**	0.7275
BPMLL	0.9477	0.5047	0.2400	0.1043	0.1107	0.4968
BR	0.7893	0.5085	0.3497	0.1403	0.1857	0.5040
CC	0.7812	0.4764	0.3677	0.1435	0.2202	0.5456
ECC	0.6800	0.2406	0.5148	0.0937	0.1425	0.7649
RakEL	0.7045	0.2948	0.4791	0.1033	0.1540	0.7206
MaxLDBP	**0.6456**	**0.2296**	**0.5398**	**0.0893**	0.1561	**0.7711**

Table 5

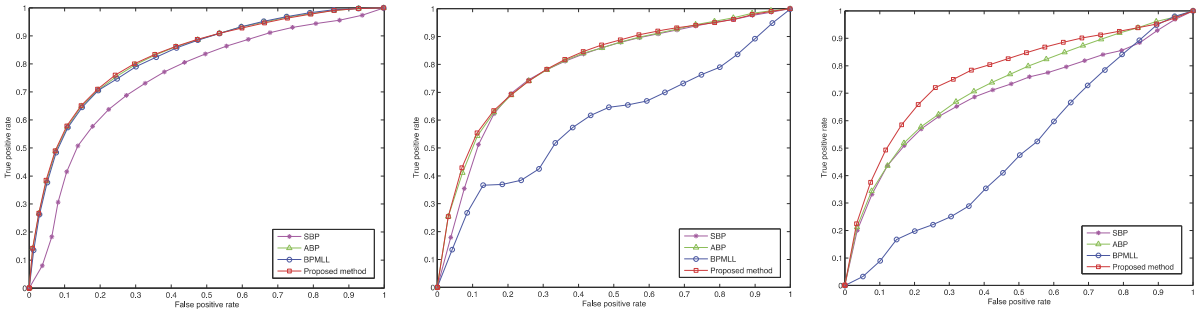
MaxLDBP compared to the rest of the algorithms on three bioinformatics datasets in six different metrics

Algorithm	wins	losses	Algorithm	wins	losses
MaxLDBP vs. SBP	17	1	**MaxLDBP** vs. CC	16	2
MaxLDBP vs. ABP	15	3	**MaxLDBP** vs. ECC	13	5
MaxLDBP vs. BPMLL	15	3	**MaxLDBP** vs. RakEL	17	1
MaxLDBP vs. RakEL	17	1			

and AUC. MaxLDBP particularly performed better than other algorithms on the Plant dataset in all metrics except for F1. The proposed method outperformed the other algorithms in One-Error, Average Precision, Hamming Loss, and AUC, but BP_MLL performed better in Ranking Loss and F1. Moreover, the ABP obtained comparable results, although it did not account for the mechanization of multi-label learning. By contrast, BPMLL only performed well on the Yeast dataset with capturing the characteristics of multi-label learning.

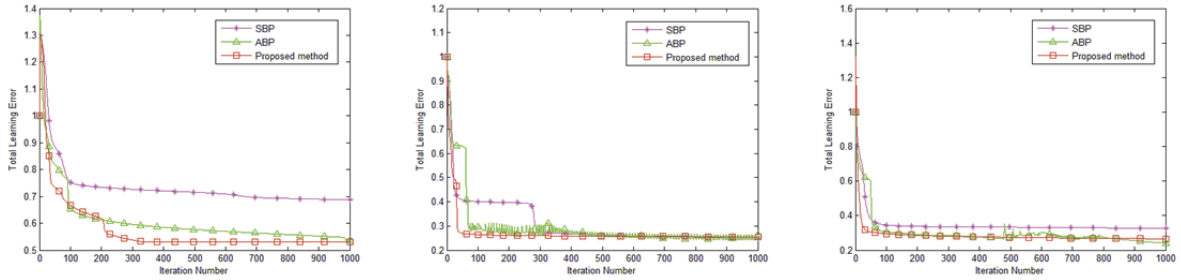
The true and false positive rates of SBP, ABP, BPMLL, and MaxLDBP were calculated to illustrate the performance of the various training algorithms based on BP methods. This was achieved by building a label confusion matrix for each label of each example. Figure 1 shows the ROC dataset curves (Yeast, Human and Plant) to visualize the performance over all instances.

For the Yeast dataset, MaxLDBP exhibited similar behavior to ABP and BPMLL, but performed better than SBP in the AUC measurement. On the Human dataset, the proposed method performed



(a) (b) (c)

Fig. 1. The true and false positive rate (ROC) curves on the datasets. (a) Yeast; (b) Human; (c) Plant.



(a) (b) (c)

Fig. 2. Convergence behavior on the datasets. (a) Yeast; (b) Human; (c) Plant.

similarly with SBP and ABP. The BPMLL algorithm was the worst method. For the Plant dataset, the proposed method outperformed the other methods; the complex correlation among labels was difficult to capture due to the limited training set size. Moreover, the issue of high dimensionality complicated optimization. The proposed method beat the other methods on the Plant dataset because it explored label correlations by maximizing the distance between the positive and negative labels.

Figure 2 illustrates how the global training error changed as the number of training epochs increased. MaxLDBP achieved the optimal solution with fewer iteration numbers compared with SBP and ABP. The convergence behavior demonstrates that the proposed method can improve multi-label data learning network efficiency.

5. Conclusions

This paper proposes an improved neural network algorithm, Max Label Distance Back Propagation Algorithm for Multi-Label Classification. This method was formulated by modifying the total error function of the standard BP by adding a penalty term, which was realized by maximizing the distance between the positive and negative labels. It controlled the magnitude of the weights and improved the network's generalization performance. The method was compared against state-of-the-art multi-label classification methods through extensive experiments; the results illustrated that the proposed method was effective for multi-label classification in bioinformatics, and obtained competitive or better performance compared with five typical multi-label learning algorithms.

Acknowledgment

This research was supported by the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under grant 2014BAI17B01, the Fundamental Research Funds for the Central Universities under Grant N140403004 as well as N140407001, and the Postdoctoral Science Foundation of China 2015M570254.

References

[1] A. Clare, Machine learning and data mining for yeast functional genomics, Ph.D. Dissertation, University of Wales Aberystwyth, 2003.
[2] M.L. Zhang and Z.H. Zhou, Multi-label neural networks with applications to functional genomics and text categorization, IEEE Transactions on Knowledge and Data Engineering **18** (2006), 1338–1351.
[3] M.L. Zhang and Z.H. Zhou, A review on multi-label learning algorithms, IEEE Transactions on Knowledge and Data Engineering **26** (2014), 1819–1837.
[4] M.R. Boutell, J. Luo, X. Shen and C.M. Brown, Learning multi-label scene classification, Pattern Recognition **37** (2004), 1757–1771.
[5] E. Hüllermeier, J. Fürnkranz, W. Cheng and K. Brinker, Label ranking by learning pairwise preferences, Artificial Intelligence **172** (2008), 1897–1916.
[6] J. Fürnkranz, E. Hüllermeier, E.L. Mencía and K. Brinker, Multilabel classification via calibrated label ranking, Machine Learning **73** (2008), 133–153.
[7] G. Tsoumakas, I. Katakis and I. Vlahavas, Random k-Label sets for multi-label classification, IEEE Transactions on Knowledge and Data Engineering **23** (2011), 1079–1089.
[8] A. Clare and R.D. King, Knowledge discovery in multi-label phenotype data, Processing of the 5th European Conference on Principles of Data Mining and Knowledge Discovery **2168** (2001), pp. 42–53.
[9] B.E. Schapire and Y. Singer, Boostexter: A boosting-based system for text categorization, Machine Learning **39** (2000), 135–168.
[10] F.D. Comité, R. Gilleron and M. Tommasi, Learning multi-label alternating decision trees from texts and data, Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition **2734** (2003), pp. 35–49.
[11] T.P. Vogl, J.K. Mangis, J.K. Rigler, W.T. Zink and D.L. Alkon, Accelerating the convergence of the backpropagation method, Biological Cybernetics **59** (1988), 257–263.
[12] A. Elisseeff and J. Weston, A kernel method for multi-labelled classification, in: Advances in Neural Information Processing Systems, MIT Press Cambridge, MA, USA, 2011, pp. 681–687.
[13] D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning internal representations by error propagation, in: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press Cambridge, MA, USA, 1985, pp. 318-362.
[14] J. Reed, B. Pfahringer and G. Holmes, Classifier chain for multi-label classification, Machine Learning **85** (2011), 333–359.
[15] G. Tosumakas, E. Spyromitros-Xioufis, J. Vilcek and I. Vlahavas, MULAN: A java library for multi-label learning, The Journal of Machine Learning Research **12** (2011), 2411–1414.