Error Back-Propagation Algorithm for Classification of Imbalanced Data

Sang-Hoon Oh*,[a]

[a]*Dept. Information Communication Eng., Mokwon University, Daejon, Korea*

Abstract

Classification of imbalanced data is pervasive but it is a difficult problem to solve. In order to improve the classification of imbalanced data, this letter proposes a new error function for the error back-propagation algorithm of multilayer perceptrons. The error function intensifies weight-updating for the minority class and weakens weight-updating for the majority class. We verify the effectiveness of the proposed method through simulations on mammography and thyroid data sets.

Key words: error back-propagation, imbalanced data, error function

1. Introduction

In many classification problems, unusual or interesting class is rare among a general population. This data imbalance has been reported in a wide range of applications such as credit assessment[1], gene ontology[2], remote sensing[3], bio-medical diagnoses[4], etc. However, conventional classifiers show poor performances in these applications since they are based on the assumption that class priors are relatively balanced and error costs of all classes are equal[5].

Many methods have been developed for classification of the imbalanced data. At the data level approach, class distribution is re-balanced by under-sampling[4, 6], over-sampling[7], or combination of the two[7]. At the algorithmic level, modifying error function[3] adapts existing classifier learning

*Corresponding author

Email address: shoh@mokwon.ac.kr (Sang-Hoon Oh)

algorithms to strengthen learning with regards to the minority class. In addition, there are cost-sensitive learning and threshold moving methods at the algorithmic level approach[6, 8]. Also, ensemble scheme has many advantages over each individual classifier[4, 9].

Among the above approaches, developing a better classifier at the algorithmic level is critical because it is the essential part in the data level approach or ensemble of classifiers. In this letter, we propose an error function for the EBP (error back-propagation) algorithm of MLP's (multilayer perceptrons). The proposed error function intensifies weight-updating for the minority class and weakens weight-updating for the majority class. The rest of this letter is organized as follows. In Section 2, we propose an error function which can control the strength of weight-updating with regards to the minority or majority classes. In Section 3, we demonstrate the effectiveness of the proposed method, and Section 4 concludes this letter.

2. Error Function for Classification of Imbalanced Data

Consider an MLP consisting of N inputs, H hidden, and M output nodes, which is denoted as "$N-H-M$" MLP. When a pth training pattern $\mathbf{x}^{(p)} = [x_1^{(p)}, x_2^{(p)}, \ldots, x_N^{(p)}]$, $(p = 1, 2, \ldots, P)$ is presented to the MLP, the jth hidden node is given by

$$h_j^{(p)} \triangleq h_j(\mathbf{x}^{(p)}) = \tanh(\sum_{i=0}^{N} w_{ji} x_i^{(p)}/2), \quad j = 1, 2, \ldots, H. \tag{1}$$

Here, $x_0^{(p)} = 1$ and w_{ji} denotes the weight connecting the ith input x_i to h_j. The kth output node is

$$y_k^{(p)} \triangleq y_k(\mathbf{x}^{(p)}) = \tanh(\hat{y_k}^{(p)}/2), \quad k = 1, 2, \ldots, M, \tag{2}$$

where $\hat{y_k}^{(p)} = \sum_{j=0}^{H} v_{kj} h_j^{(p)}$. Also, $h_0^{(p)} = 1$ and v_{kj} denotes the weight connecting h_j to y_k.

Let the desired output vector corresponding to the training pattern $\mathbf{x}^{(p)}$ be $\mathbf{t}^{(p)} = [t_1^{(p)}, t_2^{(p)}, \ldots, t_M^{(p)}]$, where the class from which $\mathbf{x}^{(p)}$ originates is coded as follows:

$$t_k^{(p)} = \begin{cases} +1, & \text{if } \mathbf{x}^{(p)} \text{ originates from class } k \\ -1, & \text{otherwise.} \end{cases} \tag{3}$$

We call y_k the target node of class k. The conventional error function for P training patterns is

$$E = \frac{1}{2}\sum_{p=1}^{P}\sum_{k=1}^{M}(t_k^{(p)} - y_k^{(p)})^2. \tag{4}$$

To minimize E, weights are iteratively updated by the EBP algorithm[10].

Let us assume that there are two classes, where one is the minority class C_1 with P_1 training patterns and the other is the majority class C_2 with P_2 training patterns ($P_2 >> P_1$). Then, weight-updating in the EBP algorithm is dominated by the P_2 patterns and the boundary of the majority class is enlarged to the minority class[4]. This boundary distortion causes poor performance[3].

Here, we assume that MLP has two outputs whose targets are coded as in (3). During training, y_2 is selected as a target node P_2 times and y_1 is selected P_1 times. Thus, in order to prevent the boundary distortion, we should intensify weight-updating with regards to y_1 and weaken weight-updating with regards to y_2. Accordingly, we propose the error function

$$E_{prop} = -\sum_{p=1}^{P}[\int \frac{{t_1^{(p)}}^{n+1}(t_1^{(p)} - y_1^{(p)})^n}{2^{n-2}(1 - {y_1^{(p)}}^2)}dy_1^{(p)} + \int \frac{{t_2^{(p)}}^{m+1}(t_2^{(p)} - y_2^{(p)})^m}{2^{m-2}(1 - {y_2^{(p)}}^2)}dy_2^{(p)}], \tag{5}$$

where n and m ($n < m$) are positive integers, and $t_k^{(p)} = \pm 1$. If $n = m$, E_{prop} is the same as the nth order error function proposed in [11]. Then, the error signal of output layer is given by

$$\delta_k^{(p)} = -\frac{\partial E_{prop}}{\partial \hat{y_k}^{(p)}} = \begin{cases} {t_1^{(p)}}^{n+1}(t_1^{(p)} - y_1^{(p)})^n/2^{n-1}, & \text{where } k = 1, \\ {t_2^{(p)}}^{m+1}(t_2^{(p)} - y_2^{(p)})^m/2^{m-1}, & \text{where } k = 2. \end{cases} \tag{6}$$

Since $n < m$, $|\delta_1^{(p)}| \geq |\delta_2^{(p)}|$ for $-1 < y_k^{(p)} < 1$. That is, the parameters n and m in (5) generate a strong error signal for the target node of the minority class, y_1, and a weak error signal for the target node of the majority class, y_2. Then, associated weights are updated in proportion to $\delta_1^{(p)}$ and $\delta_2^{(p)}$, respectively.

It was reported that the nth order error function with $n \geq 2$ shows better performance than $n = 1$[11, 12]. Thus, we will use $n = 2$ for updating weights associated with the minority class. Although there are many possibilities in selecting m value which controls the weight-updating for the majority class,

we will use $m = 4$ for simplicity. Through many simulations, it was verified that various m values in the range of $3 \leq m \leq 10$ show similar learning performances.

Since the targets are coded as shown in (3), y_1 has its target value '$1'$ P_1 times and '$-1'$ P_2 times from total P training patterns. The case of y_2 is vice versa. In order to fix this imbalance, $\delta_k^{(p)}$'s are regulated as

$$\delta_k^{(p)} \rightarrow \begin{cases} \gamma\delta_k^{(p)}, & \text{if } (k = 1 \text{ and } t_k^{(p)} = -1) \text{ or } (k = 2 \text{ and } t_k^{(p)} = 1), \\ \delta_k^{(p)}, & \text{otherwise,} \end{cases} \tag{7}$$

with the parameter $\gamma = P_1/P_2$. Table 1 summarizes the proposed algorithm.

When applying MLP's to two-class problems, we can use a single output architecture. In the imbalanced data problems, however, this letter proposes to generate a strong error signal for the target node of minority class and a weak error signal for the other target node. Because of this strategy, the proposed algorithm adopts the MLP with two output nodes.

In the limit $P \rightarrow \infty$, the minimizer of E_{prop} converges (under certain regularity conditions, Theorem 1 in [13]) towards the minimizer of the function

$$E\{\ell_n(T_1, y_1(\mathbf{X})) + \ell_m(T_2, y_2(\mathbf{X}))\}, \tag{8}$$

where $E\{.\}$ is the expectation operator,

$$\ell_n(t, y) = -\int \frac{t^{n+1}(t-y)^n}{2^{n-2}(1-y^2)} dy, \tag{9}$$

$\mathbf{X}$ is the random vector denoting an input pattern, and T_k is the random variable denoting the target. $\ell_m(t, y)$ can be represented by substituting n with m in (9). The expectation is given by

$$E\{\ell_n(T_k, y_k(\mathbf{X}))\} = \int [Q_k(\mathbf{x})\ell_n(1, y_k(\mathbf{x})) + (1 - Q_k(\mathbf{x}))\ell_n(-1, y_k(\mathbf{x}))] f(\mathbf{x}) d\mathbf{x}, \tag{10}$$

where $Q_k(\mathbf{x}) = Pr[\mathbf{X} \text{ originates from class } k | \mathbf{X} = \mathbf{x}]$.

For a fixed $Q_k(\mathbf{x}), 0 < Q_k(\mathbf{x}) < 1$, the optimal solution minimizing the criterion (8) is given by $\mathbf{b}(\mathbf{X}) = [b_1(\mathbf{X}), b_2(\mathbf{X})]^T$, whose components are

$$b_1(\mathbf{x}) = g(h_n(Q_1(\mathbf{x}))) \text{ and } b_2(\mathbf{x}) = g(h_m(Q_2(\mathbf{x}))). \tag{11}$$

Here, h_n & $h_m : (0,1) \rightarrow (0,\infty)$ and $g : (0,\infty) \rightarrow (-1,1)$ are given by

$$h_n(q) = \left(\frac{1-q}{q}\right)^{1/n}, h_m(q) = \left(\frac{1-q}{q}\right)^{1/m}, \text{ and } g(u) = \frac{1-u}{1+u}. \quad (12)$$

Fig. 1 shows the solution with $n = 2$ and $m = 4$. Notice that $g \circ h_n$ and $g \circ h_m$ are strictly increasing and the Bayes classifier can be defined by

$$\text{decide } k \ , \quad \text{if } k = \arg_k[\max\, y_k(\mathbf{x})]. \quad (13)$$

3. Simulations

We have verified the proposed algorithm using "Ann-thyroid"[14] and "Mammography"[7] data sets. The "Ann-thyroid" data is transformed into two-class problems. "Ann-thyroid13(23)" refers to a problem where class1(2) is the minority class while class 3 is treated as the majority class[4]. Tables 2 and 3 describe data set distributions for training and test. For "Mammography" data set, we have used "5-fold cross-validation" since its test data is not provided.

21-16-2 MLP is used for "Ann-thyroid13(23)" and 6-4-2 MLP is used for "Mammography". The proposed method is compared with the conventional EBP algorithm[10], the two-phase method with a parameter T[3], and the threshold moving method with a parameter TH[6]. In the test phase of threshold moving method[6], the class returned is $\arg_k[\max y_k^*]$ where

$$y_k^* = \begin{cases} \frac{1+y_k}{2} \times TH, & \text{for } k = 1, \\ \frac{1+y_k}{2}, & \text{for } k = 2, \end{cases} \quad (14)$$

and $y_k \in (-1,1)$ is the output of conventional MLP. Learning rates η's are derived so that $E\{\eta|\delta_k^{(p)}|\}$ has the same value in each method[11]. As a result, we used $\eta = 0.001 \times [(n+1)+(m+1)]/2$ for the proposed method and $\eta = 0.006$ for the other methods. Let us denote the accuracy for C_1 as A_1 and the accuracy for C_2 as A_2. When data is imbalanced, the total accuracy is inadequate as a performance measure since it heavily relies on A_2. Accordingly, we used the G-Mean (geometric mean) of the two as a performance measure[4]. During training, the performances for test or validation sets were measured in every 10 epochs.

We tried various T and TH values for the two-phase and threshold moving methods, respectively. The best result among them was selected to draw

figures. Nine simulations were conducted using each method with same initializations and the results were averaged to draw figures. The initial weights were drawn at random from a uniform distribution on $[-1 \times 10^{-4}, 1 \times 10^{-4}]$.

Fig. 2 shows the G-Mean in each method for "Ann-thyroid13". The conventional EBP method shows the worst result. Although the two-phase and threshold moving methods improved the performance, they show fluctuations during training. This is due to the incorrect saturation of output nodes, that is, output nodes are in the wrong extreme region of sigmoid function[11]. On the contrary, the proposed method shows better result without fluctuations. Thus, we can argue that the proposed method successfully regulates weight-updating to resolve the imbalanced data problem. Also, the proposed error function inherits the characteristic of the nth order error function, which dramatically reduces the incorrect saturation[11].

For more precise comparison, Table 4(a) shows mean, minimum, and maximum values of A_1, A_2, and G-Mean, respectively. To evaluate them, we extracted A_1, A_2, and G-Mean values at the epoch which showed the best G-Mean in every simulation. And those values were used to calculate the mean, minimum, and maximum values, respectively. As expected, A_1 and G-Mean are the worst in the conventional method. The two-phase and threshold moving methods improved A_1 and G-Mean. The proposed method improved A_1 very much and attained the best G-Mean. Also, $|A_2 - A_1|$ is minimum in the proposed method.

Fig. 3 and Table 4(b) show the simulation results for "Ann-thyroid23" data, and Fig. 4 and Table 4(c) correspond to "Mammography". In these problems, the simulations show similar tendency of A_1, A_2, and G-Mean.

4. Conclusion

In this letter, we proposed an error function for the EBP algorithm of MLP's in order to improve classification of imbalanced data. The proposed error function regulated the updating amount of weights with regards to minority and majority classes. Comparisons were conducted through simulations of "Ann-thyroid" and "Mammography" data sets. The conventional EBP showed the worst A_1 and G-Mean. The two-phase method improved A_1 and G-Mean, but it was unsatisfactory. The threshold moving method could improve the performances further. However, many trials were needed until finding an optimum threshold value. On the contrary, the proposed method attained the best result with the criteria of A_1, G-Mean, and $|A_2 - A_1|$.

The proposed algorithm assumed that targets of MLP are coded as in (3) for two-class problems. If we use a different coding of targets, we should modify the proposed error function E_{prop}. Also, we may not directly use the proposed algorithm for multi-class problems with imbalanced data.

5. Acknowledgement

The author wishes to thank professor Haesun Park for her helpful discussions and proof-readings. Also, we are thankful for critical comments of the anonymous reviewers.

References

[1] Y.-M. Huang, C.-M. Hung, and H. C. Jiau, Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem, Nonlinear Analysis: Real World Applications 7(2006) 720-747.

[2] R. Bi, Y. Zhou, F. Lu, and W. Wang, Predicting gene ontology functions based on support vector machines and statistical significance estimation, Neurocomputing 70(2007) 718-725.

[3] L. Bruzzone and S. B. Serpico, Classification of imbalanced remote-sensing data by neural networks, Pattern Recognition Letters 18(1997) 1323-1328.

[4] P. Kang and S. Cho, EUS SVMs: Ensemble of under-sampled SVMs for data imbalance problems, in: Proc. ICONIP'06, (Springer, Berlin, 2006) 837-846.

[5] F. Provost and T. Fawcett, Robust classification for imprecise environments, Machine Learning 42(2001) 203-231.

[6] Z.-H. Zhou and X.-Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, IEEE Trans. Knowledge and Data Eng. 18(2006) 63-77.

[7] N. V. Chalwa, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16(2002) 321-351.

[8] H. Zhao, Instance weighting versus threshold adjusting for cost-sensitive classification, Knowl. Inf. Syst. 15(2008) 321-334.

[9] Y. Sun, M. S. Kamel, A. K.C. Wong, and Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognition 40(2007) 3358-3378.

[10] D. E. Rumelhart and J. L. McClelland, Parallel Distributed Processing, MIT Press, Cambridge, MA, 1986.

[11] S.-H. Oh, Improving the error back-propagation algorithm with a modified error function, IEEE Trans. Neural Networks 8(1997) 799-803.

[12] S.-H. Oh and S.-Y. Lee, An adaptive learning rate with limited error signals for training of multilayer perceptrons, ETRI Journal 22(2000) 10-18.

[13] H. White, Learning in artificial neural networks: a statistical perspective, Neural Computation 1(1989) 425-464.

[14] A. Frank and A. Asuncion, UCI Machine Learning Repository (2010), http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences.

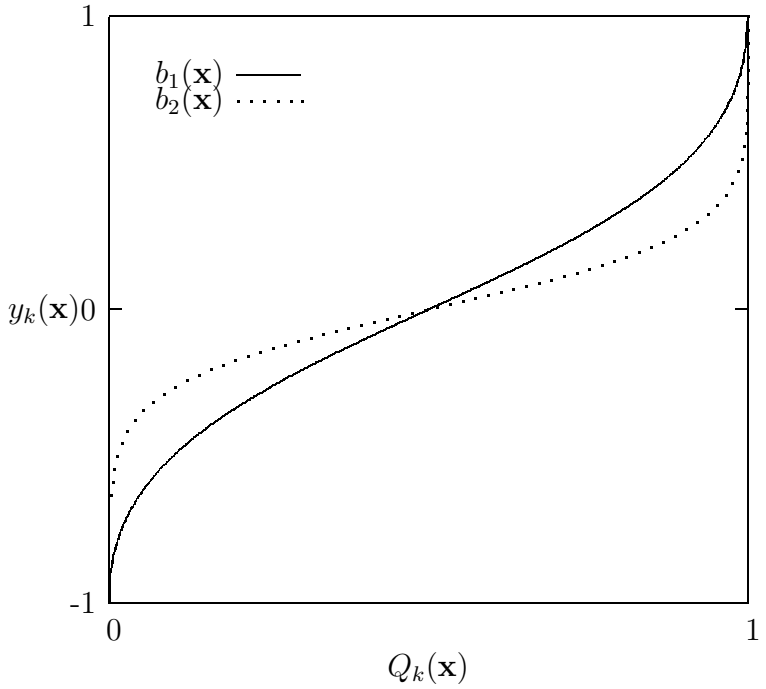

Figure 1: The optimal solutions of $y_k(\mathbf{X})$ for minimizing $E\{E_{prop}(\mathbf{X})\}$. $E\{.\}$ denotes the expectation operator and $E_{prop}(\mathbf{X})$ is the proposed error function when a random vector $\mathbf{X}$ is presented to an MLP as an input pattern. Also, $Q_k(\mathbf{x})$ is the posterior probability $Pr[\mathbf{X}$ originates from class $k|\mathbf{X} = \mathbf{x}]$.

Table 1: Summary of the proposed EBP algorithm for imbalanced data

1.	Initialize an MLP with random weights
2.	Present a training pattern to MLP
3.	Calculate $h_j^{(p)}$ and $y_k^{(p)}$ as in Eq. (1)-(2)
4.	Calculate $\delta_k^{(p)}$ according to Eq. (6)
5.	Regulate $\delta_k^{(p)}$ according to Eq. (7)
6.	Update v_{kj} and w_{ji} as the EBP algorithm
7.	Return to step 2

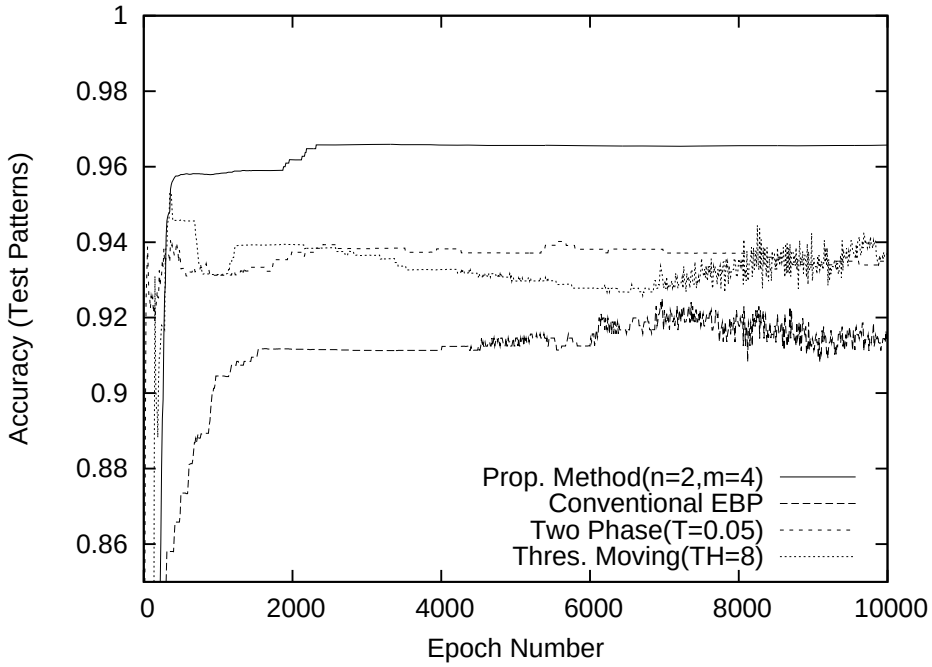

Figure 2: The geometric mean of class accuracies for "Ann-thyroid13".

Table 2: Data set distribution for training

Data Set	Minority Class	Majority Class	Total Patterns	Minority Ratio
Ann-thyroid13	93	3,488	3,581	2.60 %
Ann-thyroid23	191	3,488	3,679	5.19 %
Mammography	260	10,923	11,183	2.32 %

Table 3: Data set distribution for test

Data Set	Minority Class	Majority Class	Total Patterns	Minority Ratio
Ann-thyroid13	73	3,178	3,251	2.25 %
Ann-thyroid23	177	3,178	3,355	5.28 %

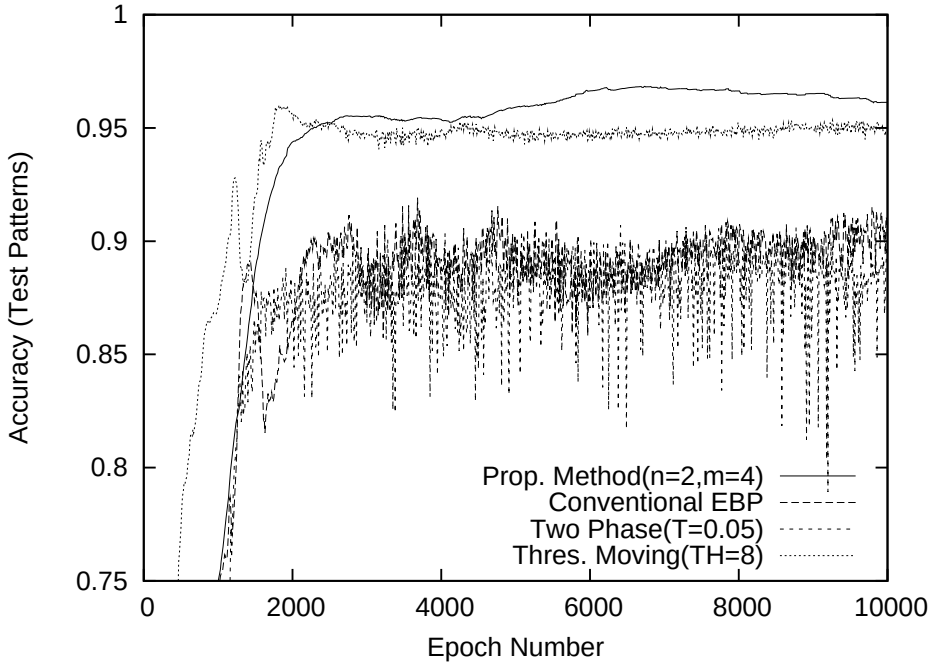

Figure 3: The geometric mean of class accuracies for "Ann-thyroid23".

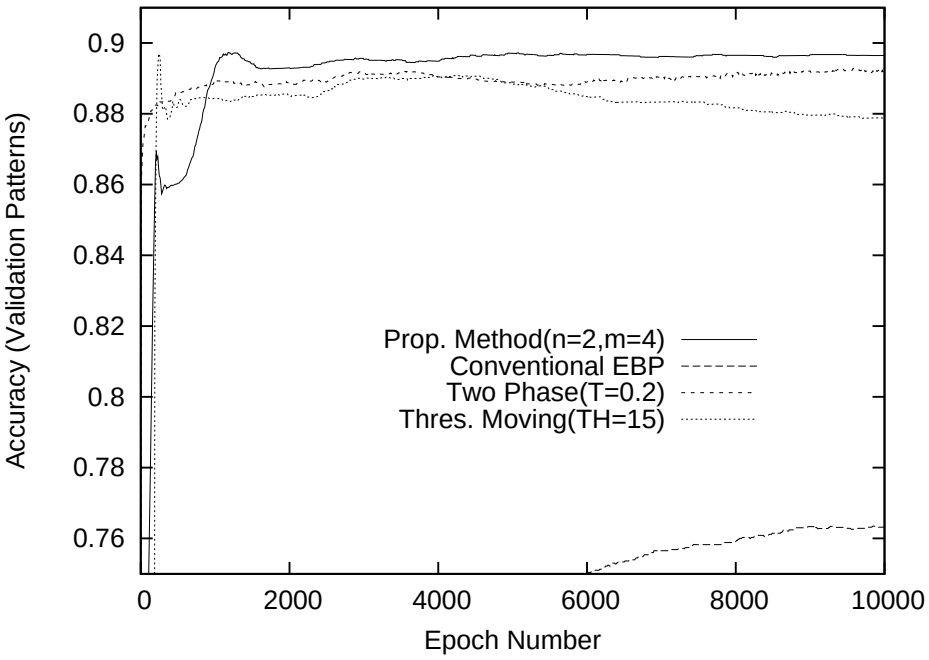

Figure 4: The geometric mean of class accuracies for "Mammography".

Table 4: Test results for (a)"Ann-thyroid13", (b)"Ann-thyroid23", and (c)"Mammography". A_1 denotes the accuracy for the minority class, A_2 is for the majority class, and G-Mean is the geometric mean of A_1 and A_2.

(a)

Training Method		Conv. EBP	Two-Phase	Thres. Mov.	Prop.
A_1	Mean	86.8 %	91.2 %	91.8 %	95.0 %
	Min.	86.3 %	89.0 %	91.8 %	94.5 %
	Max.	89.0 %	94.5 %	91.8 %	95.9 %
A_2	Mean	99.4 %	98.7 %	99.0 %	98.8 %
	Min.	99.3 %	95.7 %	99.0 %	98.8 %
	Max.	99.4 %	99.4 %	99.2 %	98.9 %
G-Mean	Mean	92.9 %	94.8 %	95.3 %	96.9 %
	Min.	92.6 %	94.0 %	95.3 %	96.6 %
	Max.	94.0 %	95.8 %	95.4 %	97.4 %

(b)

Training Method		Conv. EBP	Two-Phase	Thres. Mov.	Prop.
A_1	Mean	90.3 %	89.8 %	97.0 %	97.7 %
	Min.	88.7 %	87.6 %	94.9 %	96.6 %
	Max.	92.7 %	93.2 %	98.3 %	98.3 %
A_2	Mean	98.0 %	97.1 %	96.0 %	96.3 %
	Min.	96.8 %	95.2 %	94.7 %	96.0 %
	Max.	98.5 %	98.8 %	97.5 %	96.8 %
G-Mean	Mean	94.1 %	93.4 %	96.5 %	97.0 %
	Min.	93.2 %	92.7 %	95.9 %	96.7 %
	Max.	95.2 %	94.2 %	96.9 %	97.3 %

(c)

Training Method		Conv. EBP	Two-Phase	Thres. Mov.	Prop.
A_1	Mean	60.4 %	85.3 %	85.0 %	87.8 %
	Min.	49.1 %	75.5 %	77.4 %	79.2 %
	Max.	69.4 %	93.3 %	91.8 %	95.6 %
A_2	Mean	99.6 %	95.9 %	96.1 %	94.1 %
	Min.	99.4 %	93.5 %	93.4 %	92.7 %
	Max.	99.8 %	97.6 %	97.6 %	95.7 %
G-Mean	Mean	77.4 %	90.4 %	90.3 %	90.9 %
	Min.	69.9 %	85.7 %	86.3 %	86.6 %
	Max.	83.1 %	94.3 %	94.4 %	95.1 %