# A Soft Target Learning Method of Posterior Pseudo-probabilities Based Classifiers with Its Application to Handwritten Digit Recognition

*Xuefeng Chen*[1] *Xiabi Liu*[1,2,*] *Yunde Jia*[1]

[1] Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

[2] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China

{crocodel, liuxiabi, jiayunde}@bit.edu.cn

## Abstract

*This paper proposes a soft target discriminative learning method for posterior pseudo-probabilities based classification. The empirical loss is measured based on two soft targets which are corresponding with positive samples and negative samples of the class. The learning objective is to minimize empirical loss and maximize the difference between two soft targets. Consequently, we obtain unknown parameters in posterior pseudo-probabilities based classifiers by optimizing the objective using the gradient descent algorithm. We apply the proposed soft target method to handwritten digit recognition. Experimental results on MNIST database show the effectiveness of our method.*

**Keywords**: Discriminative learning; Soft target; Posterior pseudo-probabilities; Bayesian classifiers; handwritten digit recognition;

## 1. Introduction

Posterior pseudo-probability is a new tool in Bayesian classification, which has been successfully applied to text extraction, image retrieval, and digit recognition [1-4]. The original approach to learning posterior pseudo-probability based classifiers is called Max-Min posterior Pseudo-probabilities (MMP). In the MMP, the posterior pseudo-probabilities of each class for its positive samples are maximized towards 1, while those for its negative samples are minimized towards 0. However, it is nearly impossible to reach hard targets of 0 and 1. Furthermore, using hard targets risks overfitting the training samples. Recently, soft target or soft margin based learning methods have received a lot of interests in the community of machine learning. Previous works show that soft target learning methods provide better generalization performance [5-9].

In this paper, we propose a novel soft target learning method of posterior pseudo-probabilities based classifiers. Two posterior pseudo-probabilities of each class for its positive samples and its negative samples are introduced as soft targets and used to measure the empirical loss. Accordingly, we obtain the values of soft targets and the optimal parameters in the posterior pseudo-probability measure functions of the classes through minimizing the empirical loss and maximizing the difference between two soft targets. The corresponding objective function is designed and solved by the gradient descent algorithm.

We applied the proposed soft target learning method to handwritten digit recognition. The experiments were conducted on MNIST database. Using the gradient direction features [10], we achieved the recognition rates of 99.52% and 98.99% on the training set and the test set, respectively. The results are better than those from baseline MMP learning algorithm. Furthermore, the soft target learning method is much faster than the MMP learning algorithm since the samples with zero classification loss are excluded from the learning process. Interestingly, the soft targets learned from the training data seem to be useful for measuring separability between classes. For the class which is distinctly distinguished from other classes, such as digit 0, the difference between the two soft targets is large. Oppositely, it is small for the class which is easy to be confused with some other classes, such as digit 8.

The rest of this paper is organized as follows. Section 2 briefly introduces the posterior pseudo-probabilities based classification approach. Section 3 presents the proposed soft target learning method. Section 4 discusses the application of the proposed method to handwritten digit

[*] Corresponding author. Tel.: +86 10 68913447, Fax: +86 10 86343158

recognition and the corresponding experimental results. We conclude in Section 5.

## 2. Posterior pseudo-probabilities based classification

Here we briefly introduce the posterior pseudo-probabilities based classification approach. The reader is referred to our paper for more details [1].

Let $\boldsymbol{x}$ be the feature vector, $C_i$ be the $i$ -th class, $p(\boldsymbol{x}|C_i)$ be the class-conditional probability density function, then the posterior pseudo-probability of being $C_i$ for $\boldsymbol{x}$ is computed as

$$f\left(p(\boldsymbol{x}|C_i)\right) = 1 - \exp\left(-\lambda p(\boldsymbol{x}|C_i)\right), \qquad (1)$$

where $\lambda$ is a positive number. $f\left(p(\boldsymbol{x}|C_i)\right)$ is a smooth, monotonically increasing function of $p(\boldsymbol{x}|C_i)$. When $p(\boldsymbol{x}|C_i) = 0$ , $f\left(p(\boldsymbol{x}|C_i)\right) = 0$ , and when $p(\boldsymbol{x}|C_i) = +\infty$ , $f\left(p(\boldsymbol{x}|C_i)\right) = 1$ . We introduce and use $f\left(p(\boldsymbol{x}|C_i)\right)$ to imitate the posterior probability.

For any input pattern, we compute the corresponding posterior pseudo-probabilities of all the classes under consideration. Then the input pattern is classified as the class $C^*$ with the maximum posterior pseudo-probability, i.e.

$$C^* = \arg\max_{C_i} f\left(p(\boldsymbol{x}|C_i)\right). \qquad (2)$$

## 3. Soft Target Learning

### 3.1. Soft target posterior pseudo-probabilities

The posterior pseudo-probability takes values in $[0,1]$. We expect that the posterior pseudo-probabilities of a class for all its positive samples are measured as 1, while those for all its negative samples are measured as 0. However, it is nearly impossible to reach hard target values of 0 and 1. So we introduce two soft target values of posterior pseudo-probabilities for positive samples and negative samples of each class. Suppose they are $\widehat{H}$ and $\overline{H}$ in the following description. Fig. 1 illustrates two soft target values through an example, where white dots and black dots denote posterior pseudo-probabilities for the positive samples and the negative samples of a class, respectively. In Fig. 1, $d$ is the difference between two soft targets $\hat{H}$ and $\overline{H}$ :

$$d = \hat{H} - \overline{H} \qquad (3)$$

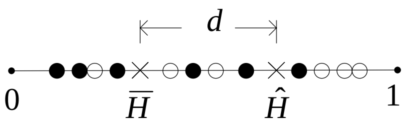

**Figure 1**. Illustration of soft target posterior pseudo-probabilities.

### 3.2. Empirical loss and Objective function

Let $\hat{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$ be the feature vector of arbitrary positive and negative sample of the $i$ -th class, respectively. Based on soft targets described above, the empirical loss on positive samples and negative samples of the $i$ -th class are respectively measured as

$$\hat{l}(\hat{\boldsymbol{x}};\boldsymbol{\Lambda}) = \begin{cases} 0 & , \quad f\left(p(\hat{\boldsymbol{x}}|C_i)\right) > \hat{H} \\ \hat{H} - f\left(p(\hat{\boldsymbol{x}}|C_i)\right), & \quad f\left(p(\hat{\boldsymbol{x}}|C_i)\right) \le \hat{H} \end{cases}, \qquad (4)$$

and

$$\bar{l}(\overline{\boldsymbol{x}};\boldsymbol{\Lambda}) = \begin{cases} 0 & , \quad f\left(p(\overline{\boldsymbol{x}}|C_i)\right) < \overline{H} \\ f\left(p(\overline{\boldsymbol{x}}|C_i)\right) - \overline{H}, & \quad f\left(p(\overline{\boldsymbol{x}}|C_i)\right) \ge \overline{H} \end{cases}. \qquad (5)$$

In Eq. 4-5, $\boldsymbol{\Lambda}$ denote the unknown parameters in the empirical loss measure function, including $\widehat{H}$ , $\overline{H}$ , and those in $f\left(p(\boldsymbol{x}|C_i)\right)$.

Let $m$ and $n$ be the number of positive samples and negative samples of the $i$ -th class in the training set. Then the total empirical loss $L(\boldsymbol{\Lambda})$ for the $i$ -th class is defined as

$$L(\boldsymbol{\Lambda}) = \frac{1}{m}\sum_{i=1}^{m} \hat{l}^2(\hat{\boldsymbol{x}}_i;\boldsymbol{\Lambda}) + \frac{1}{n}\sum_{i=1}^{n} \bar{l}^2(\overline{\boldsymbol{x}}_i;\boldsymbol{\Lambda}) \qquad (6)$$

Besides the empirical loss, the difference between two soft targets is also important for the performance of classifiers. Even zero empirical loss is meaningless if the difference between two soft targets is small or even negative. Therefore, the objective of our soft target learning method is to minimize the empirical loss and maximize the difference between $\widehat{H}$ and $\overline{H}$ . Let $\omega$ be the weight to control the tradeoff between the empirical loss and the difference between two soft targets, then the objective function for learning parameters is designed as

$$F(\boldsymbol{\Lambda}) = \omega(1-d)^2 + (1-\omega)L(\boldsymbol{\Lambda}) \qquad (7)$$

Consequently, we can obtain the optimum parameter set $\boldsymbol{\Lambda}^*$ by minimizing $F(\boldsymbol{\Lambda})$:

$$\boldsymbol{\Lambda}^* = \arg\min_{\boldsymbol{\Lambda}} F(\boldsymbol{\Lambda}). \qquad (8)$$

### 3.3. Optimization Methods

The optimum parameter set is searched by minimizing Eq. 6 through the gradient descent algorithm. In fact, the following iterative equation is used to update the parameters:

$$\boldsymbol{\Lambda}_{t+1} = \boldsymbol{\Lambda}_t - \alpha_t \nabla F\left(\boldsymbol{\Lambda}_t\right), \tag{9}$$

where $\boldsymbol{\Lambda}_t$ and $\alpha_t$ are the parameter set and the step size in the $t$-th iteration respectively, $\nabla F\left(\boldsymbol{\Lambda}_t\right)$ is the partial derivatives of $F\left(\boldsymbol{\Lambda}\right)$ with respect to the parameters in $\boldsymbol{\Lambda}_t$.

According to Eq. 9, the soft target learning algorithm for each class is described as follows. The whole procedure of the soft target learning is to perform this algorithm for all the classes under consideration.

Step1. Compute the partial derivative of $F\left(\boldsymbol{\Lambda}\right)$ with respect to each parameter, where all positive samples and negative samples of the class are involved.

Step2. Compute the step size $\alpha_t$ as $\alpha_t = \alpha_0\left(t_{\max} - t\right)/t_{\max}$, where $t_{\max}$ is the preset maximum number of iterations.

Step3. Update the parameters using Eq. 9.

Step4. Repeat Step 1-3 until convergence or $t_{\max}$ is reached. Let $\varepsilon$ be an infinitesimal, the convergence condition is

$$F\left(\boldsymbol{\Lambda}_t\right) - F\left(\boldsymbol{\Lambda}_{t+1}\right) \le \varepsilon .$$

## 4. Handwritten digit recognition based on soft target learning

### 4.1. Digit modeling and learning

We apply the proposed soft target learning method to handwritten digit recognition. The gradient direction features are extracted from the original gray-scale images and used to represent digits in the experiments [10]. In this paper, the original 200-D feature vector is transformed into 50-D using the Principal Component Analysis (PCA) method.

In this work, the feature vectors extracted from the instances of each digit class is assumed to be of Gaussian mixture model (GMM). Let $k$ be the number of Gaussian components in the GMM, $w_k$, $\boldsymbol{\mu_k}$, and $\boldsymbol{\Sigma_k}$ respectively be the weight, the mean, and the covariance matrix of the $k$-th Gaussian component, $\sum_{k=1}^{K} w_k = 1$, then we have

$$p\left(\boldsymbol{x}\middle|C_i\right) = \sum_{k=1}^{K} w_k N\left(\boldsymbol{x}\middle|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right), \tag{10}$$

where

$$\begin{aligned} & N\left(\boldsymbol{x}\middle|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \\ & = \left(2\pi\right)^{-25}\left|\boldsymbol{\Sigma}_k\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}_k\right)^T \boldsymbol{\Sigma}_k{}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}_k\right)\right) \end{aligned} \tag{11}$$

$\boldsymbol{\Sigma_k}$ is further assumed to be diagonal for simplicity, i.e., $\boldsymbol{\Sigma_k} = \left[\sigma_{kj}\right]_{j=1}^{50}$.

By substituting Eq. 10 into Eq. 1, we get the corresponding digit classification algorithm based on posterior pseudo-probabilities. Under soft target learning scheme described in Section 3, the unknown parameter set in the process of learning classifiers is

$$\boldsymbol{\Lambda} = \{\lambda, w_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \hat{H}, \overline{H}\}, k = 1, \cdots, K . \tag{12}$$

In Eq. 12 some parameters must satisfy certain constraints, which are transformed to unconstrained domain for easier implementation. The constraints and transformation of parameters are listed as follows.

1)$\because 0 < \hat{H} < 1, \therefore \hat{H} \rightarrow h_1 : \hat{H} = \dfrac{1}{1+e^{-h_1}}$

2)$\because 0 < \overline{H} < 1, \therefore \overline{H} \rightarrow h_2 : \overline{H} = \dfrac{1}{1+e^{-h_2}}$

3)$\because \sum w_k = 1, \therefore w_k \rightarrow \tilde{w}_k : w_k = \dfrac{e^{\tilde{w}_k}}{\sum e^{\tilde{w}_k}}$

4)$\because \lambda > 0, \therefore \lambda \rightarrow \tilde{\lambda} : \tilde{\lambda} = \ln \lambda$

5)$\because \sigma_{kj} > 0, \therefore \sigma_{kj} \rightarrow \tilde{\sigma}_{kj} : \tilde{\sigma}_{kj} = \ln \sigma_{kj}$ .

To sum up, the unknown parameter set after transformation is

$$\tilde{\boldsymbol{\Lambda}} = \{\tilde{\lambda}, \tilde{w}_k, \boldsymbol{\mu}_k, \tilde{\boldsymbol{\Sigma}}_k, h_1, h_2\}, k = 1, \cdots, K \tag{13}$$

Accordingly, we use soft target learning method to estimate these parameters and transform them into the original ones. After training, the input pattern is classified according to Eq. 2.

The partial derivatives of $F\left(\boldsymbol{\Lambda}\right)$ with respect to the parameters in $\tilde{\boldsymbol{\Lambda}}$ are provided in Appendix.

### 4.2. Experimental results

We conducted the experiment of handwritten digit recognition on the MNIST database [11], which includes 60000 training samples and 10000 test samples.

Since automatic determination of the number of components in the GMM is an open problem, we set it to 10 through experiments. At first, we used the Expectation-Maximization (EM) algorithm on positive samples to get the Maximum Likelihood Estimation (MLE) of parameters in the GMM and set $\lambda$ through experiments.

Then the MMP learning method and the soft target learning method were respectively used on all the samples including positive samples and negative samples to revise the initial parameters obtained by the EM algorithm. In the soft target learning, the $\widehat{H}$ and $\overline{H}$ for the $i$-th class are initialized as

$$\widehat{H} = \underset{\hat{\mathbf{x}} \in \hat{\mathbf{X}}}{Min}\left[ f\left( p\left( \hat{\mathbf{x}} \middle| C_i \right) \right) \right] \tag{14}$$

and

$$\overline{H} = \underset{\bar{\mathbf{x}} \in \overline{\mathbf{X}}}{Max}\left[ f\left( p\left( \bar{\mathbf{x}} \middle| C_i \right) \right) \right], \tag{15}$$

where $\hat{X}$ and $\overline{X}$ are the sets of positive samples and negative samples of the $i$-th class.

It should be noted that in all the experiments, the value of 0.05 is assigned to $\omega$ in Eq. 7.

Using each of three sets of estimated parameters by the EM algorithm, the MMP algorithm, and the soft target algorithm, closed and open tests of posterior pseudo-probabilities based handwritten digit recognition were implemented. The same experiments were also conducted by using Multi-Layer Percetron (MLP) with back-propagation (BP) learning algorithm.

The recognition rates achieved by four methods are listed in Table 1, where "Train" and "Test" respectively denote recognition rates on training set and test set. The generalization ability of learning algorithm is indicated by the ratio of the recognition rate on the test set to that on the training set, which is denoted as "Test / Train" in Table 1. Obviously, the more value of the ratio is, the better generalization is. Compared with the EM algorithm, the soft target learning brings 79.49% reduction in error rate for the training data, and 57.56% reduction in error rate for the test data. Compared with the MMP algorithm, the soft target learning brings 18.64% reduction in error rate for the training data, and 20.47% reduction in error rate for the test data. Compared with the MLP, the posterior pseudo-probability based classifier with the soft target learning algorithm brings 21.31% reduction in error rate for the training data, and 34.84% reduction in error rate for the test data. Moreover, these results in the Table 1 also show that the soft target algorithm has better generalization ability than the MMP algorithm and the MLP.

**Table 1.** Performance comparison of EM algorithm, MMP algorithm, soft target algorithm, and MLP.

| Learning algorithms | Train(%) | Test(%) | Test / Train |
|---|---|---|---|
| EM | 97.66 | 97.62 | 0.9995 |
| MLP | 99.39 | 98.45 | 0.9905 |
| MMP | 99.41 | 98.73 | 0.9931 |
| Soft Target | 99.52 | 98.99 | 0.9947 |

In the experiments, we also recorded the estimated soft targets and corresponding numbers of training samples for which the empirical loss is non-zero. The results are listed in Table 2, where $r$ denote corresponding numbers of training samples for which the empirical loss is non-zero. As shown in Table 2, $r$ is much smaller than the number of all training samples. So the learning speed can be improved by using soft target learning. Furthermore, it seems that the difference between two soft targets is useful for measuring the separabiltiy between classes. If a digit class is distinctly distinguished from other digit classes, the difference between $\widehat{H}$ and $\overline{H}$ is large, such as the case corresponding with digit 0. Oppositely, it is small for the digit class which is easy to be confused with other digit classes, such as the case corresponding with digit 8. We will investigate this interesting feature in the future.

**Table 2.** The values of two soft targets, the difference between two soft targets and the number of training samples with non-zero classification loss after soft target learning for each digit class.

| Classes | $\widehat{H}$ | $\overline{H}$ | $d$ | $r$ |
|---|---|---|---|---|
| 0 | 0.61 | 0.15 | 0.46 | 1561 |
| 1 | 0.57 | 0.21 | 0.36 | 1794 |
| 2 | 0.57 | 0.30 | 0.27 | 2260 |
| 3 | 0.58 | 0.27 | 0.31 | 2290 |
| 4 | 0.58 | 0.27 | 0.31 | 2268 |
| 5 | 0.54 | 0.29 | 0.25 | 2280 |
| 6 | 0.60 | 0.25 | 0.35 | 1601 |
| 7 | 0.57 | 0.32 | 0.25 | 2562 |
| 8 | 0.56 | 0.33 | 0.23 | 2750 |
| 9 | 0.56 | 0.31 | 0.25 | 2037 |

## 5. Conclusions

In this paper, a novel soft target method has been proposed to learn posterior pseudo-probabilities based classifiers. Our main contribution is the learning objective function which is designed on two soft targets corresponding with positive samples and negative samples of each class. We try to minimize the empirical loss measured based on two soft targets and maximize the difference between two soft targets. In this way, the unknown parameters in the posterior pseudo-probability measure function of each class and the values of two soft targets are estimated from the training data.

We apply the proposed soft target learning method to handwritten digit recognition. The experiments were conducted on MNIST database. Compared with the Maximum Likelihood Estimation (MLE) of parameters, the soft target learning brings 79.49% reduction in error rate on the training set and 57.56% reduction in error rate on the test set. Compared with the original MMP learning approach to posterior pseudo-probability based classifiers, the soft target learning brings 18.64% reduction in error rate on the training set and 20.47% reduction in the error rate on the test set. Compared with the Multi-Layer Perception (MLP), the posterior pseudo-probability based classifier with soft target learning algorithm brings 21.31% reduction in error rate on the training set and 34.84% reduction in the error rate on the test set. Furthermore, an interesting feature of soft targets was observed in the experiments. It seems that the soft targets after training reflect the seperability between the classes. The difference between two soft targets for the class which is distinctly distinguished from other classes is larger than that for the class which is easy to be confused with other classes. We will investigate this interesting feature in the future work.

## Appendix

This appendix provides the partial derivatives of Eq. 7 with respect to the parameters in it, where $N\left(\boldsymbol{x}\middle|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)$ is simplified as $N_k(\boldsymbol{x})$, $k$ is the sequence number of the component in the GMM, $j$ is the sequence number of the element in the 50-D directional feature vector:

$$\frac{\partial F}{\partial h_1}=2\left(\omega(1-d)+\frac{1-w}{m}\sum_{i=1}^{m}\hat{l}(\hat{\boldsymbol{x}}_i;\boldsymbol{\Lambda})\right)\hat{H}^2e^{-h_1}, \tag{16}$$

$$\frac{\partial F}{\partial h_2}=-2\left(\omega(1-d)+\frac{1-w}{n}\sum_{i=1}^{n}\bar{l}(\bar{\boldsymbol{x}}_i;\boldsymbol{\Lambda})\right)\overline{H}^2e^{-h_2}, \tag{17}$$

$$\frac{\partial F}{\partial\tilde{\lambda}}=2(1-\omega)\left(\frac{1}{n}\sum_{i=1}^{n}\bar{l}(\bar{\boldsymbol{x}}_i;\boldsymbol{\Lambda})\frac{\partial f}{\partial\tilde{\lambda}}-\frac{1}{m}\sum_{i=1}^{m}\hat{l}(\hat{\boldsymbol{x}}_i;\boldsymbol{\Lambda})\frac{\partial f}{\partial\tilde{\lambda}}\right), \tag{18}$$

$$\frac{\partial F}{\partial\tilde{w}_k}=2(1-\omega)\left(\frac{1}{n}\sum_{i=1}^{n}\bar{l}(\bar{\boldsymbol{x}}_i;\boldsymbol{\Lambda})\frac{\partial f}{\partial\tilde{w}_k}-\frac{1}{m}\sum_{i=1}^{m}\hat{l}(\hat{\boldsymbol{x}}_i;\boldsymbol{\Lambda})\frac{\partial f}{\partial\tilde{w}_k}\right), \tag{19}$$

$$\frac{\partial F}{\partial\mu_{kj}}=2(1-\omega)\left(\frac{1}{n}\sum_{i=1}^{n}\bar{l}(\bar{\boldsymbol{x}}_i;\boldsymbol{\Lambda})\frac{\partial f}{\partial\mu_{kj}}-\frac{1}{m}\sum_{i=1}^{m}\hat{l}(\hat{\boldsymbol{x}}_i;\boldsymbol{\Lambda})\frac{\partial f}{\partial\mu_{kj}}\right), \tag{20}$$

$$\frac{\partial F}{\partial\tilde{\sigma}_{kj}}=2(1-\omega)\left(\frac{1}{n}\sum_{i=1}^{n}\bar{l}(\bar{\boldsymbol{x}}_i;\boldsymbol{\Lambda})\frac{\partial f}{\partial\tilde{\sigma}_{kj}}-\frac{1}{m}\sum_{i=1}^{m}\hat{l}(\hat{\boldsymbol{x}}_i;\boldsymbol{\Lambda})\frac{\partial f}{\partial\tilde{\sigma}_{kj}}\right). \tag{21}$$

In Eq. 18-21,

$$\frac{\partial f}{\partial\tilde{\lambda}}=\left(\sum_{k=1}^{K}w_kN_k(\boldsymbol{X})\right)e^{\left(-\lambda\left(\sum_{k=1}^{K}w_kN_k(\boldsymbol{X})\right)+\tilde{\lambda}\right)}, \tag{22}$$

$$\frac{\partial f}{\partial\tilde{w}_k}=\lambda w_kN_k(\boldsymbol{X})(1-w_k)e^{\left(-\lambda\left(\sum_{k=1}^{K}w_kN_k(\boldsymbol{X})\right)\right)}, \tag{23}$$

$$\frac{\partial f}{\partial\mu_{kj}}=\lambda w_kN_k(\boldsymbol{X})\left(\frac{x_j-\mu_{kj}}{\sigma_{kj}}\right)e^{\left(-\lambda\left(\sum_{k=1}^{K}w_kN_k(\boldsymbol{X})\right)\right)}, \tag{24}$$

$$\frac{\partial f}{\partial\tilde{\sigma}_{kj}}=\lambda w_kN_k(\boldsymbol{X})\left(\frac{\left(x_j-\mu_{kj}\right)^2}{2\sigma^2{}_{kj}}\right)e^{\left(-\lambda\left(\sum_{k=1}^{K}w_kN_k(x)\right)+\tilde{\sigma}_{kj}\right)}. \tag{25}$$

## References

[1] Xiabi Liu,Yunde Jia, Xuefeng Chen, Yuan Deng, and Hui Fu, “Max-Min Posterior Pseudo-Probabilities Estimation of Posterior Pseudo-Probabilities Estimation of Posterior Class”, Technical Report, 2006 , http://www.mcislab.org.cn/technicalreports/MMP.PDF .

[2] Xiabi Liu, HuiFu, Yunde Jia, “Gaussian mixture modeling and learning of neighboring characters for multilingual text extraction in images”, Pattern Recognition, Vol. 41, 2008, pp. 484-493.

[3] Yuan Deng, Xiabi Liu, Yunde Jia, “Learning Semantic Concepts for Image Retrieval Using the Max-min Posterior Pseudo-Probabilities Method”, IEEE International Conference on Multimedia and Expo, 2007, pp. 1970-1973.

[4] Xuefeng Chen, Xiabi Liu, Yunde Jia, “Learning Handwritten Digit Recognition by the Max-Min Posterior Pseudo-Probabilities Method”, Ninth International Conference on Document Analysis and Recognition, 2007, pp. 342-346.

[5] Michael Rimer, Tony Martinez, “Classification-based objective functions”, Machine Leaning, Vol.63, No.2, 2006, pp. 183-205.

[6] Rich Caruana, shumeet Baluja, Tom Mitchell, “Using the Future to “Sort Out” the Present: Rankprop and Multitask Learning for Medical Risk Evaluation”, Neural Information Processing Systems 8, 1996.

[7] Jinyu Li, Ming Yuan, Chin-Hui Lee, “Approximate Test Risk Bound Minimization Through Soft Margin Estimation”, IEEE Trans. Audio, Speech, And Language Processing, Vol.15, No.8, 2007, pp. 2393-2404.

[8] John Shawe-Taylor, Nello Cristianini, “On the Generalization of Soft Margin Algorithms”, IEEE Trans. Information Theory, Vol. 48, No.10, 2002, pp. 2721-2735.

[9] G.RÄTSCH, T.ONODA, K.-R. MÜLLER, “Soft Margins for AdaBoost”, Machine Leaning, Vol.42, No.3, 2001, pp. 287-320.

[10] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, Hiromichi Fujisawa, “Handwritten digit recognition: benchmarking of state-of-art techniques”, Pattern Recognition, Vol.36, 2003, pp. 2271-2285.

[11] Y. LeCun, et al., “Comparison of Learning Algorithms for Handwritten Digit Recognition”, Proceedings of The International Conference on Artificial Neural Networks, Nanterre, France, 1995, pp.53-60.