Modified Minimum Classification Error Learning and Its Application to Neural Networks

Hiroshi SHIMODAIRA[1], Jun ROKUI[1], and Mitsuru NAKAI[1]

School of Information Science
Japan Advanced Institute of Science and Technology
Tatsunokuchi, Ishikawa, 923–1292 JAPAN
{sim, rokui, mit}@jaist.ac.jp
http://www-ks.jaist.ac.jp/index.html

Abstract. A novel method to improve the generalization performance of the Minimum Classification Error (MCE) / Generalized Probabilistic Descent (GPD) learning is proposed. The MCE/GPD learning proposed by Juang and Katagiri in 1992 results in better recognition performance than the maximum-likelihood (ML) based learning in various areas of pattern recognition. Despite its superiority in recognition performance, as well as other learning algorithms, it still suffers from the problem of "over-fitting" to the training samples. In the present study, a regularization technique has been employed to the MCE learning to overcome this problem. Feed-forward neural networks are employed as a recognition platform to evaluate the recognition performance of the proposed method. Recognition experiments are conducted on several sorts of data sets.

1 Introduction

It is well-known that, theoretically, the Bayes decision rule would give the optimum decision that achieves the minimum classification risk if one can predict the exact probabilistic parameters of the target categories beforehand. However, in case of real world problems, as the number of training data for estimating the probabilistic parameters by the maximum likelihood (ML) method is restricted, the ML-based Bayes classifiers sometimes performs poorer recognition than the classifiers trained by non-parametric learning scheme such as LSE (least squared error) based neural networks and discriminant learning to minimize the recognition error.

The idea of Minimum Classification Error (MCE) / Generalized Probabilistic Descent (GPD) learning was first proposed in 1992 by Juang and Katagiri [1] to establish a general learning scheme for minimizing classification error of arbitrary discriminant functions. Although a number of discriminative-learning algorithms have been proposed so far [2][3][4], the MCE learning is unique in the sense that it is applicable to arbitrary discriminant functions that are differentiable in respect to the parameters that are to be adapted. To be specific, it can be applied to discriminant functions that deal with variable record length of data like speech recognition.

The superiority of the MCE learning to the conventional ML based learning has been shown for various functions such as linear-discriminant functions, MLP (multi-layer perceptron), DTW (dynamic time warping) [5] and HMM (Hidden Markov Models) [6]. Since the MCE learning mainly tries to minimize a cost function that corresponds to the number of classification error for a given training dataset, the generalization perfomance is not adequate against unseen data. In another word, over-fitting to the training data is inevitable.

In order to improve the generalization ability of the MCE learning, a regularization technique, which is widely used to solve ill-posed problems[7], is employed in this study.

This paper is divided into five sections. The next section describes the MCE learning briefly. The third section describes the proposed algorithm of modifying the MCE. The fourth section presents experimental results. Finally, the last section is devoted to conclusion.

2 Minimum Classification Error Learning

Let $g_k(\mathbf{x}; \Lambda_k)$ be a discriminant function with positive value to discriminate a data of class Ω_k from the other classes, where $\mathbf{x} = (x_1, \ldots, x_D)$ and Λ_k denotes a vector in D-dimensional feature space and a set of parameters of the discriminant function, respectively. For an input vector $\mathbf{x}$, if the following equation holds

$$g_k(\mathbf{x}; \Lambda_k) \geq g_i(\mathbf{x}; \Lambda_i) \ \text{ for all } i \neq k \tag{1}$$

then $\mathbf{x}$ is classified to class Ω_k.

In the framework of MCE learning, misclassification measure for class Ω_k is defined as follows

$$d_k(\mathbf{x}) = -g_k(\mathbf{x}; \Lambda_k) + \left[\frac{1}{C-1} \sum_{j, j \neq k} g_j(\mathbf{x}; \Lambda_j)^\eta \right]^{1/\eta} \tag{2}$$

where C represents the number of classes and η is a positive constant. In an extreme case where η goes to infinity, the misclassification measure becomes

$$d_k(\mathbf{x}) = -g_k(\mathbf{x}; \Lambda_k) + \max_{i \neq k} g_i(\mathbf{x}; \Lambda_i). \tag{3}$$

Obviously $d_k(\mathbf{x}) \leq 0$ in case of correct classification, and $d_k(\mathbf{x}) > 0$ in case of misclassification.

Using the misclassification measure for a set of training data $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_P\}$, the objective function to be minimized is defined as an empirical average cost function as given below

$$L_0(\Lambda | X) = \frac{1}{P} \sum_{p=1}^{P} \sum_{k=1}^{C} \ell(d_k(\mathbf{x}_p)) 1(\mathbf{x}_p \in \Omega_k). \tag{4}$$

Here $\Lambda = \{\Lambda_1, \Lambda_2, \cdots, \Lambda_C\}$ and $\ell(d)$ is a smooth loss function, for which the following sigmoid function is typically used

$$\ell(d) = \frac{1}{1 + e^{-\xi(d+\theta)}}. \tag{5}$$

1( ) in (4) is an indicator function which has value of one when the argument is true and zero otherwise.

In order to minimize the objective function of (4), the well-known *gradient descent method* can be applied and the set of parameter of each discriminant function is adapted by the following rule:

$$\Lambda^{(t+1)} = \Lambda^{(t)} - \varepsilon \nabla L_0(\Lambda^{(t)}|X) \tag{6}$$

where $\Lambda^{(t)}$ denotes the parameter set at the t-th iteration and ε denotes the learning parameter of a positive small value.

Instead of using the parameter updating rule of (6), Juang and Katagiri showed another updating rule called Generalized Probabilistic Descent (GPD) which is given by

$$\Lambda^{(t+1)} = \Lambda^{(t)} - \varepsilon_t U \nabla \ell(d_k(\mathbf{x})). \tag{7}$$

Here U is a positive-definite matrix and ε_t is a small positive real number. Compared to the updating rule of (6) that tries to minimize the empirical average cost of (4), (7) is expected to minimize the expected cost of the following equation

$$L(\Lambda) = E[\ell(d(\mathbf{x}))] = \sum_k P(C_k) \int \ell(d_k(\mathbf{x})) p(\mathbf{x}|C_k) d\mathbf{x}. \tag{8}$$

Here $P(C_k)$ and $p(\mathbf{x}|C_k)$ are the a priori and conditional probabilities, respectively. The convergence to a local minimum by the rule (7) is guaranteed when an infinite sequence of random observation $\{\mathbf{x}\}$ are presented during training and the conditions $\sum_{t=1}^{\infty} \varepsilon_t \leftarrow \infty,\ \sum_{t=1}^{\infty} \varepsilon_t^2 < \infty$ are satisfied.

3 Modification of the MCE Learning

In any real-world pattern classification problems, the number of training samples available is finite and relatively small, and the MCE/GPD learning described in the previous section basically tries to minimize an empirical error [8]. Therefore, the MCE learning scheme suffers from the problem of over-fitting to the training dataset as it is with other training schemes.

In order to prevent the over-fitting effect and improve generalization performance, McDermott and Katagiri [5] proposed a method to adapt the slope parameter ξ in (5), which is expected to control the sensitivity of forming the decision boundary against the distribution of training data. In other words, as the parameter ξ increases, the sensitivity increases and the number of training patterns that dominate the shape and location of the boundary becomes fewer. In this sense, the parameter ξ influences the generalization performance of the discriminant functions. One of the drawbacks of this approach is the relationship between ξ and the shape of decision boundary in the feature space is not clear because it is not the shape of decision boundary but the sharpness of the sigmoid function of the distortion measure that ξ controls.

From the view point of generalization, the mapping function from input to output that the recognizer tries to learn should be, in some sense, smooth. In other words, a small change in the inputs should produce a small change in the outputs. This assumption of smoothness as a priori knowledge is natural in case of real-world pattern recognition problems such as character recognition and speech recognition. Based on this assumption, we propose a new method to improve the generalization performance of the MCE learning. Basic idea is to utilize a regularization technique instead of the original definition. In the framework of regularization, the new objective function $\tilde{L}(\Lambda)$ has the form

$$\tilde{L}(\Lambda|X) = L_0(\Lambda|X) + \gamma F(\Lambda), \tag{9}$$

where F is the penalty term for adding smoothness to the discriminant functions, and the parameter γ controls the extent to which the penalty term F influences the form of the solution.

Regularization has been widely applied in the field of image restoration and neural networks. In contrast to the case specific regularizers proposed so far, we employ the so called Tikhonov regularizers [7] for our purpose. This is due to the fact that the MCE/GPD learning is a general learning scheme that is applicable to any first order differentiable discriminant functions, and therefore the regularizer should not be case specific.

The class of Tikhonov regularizers has the form

$$F = \frac{1}{2}\sum_{r=0}^{R}\int_a^b h_r(x)\left(\frac{d^r y}{dx^r}\right)^2 dx \tag{10}$$

in which x, y denote the input, output variable, respectively, and $h_r(x) \geq 0$ for $r = 0, \ldots, R-1$ and $h_R(x) > 0$.

In the present study, as a simple case of the Tikhonov regularizer, we have employed the following empirical penalty term given in [9][10], which is

$$F(\Lambda|X) = \frac{1}{2P}\sum_{k=1}^{C}\sum_{p=1}^{P}\sum_{i=1}^{D}\left(\frac{\partial^2 g_k(\mathbf{x}_p)}{\partial x_{pi}^2}\right)^2 \tag{11}$$

where $\mathbf{x}_p = (x_{n1}, x_{n2}, \ldots, x_{nD})$ represents the p-th training data in D dimensional space. The parameter updating rule of (6) is now

$$\Lambda^{(t+1)} = \Lambda^{(t)} - \varepsilon\nabla\tilde{L}(\Lambda^{(t)}|X). \tag{12}$$

The MCE learning algorithm based on the proposed criterion will be referred as *mMCE* in the following text.

4 mMCE based Neural Networks

The modified MCE learning criterion given in (9) can be applied to arbitrary discriminant functions that are second order differentiable in respect to the variables of the functions. In the present study, multi-layer perceptron type neural network is employed to evaluate the performance.

For the p-th training data $\mathbf{x}_p \in R^D$, let $i_{pj}^{(m)}$ and $o_{pj}^{(m)}$ be the input and output of the j-th cell of layer m respectively. Then the input value of the j-th cell of layer m is given by

$$i_{pj}^{(m)} = \sum_{i=1}^{n_{m-1}} w_{ji}^{(m,m-1)} o_{pi}^{(m-1)} + \theta_j^{(m)}. \tag{13}$$

Here $w_{ji}^{(m,m-1)}$ is the connection weight between the j-th cell of layer m and the i-th cell of layer $m-1$, $\theta_j^{(m)}$ is a constant and n_m represents the number of cells in layer m. The output of each cell is given by

$$o_j^{(m)} = f(i_j^{(m)}) \tag{14}$$

where $f(\ )$ is a sigmoid function of the form

$$f(x) = \frac{1}{1+e^{-x}}. \tag{15}$$

In the framework of the classical error back-propagation (EBP) [11], the object function is defined on the basis of least squared error (LSE)

$$E_{sq} = \frac{1}{2}\sum_{p=1}^{P}\sum_{k=1}^{n_3}\left(t_{pk} - o_{pk}^{(3)}\right)^2, \tag{16}$$

in which three-layer network is assumed and t_{pk} is the desired output (teacher) for the k-th output cell against the p-th input $\mathbf{x}_p$.

On the other hand, in the proposed mMCE, the objective function is defined as

$$\tilde{L}(\Lambda|X) = \frac{1}{P}\sum_{p=1}^{P} L_{0p}(\Lambda|X) + \gamma\frac{1}{P}\sum_{p=1}^{P}\sum_{i=1}^{n_1} F_{pi}(\Lambda|X) \tag{17}$$

where

$$L_{0p}(\Lambda) = \sum_{i=1}^{n_3} \ell(d_i(\mathbf{x}_p))1(\mathbf{x}_p \in C_i), \tag{18}$$

$$F_{pi}(\Lambda) = \frac{1}{2}\sum_{k=1}^{n_3}\left(w_{kj}^{(32)}(w_{ji}^{(21)})^2 f''(i_{pj}^{(2)})\right)^2. \tag{19}$$

The weight adjustment $\Delta w_{pij}^{(m,m-1)}$ corresponding to $\nabla\tilde{L}$ in (12) is

$$\Delta w_{pij}^{(m,m-1)} = \frac{\partial L_{0p}(\Lambda)}{\partial w_{ij}^{(m,m-1)}} + \gamma\frac{\partial F_p}{\partial w_{ij}^{(m,m-1)}}. \tag{20}$$

In the output layer where $m = 3$,

$$\frac{\partial L_{0p}}{\partial w_{kj}^{(32)}} = \ell'(d_k(\mathbf{x}_p)) \frac{\partial d_k(\mathbf{x}_n)}{\partial i_{pk}^{(3)}} o_{pj}^{(2)} 1(\mathbf{x}_p \in C^k), \tag{21}$$

$$\frac{\partial F_{pi}}{\partial w_{kj}^{(32)}} = \frac{1}{2} w_{ji}^{(21)} f''(i_{pj}^{(2)}) Q_{pki} \tag{22}$$

where

$$Q_{pki} = \sum_{j'=1}^{n_2} w_{kj'}^{(32)} {w_{j'i}^{(21)}}^2 f''(i_{pj'}^{(2)}). \tag{23}$$

In the hidden layer where $m = 2$,

$$\frac{\partial L_{0p}}{\partial w_{ji}^{(21)}} = \sum_{k=1}^{n_3} \left(\frac{\partial L_p}{\partial i_{pk}^{(3)}} w_{kj}^{(32)} \right) \frac{\partial \ell(d_j(i_{pj}^{(2)}))}{\partial i_{pj}^{(2)}} o_{pi}^{(1)}, \tag{24}$$

$$\begin{aligned} \frac{\partial F_{pi}}{\partial w_{ji'}^{(21)}} = \frac{1}{2} \Big(& 2\delta_{ii'} f''(i_{pj}^{(2)}) w_{ij}^{(21)} + i_{pi'}^{(1)} {w_{ij}^{(21)}}^2 \Big[(1 - 2f(i_{pj}^{(2)})) f''(i_{pj}^{(2)}) \\ & -2f'(i_{pj}^{(2)})^2 \Big] \Big) \sum_{k=1}^{n_3} w_{kj}^{(32)} Q_{pki}. \end{aligned} \tag{25}$$

It can be seen in the above formulation that the weight adaptation takes place backward from the output layer to the input layer.

5 Experiments

Performance evaluation was conducted on several types of datasets in UCI machine learning repository [12] and ATR speech database [13].

In order to compare the performance of the proposed method with other learning algorithms, the EBP based neural networks, the original MCE based neural networks, and Bayes quadratic discriminant functions where a single Gaussian distribution (full covariance) is assumed for each category were applied on the same datasets.

Since the MCE and mMCE learning are computationally expensive, the initial parameters used in the parameter updating rule of (6) were set to the one obtained by the LSE based EBP learning.

Three-layer feed-forward neural networks were employed for the experiments, the parameter γ in (9) was set to 0.01 and the slope parameter ξ in (5) was set to 1.0.

In case where the absolute recognition performance of the recognizer is an important topic to discuss, one has to pay careful attention in choosing the parameters of neural networks such as the number of nodes in the hidden-layer and learning parameters. However, since the purpose of our experiment is to see how the proposed method improves the generalization performance of the original MCE learning, optimization of the network architecture and learning parameters is not very important.

Table 1. Performance comparison in two-class problems

Method		Data set		
		Cancer	House	Sonar
	# classes	2	2	2
	# training data	420	265	141
	# test data	279	170	67
	# attributes	9	15	60
	# hidden units	12	12	12
Bayes/ML	training	95.0	98.8	100.0
NN/EBP		91.9	96.3	95.0
NN/MCE		93.6	97.4	92.9
NN/mMCE		95.0	94.3	91.5
Bayes/ML	testing	95.7	96.4	74.6
NN/EBP		90.3	96.5	82.1
NN/MCE		94.3	95.3	85.1
NN/mMCE		95.7	97.7	89.6

A. Results for Two-Class Problems

Experiments were, at first, performed for two-class problems on the UCI datasets "cancer", "house" and "sonar". Each dataset was divided into two groups, one was used for training and the other was used for testing.

The experimental results (correct classification rates ([%])) are summarized in Table 1. It can be seen that mMCE gives the best test-set performance among the three methods for each dataset. Compared to the performance improvements from MCE to mMCE for the training set and testing set, the improvement on the training set is larger than that of the testing set. This certifies that the proposed penalty term of (11) is effective for improving the generalization performance of the recognizer.

Fig. 1 shows the learning curves of the loss function L_0, the penalty function F, and the mMCE's total loss function L in (9). Fig. 2 shows the correct classification rates in terms of the slope parameter ξ in (5). Although ξ influences the correct classification rate, mMCE performs better than MCE for any value of ξ. This shows the proposed approach is more effective than the McDermott's approach [5] discussed in Section 3.

B. Results of Multi-Class Problems

In order to evaluate the performance on different datasets, speech database "isolet" (isolated alphabet letters) of the UCI repository, and "vowels" (Japanese five vowels) made from the ATR continuous speech database "Set-B" were collected. In the "isolet" database, the data file "isolet1+2+3+4" was used for training and "isolet5" was used for testing. The database "vowels" was created for this research purpose by extracting 100 samples of each vowel uttered by each subject

Fig. 1. Learning curves of the loss $L_0, \tilde{L}$ and the penalty F in terms of training epochs (dataset: house)

from the ATR database containing the uttered voice of six subjects. The dataset was divided into three groups so that each group contains data of two subjects. Among these three groups, two groups were used for training and the remaining one was used for testing. All of the possible combinations (in this case, 3) were employed for both training and testing.

Table 2 shows the correct classification rate for both the training and test sets. The proposed mMCE gives better classification performance than the original MCE for the test sets.

6 Conclusion

Improvement of generalization performance of the MCE/GPD learning is proposed by employing a regularizer to the objective function to be minimized. Since the employed regularizer is not case specific but general, apart from neural networks the proposed modified MCE (mMCE) learning can be applied to various type of recognizers like HMM (hidden Markov models) and so on.

Acknowledgment

The authors would like to thank Dr. Kanad Keeni for the discussion on training neural networks.

References

1. B-H. Juang and S. Katagiri. Discriminative learning for minimum error classification. *IEEE Trans. Signal Processing*, 40(12):3043–3054, 1992.

Fig. 2. Classification performance for the test set "house" as a parameter of ξ

2. S. Amari. A theory of adaptive pattern classifiers. *IEEE Trans. Elec. Comput.*, EC-16(3):299–307, 1967.
3. T. Kohonen. Learning Vector Quantization. Technical Report TKK-F-A601, Helsinki University of Technology, Laboratory of Computer and Information Science, 1978.
4. E. Oja et al. The ALSM Algorithm – an Improved Subspace Medhotd of Classification. *Pattern Recognition*, 16(4):421–427, 1983.
5. Eric McDermott and Shigeru Katagiri. Prototype-based minimum classification error / generalized probabilistic descent training for various speech units. *Computer Speech and Language*, pages 351–368, August 1994.
6. Biing-Hwang Juang, Wu Chooud, and Chin-Hui Lee. Minimum classification error rate methods for speech recognition. *IEEE Trans. Speech and Audio Processing*, 5(3):257–265, 1997.
7. A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems.* V. H. Winston, 1977.
8. Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.
9. Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.
10. Christopher M. Bishop. Curvature-Driven Smoothing: A Learning Algorithm for Feed-forward Networks. *IEEE Trans. Neural Networks*, 4(5):882–884, 1993.
11. D. E. Rumelhart, G. E. Hinton, and R. J. Willams. Learning representations by back-propagation errors. *Nature 323 9*, 323(9):533–536, October 1986.
12. C. J. Merz and P. M. Murphy. UCI repository of machine learning databases, 1996. http://www.ics.uci.edu/∼mlearn/MLRepository.html.
13. H. Kuwabara, K. Takeda, Y. Sagisaka, S. Katagiri, S. Morikawa, and T. Watanabe. Construction of a large-scale Japanese speech database and its management system. *Proc. of Intl. Conferece on Acoustics, Speech, and Signal Processing (ICASSP-89)*, pages 560–563, 1989.

Fig. 3. Classification performance for the test set "house" as a parapmeter of γ

Table 2. Performance comparison in multi-class problems

		Data set	
		isolet(UCI)	vowels(ATR)
	# classes	26	5
	# training data	6238	4000
	# test data	1559	1000
	# attributes	617	12
Method	# hidden units	32	12
Bayes/ML	training	-	86.3
NN/EBP		89.0	87.3
NN/MCE		95.9	88.3
NN/mMCE		95.5	87.0
Bayes/ML	testing	-	73.0
NN/EBP		93.3	81.8
NN/MCE		94.8	86.4
NN/mMCE		95.3	87.8