INCREASING VIRTUAL SAMPLES THROUGH LOSS SMOOTHNESS DETERMINATION IN LARGE GEOMETRIC MARGIN MINIMUM CLASSIFICATION ERROR TRAINING

Tsukasa Ohashi[1]*, Hideyuki Watanabe*[2]*, Jun'ichi Tokuno*[1]*, Shigeru Katagiri*[1]*, Miho Ohsaki*[1]*, Shigeki Matsuda*[2]*, and Hideki Kashioka*[2]

[1] Graduate School of Engineering, Doshisha University
1-3 Tatara Miyakodani, Kyotanabe-shi, Kyoto 610-0394, Japan.
[2] National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan.
E-mail: hideyuki.watanabe@nict.go.jp

ABSTRACT

We propose a new method for automatically determining the smoothness of smooth classification error count loss for the recent Large Geometric Margin Minimum Classification Error (LGM-MCE) training. The method uses the Parzen-estimation-based formalization of MCE training, and it realizes the determination through the maximum likelihood estimation of error count risk in the one-dimensional geometric-margin-based misclassification measure. In the LGM-MCE framework, increase in the loss smoothness directly leads to an effect of producing virtual samples, which are expected to increase the training robustness to unseen samples. Focusing on this point, we also theoretically clarify the mechanism of this virtual sample generation. Through experiments, the utility of the proposed smoothness determination method is demonstrated, and the mechanism of producing virtual samples and its effect in robustness increase are also clearly illustrated.

Index Terms— geometric margin, Minimum Classification Error, loss smoothness, virtual samples, Parzen estimation.

1. INTRODUCTION

In Minimum Classification Error (MCE) training [1], the smoothness of smooth classification error count loss plays a key role in not only enabling the use of handy gradient-descent-based optimization methods but also increasing the training robustness to unseen samples. However, to increase the training robustness effectively, an appropriate setting of the smoothness degree is required. To this problem, a theoretically-grounded solution was proposed for the MCE training using a conventional functional-margin (FM)-based misclassification measure [2]. It determines the smoothness using Parzen-kernel-based error probability estimation in the FM-based misclassification measure space, and its utility was shown through systematic evaluation experiments [2].

In parallel to the advent of this smoothness determination method, a new MCE training method was developed using a geometric-margin-based misclassification measure [3]. Geometric margin (GM) is the distance between a classification boundary and its nearest training sample in a sample space. This new MCE training, referred to as Large Geometric Margin Minimum Classification Error (LGM-MCE) training, was designed to maximize GM as well as to minimize the smooth classification error count loss. Its superiority to the previous MCE, i.e., Functional-Margin MCE (FM-MCE), was also successfully demonstrated through experiments.

It is clearly worth incorporating the smoothness determination method in the LGM-MCE method. Motivated by this concern, in this paper, we propose a new training method that applies the Parzen-estimation-based loss smoothness determination mechanism to LGM-MCE training. Importantly, increase in the loss smoothness directly leads to increase in the geometric margin, which can be considered as an effect of producing virtual samples in a sample space. The effect of virtual samples is found in the literature (e.g., [4] [5]). Therefore, it is also worth investigating this effect, aiming to clarify the mechanism of robustness increase. In the paper, we thus analyze this issue and theoretically clarify how loss smoothness in the one-dimensional GM-based misclassification measure space produces virtual training samples, which simulate future unseen samples, in a usually high-dimensional sample space.

Through comparative experiments, the effect of the proposed smoothness determination method is clearly demonstrated, and the effect of producing virtual samples using loss smoothness is also excavated.

2. PARZEN-KERNEL-BASED LOSS SMOOTHNESS DETERMINATION

2.1. LGM-MCE Formalization Based on Parzen Estimation of Error Count Risk

First, we newly introduce a formalization of LGM-MCE using the Parzen estimation of error count risk.

We consider the task of classifying input pattern $\boldsymbol{x} \in \mathcal{X}$ as one of the J classes ($C_j; j = 1, ..., J$), where $\mathcal{X}$ denotes the input pattern sample space. As with the previous MCE framework, LGM-MCE training adopts the following classification decision rule based on discriminant functions:

$$C(\boldsymbol{x}) = C_k \quad \text{iff} \quad k = \arg\max_j g_j(\boldsymbol{x}; \Lambda), \tag{1}$$

where $g_j(\boldsymbol{x}; \Lambda)$ is the discriminant function of C_j that indicates the degree to which $\boldsymbol{x}$ belongs to C_j. Λ denotes the trainable parameter set of the classifier and $g_j(\boldsymbol{x}; \Lambda)$ $(j = 1, ..., J)$ is assumed to be differentiable in $\boldsymbol{x}$ and Λ.

Assume here that $\boldsymbol{x}$, which belongs to C_y, is a correctly classified training sample near the classification boundary. The LGM-MCE training focuses on Euclidean distance r between $\boldsymbol{x}$ and the boundary, which is the geometric margin (GM). Increasing the r

This work was supported in part by Grant-in-Aid for Scientific Research (B), No. 22300064.

value raises the possibility of the correct classification of easy-to-misclassify unseen samples. Based on the result of [3], r is represented as

$$r \approx \frac{-d_y(\boldsymbol{x};\Lambda)}{\|\nabla_{\boldsymbol{x}} d_y(\boldsymbol{x};\Lambda)\|}, \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean norm and $d_y(\boldsymbol{x};\Lambda)$ is the following FM-based misclassification measure defined in the previous MCE framework:

$$d_y(\boldsymbol{x};\Lambda) = -g_y(\boldsymbol{x};\Lambda) + \log\left[\frac{1}{J-1}\sum_{j,j\neq y} e^{\psi g_j(\boldsymbol{x};\Lambda)}\right]^{1/\psi}, \tag{3}$$

where $\psi > 0$. That is, the GM is (approximately) equal to the sign-reversed FM-based misclassification measure normalized by the norm of its gradient. The LGM-MCE training then adopts the following new misclassification measure, $D_y(\boldsymbol{x};\Lambda)$, that corresponds to the sign-reversed GM:

$$D_y(\boldsymbol{x};\Lambda) = \frac{d_y(\boldsymbol{x};\Lambda)}{\|\nabla_{\boldsymbol{x}} d_y(\boldsymbol{x};\Lambda)\|} \approx -r. \tag{4}$$

Note that $D_y(\boldsymbol{x};\Lambda)$ not only represents the GM but also shares a feature with the original FM-based misclassification measure $d_y(\boldsymbol{x};\Lambda)$; the positive and negative values of $D_y(\boldsymbol{x};\Lambda)$ imply misclassification and correct classification, respectively.

Similar to the original MCE framework, the LGM-MCE method employs the following smooth classification error count loss for input $\boldsymbol{x}$ that belongs to C_y:

$$\ell_y\big(D_y(\boldsymbol{x};\Lambda)\big) = \frac{1}{1+\exp\big(-\alpha_y D_y(\boldsymbol{x};\Lambda)\big)}, \tag{5}$$

where α_y is a positive number. The LGM-MCE method then minimizes the following empirical average loss using the smooth loss function and the finite training sample set $\Omega_N = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$:

$$L(\Lambda) = \frac{1}{N}\sum_{n=1}^{N} \ell_{y_n}\big(D_{y_n}(\boldsymbol{x}_n;\Lambda)\big), \tag{6}$$

where $\boldsymbol{x}_n \in \mathcal{X}$ is the n-th training sample and $y_n (= 1, ..., J)$ is the class label for $\boldsymbol{x}_n$. Minimizing $L(\Lambda)$ leads to the direct minimization of the classification error counts as well as the enhancement of GM since $\ell_y(D_y)$ is a monotone increasing function of D_y.

Recall that empirical average loss $L(\Lambda)$ is a practical approximation to the following classification error probability over whole pattern sample space, i.e., error count risk:

$$R(\Lambda) = \sum_{y=1}^{J} P(C_y) \int_{\mathcal{X}} 1\Big(D_y(\boldsymbol{x};\Lambda) > 0\Big) p(\boldsymbol{x}|C_y) \mathrm{d}\boldsymbol{x}. \tag{7}$$

Note that, in the LGM-MCE formalism, this risk defined in sample space $\mathcal{X}$ is rewritten as the following form defined in the one-dimensional, GM-based misclassification measure space:

$$R(\Lambda) = \sum_{y=1}^{J} P(C_y) \int_0^{\infty} p(z|C_y) \mathrm{d}z, \tag{8}$$

where z denotes a point in the misclassification measure space. Then the recent MCE formalization, introduced in [6], provided a procedure to estimate probability density function (pdf) $p(z|C_y)$ using the Parzen estimation. The resulting estimate is given as follows:

$$\widehat{p}(z|C_y) = \frac{1}{N_y}\sum_{k=1}^{N_y} \frac{1}{h_y}\phi\Big(\frac{z - D_y(\boldsymbol{x}_k^y;\Lambda)}{h_y}\Big), \tag{9}$$

1. Initialize $h^{(0)} > 0$ and let $\ell = 0$.
2. (E step) Compute the following $q_{m,n}$ ($n = 1, ..., N_y$; $m = 1, ..., N_y, \neq n$):
$$q_{m,n} = \frac{\exp\Big(-\frac{1}{2}\Big\{\frac{z_n - z_m}{h^{(\ell)}}\Big\}^2\Big)}{\sum_{k\neq n}^{N_y}\exp\Big(-\frac{1}{2}\Big\{\frac{z_n - z_k}{h^{(\ell)}}\Big\}^2\Big)}.$$
3. (M step) Re-estimate h as follows:
$$h^{(\ell+1)} = \sqrt{\frac{1}{N_y}\sum_{n=1}^{N_y}\sum_{m\neq n}^{N_y} q_{m,n}(z_n - z_m)^2}.$$
4. Stop the iteration if $h^{(\ell+1)}$ meets a convergence condition; otherwise, let $\ell \leftarrow \ell + 1$ and go to Step 2.
5. Output $h_y = h^{(\ell+1)}$.

Fig. 1. EM algorithm for ML estimation of Gaussian-type Parzen kernel width h_y in class C_y, where $z_n = D_y(\boldsymbol{x}_n^y;\Lambda)$ ($n = 1, ..., N_y$) and $q_{m,n}$ is posterior probability (responsibility).

where $\boldsymbol{x}_k^y$ is the k-th training sample belonging to C_y, $\phi(\cdot)$ is a Parzen kernel in the misclassification measure space, and h_y is its width for C_y.

Note that each Parzen kernel generates virtual samples around its corresponding training sample in the misclassification measure space. Accordingly, the minimization of the following risk estimate, which is given by replacing $p(z|C_y)$ in (8) with $\widehat{p}(z|C_y)$, leads to the status of Λ that approximately corresponds to minimum risk:

$$R_N(\Lambda) = \frac{1}{N}\sum_{y=1}^{J}\sum_{k=1}^{N_y}\int_0^{\infty} \frac{1}{h_y}\phi\Big(\frac{z - D_y(\boldsymbol{x}_k^y;\Lambda)}{h_y}\Big)\mathrm{d}z. \tag{10}$$

Here, $L(\Lambda)$ of (6) coincides with $R_N(\Lambda)$ of (10) when loss $\ell_y(\cdot)$ is redefined as

$$\ell_y\big(D_y(\boldsymbol{x}_k^y;\Lambda)\big) = \int_0^{\infty} \frac{1}{h_y}\phi\Big(\frac{z - D_y(\boldsymbol{x}_k^y;\Lambda)}{h_y}\Big)\mathrm{d}z. \tag{11}$$

In addition, when using Gaussian-type Parzen kernel $\phi(u) = (1/\sqrt{2\pi})\exp(-u^2/2)$ and setting smoothness control parameter α_y as $\alpha_y = 4/(\sqrt{2\pi}h_y)$, $\ell_y(\cdot)$ of (11), which reduces to the cumulative Gaussian distribution function, closely resembles $\ell_y(\cdot)$ of (5) (logistic function), as described in Chapter 4 of [7]. Note the direct link between loss smoothness α_y and Parzen kernel width h_y. Thus, if Parzen estimate $\widehat{p}(z|C_y)$ accurately approximates the true pdf $p(z|C_y)$, $R_N(\Lambda)$ becomes closer to true risk $R(\Lambda)$ of (8), and empirical average loss $L(\Lambda)$ also approaches $R(\Lambda)$.

2.2. Parzen Kernel Width Estimation for Loss Smoothness Determination

A trainable parameter for Parzen estimate $\widehat{p}(z|C_y)$ is kernel width h_y. To find a desirable status of h_y for a Gaussian-type kernel, a cross-validation Maximum Likelihood (ML) method was proposed for the FM-MCE method [2]. Its procedure is summarized in Fig. 1. The width estimation was conducted in a class-by-class mode. To set initial value $h_y^{(0)}$, for example, the interquartile range (IQR)-based method was applicable [8]. Finally, the loss of (5) was used by computing $\alpha_y = 4/(\sqrt{2\pi}h_y)$.

To the LGM-MCE case, this cross-validation ML method is straightforwardly applied.

3. LOSS SMOOTHNESS EFFECT IN SAMPLE SPACE

Parzen-kernel-based formalization illustrates that loss smoothness produces virtual samples in the misclassification measure space. However, a mechanism has not yet been clarified to explain how the smoothness, which is determined in the one-dimensional misclassification measure space, generates virtual samples in (usually high-dimensional) sample space $\mathcal{X}$. Since LGM-MCE's misclassification measure is directly related to the sample space GM, a relationship can perhaps be excavated between the smoothness determination in LGM-MCE and a mechanism that produces virtual samples in a sample space.

Assume that $\boldsymbol{x}_k^y$ is correctly classified by the classifier with parameter set Λ. Consider a set of points where the misclassification measure value is 0, $\mathcal{B}_y(\Lambda) = \{\boldsymbol{x} \in \mathcal{X} | D_y(\boldsymbol{x};\Lambda) = 0\}$, which is a boundary representing whether the patterns are classified as C_y. Furthermore, let $\boldsymbol{x}_k^{y*}$ be a point on $\mathcal{B}_y(\Lambda)$ closest to $\boldsymbol{x}_k^y$ (Fig. 2). $\boldsymbol{x}_k^{y*}$ can be obtained by solving the following constrained minimization problem:

$$\text{minimize}_{\boldsymbol{x}} \ \|\boldsymbol{x} - \boldsymbol{x}_k^y\|^2 \quad \text{subject to } D_y(\boldsymbol{x};\Lambda) = 0. \tag{12}$$

Applying Lagrange's method to the above problem, we get

$$\boldsymbol{x}_k^{y*} - \boldsymbol{x}_k^y = \lambda \nabla_{\boldsymbol{x}} D_y(\boldsymbol{x}_k^{y*};\Lambda), \tag{13}$$

where λ is a positive constant related to the Lagrange multiplier. From (4) and considering $d_y(\boldsymbol{x}_k^{y*};\Lambda) = 0$, we get

$$\nabla_{\boldsymbol{x}} D_y(\boldsymbol{x}_k^{y*};\Lambda) = \frac{\nabla_{\boldsymbol{x}} d_y(\boldsymbol{x}_k^{y*};\Lambda)}{\|\nabla_{\boldsymbol{x}} d_y(\boldsymbol{x}_k^{y*};\Lambda)\|}. \tag{14}$$

$\nabla_{\boldsymbol{x}} D_y(\boldsymbol{x}_k^{y*};\Lambda)$ is a normal unit vector of $\mathcal{B}_y(\Lambda)$ at $\boldsymbol{x}_k^{y*}$. Furthermore, noting that $\|\boldsymbol{x}_k^{y*} - \boldsymbol{x}_k^y\|$ is the GM for $\boldsymbol{x}_k^y$, the direction of vector $\nabla_{\boldsymbol{x}} D_y(\boldsymbol{x}_k^{y*};\Lambda)$ starting from $\boldsymbol{x}_k^y$ approximately coincides with the axis of the GM-based misclassification measure space if $\boldsymbol{x}_k^y$ is close to $\mathcal{B}_y(\Lambda)$. Therefore, from the viewpoint of sample space $\mathcal{X}$, Parzen kernel $(1/h_y)\phi\Big(\{z - D_y(\boldsymbol{x}_k^y;\Lambda)\}/h_y\Big)$ is allocated to $\boldsymbol{x}_k^y$ along this axis (Fig. 2).

If ϕ is Gaussian-type, width h_y plays a role of standard deviation in the GM-based misclassification measure space. Note that the GM value is common to all C_y samples placed parallel to $\mathcal{B}_y(\Lambda)$. Thus, we reach a new finding. The allocation of a Parzen kernel with width h_y in the GM-based misclassification measure space results in virtual samples in the region of sample space $\mathcal{X}$ sandwiched between two hypersurfaces that are $2h_y$ apart and parallel to boundary $\mathcal{B}_y(\Lambda)$, as illustrated in Fig. 2. Furthermore, it can also be shown that loss slant α_y in (5) directly determines, based on $\alpha_y = 4/(\sqrt{2\pi}h_y)$, the value of virtual-sample region $2h_y$ in $\mathcal{X}$ and vice versa.

4. EXPERIMENTAL EVALUATION

4.1. Effect of Smoothness Determination Method in LGM-MCE Training

To evaluate the effect of Parzen-kernel-based loss smoothness determination embedded in LGM-MCE training, we experimentally compared the proposed smoothness determination and conventional empirical smoothness search methods. We employed a fundamental but powerful multi-prototype classifier whose discriminant function was $g_j(\boldsymbol{x};\Lambda) = -\|\boldsymbol{x} - \boldsymbol{p}_j\|^2$, where $\boldsymbol{p}_j$ represented the nearest prototype to $\boldsymbol{x}$ among the prototypes for C_j. Then setting $\psi \to \infty$ in (3), we obtained the following simple but practical misclassification measure:

$$D_y(\boldsymbol{x};\Lambda) = \frac{\|\boldsymbol{x} - \boldsymbol{p}_y\|^2 - \|\boldsymbol{x} - \boldsymbol{p}_i\|^2}{2\|\boldsymbol{p}_y - \boldsymbol{p}_i\|}, \tag{15}$$

where $\boldsymbol{p}_y$ and $\boldsymbol{p}_i$ represented the correct and best-incorrect class prototypes for $\boldsymbol{x}$, respectively.

We used the Letter Recognition Data set consisting of 20,000 English font character image samples [9]. To simulate realistic use situations, we employed the hold-out evaluation manner by dividing the set into a training subset of 1,000 samples, a validation subset of 1,000 samples (used for hyperparameter setting), and a testing subset of the remaining 18,000 samples.

Figure 3 depicts the typical recognition accuracy of the proposed method that automatically estimated α_y's and the accuracy changes of the conventional method with the difference in α_y (common to all classes) for the testing subset. In the proposed method, since the misclassification measure value depends on Λ, unknown pdf $p(z|C_y)$ changes with the progress of the MCE training and the kernel width should thus be repeatedly re-estimated. However, we experimentally confirmed that recognition accuracy was almost independent of how often the kernel width was re-estimated and therefore show the typical result for the proposed method. The figure clearly proves that the proposed method (through automatic determination of α_y) easily achieves recognition accuracy that matches the best accuracy of the conventional method (through a burdensome search for α_y).

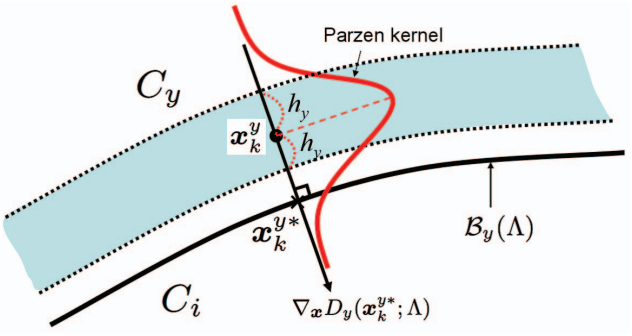

Fig. 2. Parzen kernel and virtual sample distribution in two-dimensional sample space. Virtual samples can exist in the shaded band-like region.

4.2. Effect of Virtual Sample Generation through Loss Smoothing

To evaluate the effect of virtual sample generation through loss smoothing, we compared recognition accuracy between the proposed LGM-MCE method with automatic loss-smoothness determination and the LGM-MCE training with the non-smooth stepwise loss function and with an increased number of real (non-virtual) training samples. Since the stepwise loss is not differentiable, conventional gradient-descent methods are not applicable for cases of increasing real samples. To alleviate this problem, we formalized the line-search-based method, as summarized in Fig. 4. In the figure, $L_\alpha(\Lambda)$ and $L_\beta(\Lambda)$ denote empirical average loss where class-by-class loss slant is set to $\alpha = (\alpha_1, ..., \alpha_J)$ and $\beta = (\beta_1, ..., \beta_J)$, respectively. For considering virtual samples (with smooth loss), we set both α and β to the same vector, of which element values are obtained by the automatic loss-smoothness determination method.

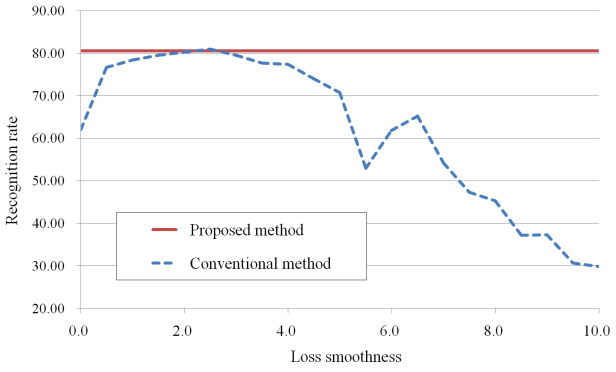

Fig. 3. Result of comparative evaluation on recognition rate between proposed smoothness determination and conventional empirical smoothness search methods.

1. Initialize Λ_0, compute $L_\alpha(\Lambda_0)$, and let $t = 0$.
2. Compute steepest descent direction $\boldsymbol{d}_t = -\nabla_\Lambda L_\beta(\Lambda_t)$.
3. Solve the following line search problem and update Λ_t: $\epsilon_t = \arg\min_{\epsilon>0} L_\alpha(\Lambda_t + \epsilon \boldsymbol{d}_t)$, $\Lambda_{t+1} = \Lambda_t + \epsilon_t \boldsymbol{d}_t$.
4. Compute $L_\alpha(\Lambda_{t+1})$.
5. Stop the iteration if $|L_\alpha(\Lambda_{t+1}) - L_\alpha(\Lambda_t)|$ is sufficiently small, otherwise let $t \leftarrow t + 1$ and go to Step 2.

Fig. 4. Line-search-based method for both smooth and non-smooth error count loss.

For increasing real samples (with non-smooth loss), we set α to ∞, which results in empirical average non-smooth loss. Unlike the smooth loss case, we set another large-valued vector β in Step 2 to compute the descent direction in the case of non-smooth loss.

Unlike the first experiment, we selected from the Letter Recognition data set two data subsets: training and testing. As for the training subset, the number of samples was fixed to 1,000 for the virtual sample case whereas it was changed from 1,000 to 5,000 for the real sample case. The number of testing samples was 10,000 and was common to both cases.

Figure 5 shows typical classification accuracy of the virtual and real sample cases for the testing subset. Comparing the testing-subset rates for the two cases, we observed that the scores in the virtual sample case fell within the range between 3,000 and 4,000 training samples in the real sample case; the effect of loss smoothing was comparable to adding 2,000 or 3,000 real samples. This result demonstrates that loss smoothness determination clearly affects the increasing training samples through virtual sample generation.

5. CONCLUSIONS

We proposed a new classifier training method that applies an automatic loss-smoothness determination method to LGM-MCE training. Through experiment evaluations, we demonstrated that the proposed method easily achieved almost the same recognition accuracy as the best accuracy of the original LGM-MCE method that inevitably involves an empirical and burdensome smoothness search.

Furthermore, we focused on a remarkable effect of loss smoothing, virtual sample generation, in the LGM-MCE training and showed the existence region of virtual samples in an input sample space of arbitrary dimension. The comparative experiment between the smooth loss and the non-smooth loss (with increased real samples) showed that virtual sample generation through loss smoothing improved the training robustness, which was comparable to tripling or quadrupling real samples when using the non-smooth loss function.

We clearly demonstrated the effect of virtual samples for training robustness using a loss-smoothness determination method and also revealed the mechanism that produces virtual samples in a sample space, which will serve as a new basis for further study on training robustness to unseen samples.

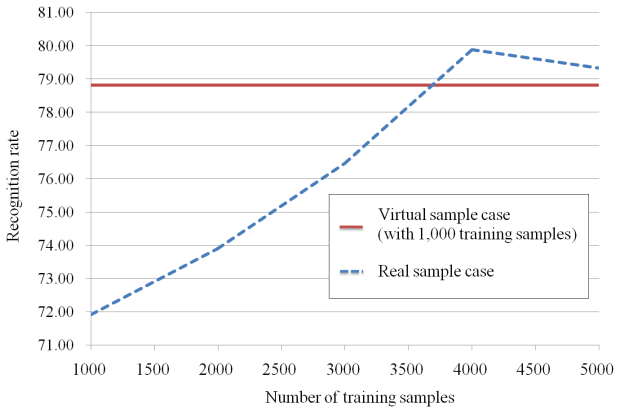

Fig. 5. Comparison of recognition rates between virtual and real sample cases.

6. REFERENCES

[1] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Processing*, vol. 40, no. 12, pp. 3043-3054, Dec. 1992.

[2] H. Watanabe, J. Tokuno, T. Ohashi, S. Katagiri, and M. Ohsaki, "Minimum classification error training with automatic setting of loss smoothness," *MLSP 2011*, Sept. 2011.

[3] H. Watanabe, S. Katagiri, and M. Ohsaki, "Minimum classification error training with geometric margin enhancement for robust pattern recognition," *MLSP 2011*, Sept. 2011.

[4] Y. Lee, J. Kang, B. Kang, and K.R. Ryu, "Bayesian sampling of virtual examples to improve classification accuracy," *ICASE 2006*, pp. 1009-1014, Oct. 2006.

[5] H. Miyao and M. Maruyama "Virtual example synthesis based on PCA for off-line handwritten character recognition," *DAS 2006*, pp. 96-105, 2006.

[6] E. McDermott and S. Katagiri, "A derivation of minimum classification error from the theoretical classification risk using Parzen estimation," *Comput. Speech Lang.*, vol. 18, pp. 107-122, April 2004.

[7] C.M.Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, New York, 2006.

[8] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall/CRC, 1986.

[9] http://archive.ics.uci.edu/ml/