SAMPLE-SEPARATION-MARGIN BASED MINIMUM CLASSIFICATION ERROR
TRAINING OF PATTERN CLASSIFIERS WITH QUADRATIC DISCRIMINANT FUNCTIONS

Yongqiang Wang\textsuperscript{1,2} and Qiang Huo\textsuperscript{1}

\textsuperscript{1}Microsoft Research Asia, Beijing, China
\textsuperscript{2}Department of Computer Science, The University of Hong Kong, Hong Kong, China
(E-mails: yqwang@cs.hku.hk, qianghuo@microsoft.com)

ABSTRACT
In this paper, we present a new approach to minimum classification error (MCE) training of pattern classifiers with quadratic discriminant functions. First, a so-called sample separation margin (SSM) is defined for each training sample and then used to define the misclassification measure in MCE formulation. The computation of SSM can be cast as a nonlinear constrained optimization problem and solved efficiently. Experimental results on a large-scale isolated online handwritten Chinese character recognition task demonstrate that SSM-based MCE training not only decreases the empirical classification error, but also pushes the training samples away from the decision boundaries, therefore a good generalization is achieved. Compared with conventional MCE training, an additional 7\% to 18\% relative error rate reduction is observed in our experiments.

Index Terms— discriminative training, minimum classification error, quadratic discriminant function, sample separation margin

1. INTRODUCTION
The formulation of minimum classification error (MCE) training of pattern classifiers pioneered by Amari [1] and extended by Juang and Katagiri [7] consists of three key steps to define an MCE objective function: 1) the specification of a discriminant function for each class and the corresponding maximum discriminant decision rule, 2) the specification of a misclassification measure for a given training sample, and 3) the specification of a smooth loss function. Combinations of different choices for each of the above three steps and optimization methods lead to various MCE training algorithms. In the past several decades, the power of MCE training has been clearly demonstrated by many research groups for different pattern classifiers in different applications. More recently, a new misclassification measure based on a so-called sample separation margin (SSM) was proposed in [6] for MCE training of piecewise linear classifiers, where each pattern class is represented by a set of prototype vectors, and Euclidean distance is used to measure the dissimilarity between an input feature vector and a prototype. Comparative experiments were conducted on the task of the recognition of isolated online handwritten Japanese Kanji characters using popular Nakayosi and Kuchibue benchmark databases. Experimental results demonstrate that MCE training with the SSM-based misclassification measure achieves significant character recognition error rate reduction compared with MCE training using two traditional misclassification measures. It is empirically observed that the SSM-based MCE training can minimize the empirical classification error while increase the sample separation margin, therefore such trained classifiers have better generalization capability [5]. Encouraged by this promising result, in this study, we extend the SSM-based MCE formulation to training pattern classifiers with quadratic discriminant functions (QDF).

The rest of the paper is organized as follows. In Section 2, we describe briefly the general formulation of MCE training. In Section 3, we introduce the formulation of SSM-based MCE training. In Section 4, we present our solution to SSM-based MCE training of QDF-based classifiers. Experimental results are reported in Section 5. Finally, the paper is concluded in Section 6.

2. GENERAL FORMULATION OF MCE TRAINING
Consider a pattern classification problem with M classes denoted as \(\{C_i|i=1,\ldots,M\}\). Suppose we are given a set of training feature vectors \(X = \{x_r \in R^D | r = 1,\ldots,R\}\) together with their labels \(I = \{i_r | r = 1,\ldots,R\}\). A three-step procedure in [7] can be used to formulate the MCE objective function as follows. Firstly, for each class \(i\), a discriminant function \(g_i(x;\Omega)\) is defined, where \(x\) is an unknown feature vector and \(\Omega\) is the set of classifier parameters. The class with the maximum discriminant score is chosen as the recognized class \(r(x;\Omega)\), i.e.,

\[
r(x;\Omega) = \arg\max_i g_i(x;\Omega).
\]

This is also known as the maximum discriminant decision rule.

The second step is to define an appropriate misclassification measure \(d(x_r,i_r;\Omega)\) for each training sample \(x_r\). The sign of \(d(x_r,i_r;\Omega)\) shall indicate whether the decision \(r(x_r;\Omega)\) is correct, while the absolute value of \(d(x_r,i_r;\Omega)\) represents the decision confidence. A popular choice of misclassification measure is as follows (e.g., [8]):

\[
d(x_r,i_r;\Omega) = -g_{i_r}(x_r;\Omega) + G_{i_r}(x_r;\Omega),
\]

where

\[
G_j(x;\Omega) = \frac{1}{\eta} \log \left[ \frac{1}{|M(x;j)|} \sum_{n \in M(x;j)} \exp[\eta g_n(x;\Omega)] \right].
\]

In the above equation, \(\eta\) is a smoothing parameter; \(M(x;j)\) is the alternative hypothesis space for \(x\), excluding the true label \(j\); and \(|S|\) denotes the cardinality of set \(S\).

The third step is to use a smooth and differentiable function to approximate the 0-1 loss incurred by making the decision \(r(x_r;\Omega)\). A popular choice is the following sigmoid function [7]:

\[
\ell(x_r,i_r;\Omega) = \frac{1}{1 + \exp[-\alpha d(x_r,i_r;\Omega) + \beta]}.
\]
where $\alpha$ and $\beta$ are two control parameters. The parameters $\Omega$ are chosen by minimizing the empirical risk function $\ell(X, \mathcal{I}, \Omega) = \frac{1}{R} \sum_{r=1}^{R} \ell(x_r, i_r; \Omega)$.

3. FORMULATION OF SSM-BASED MCE TRAINING

In [6], a so-called “sample separation margin (SSM)” is defined for piecewise linear classifiers. Here, we generalize the definition of SSM to arbitrary vector-space classifiers. Given a family of discriminant functions $\{g_i(x; \Omega)\}$, the decision boundary between $i$-th and $j$-th class becomes $D_{ij}(\Omega) = \{x | g_i(x; \Omega) - g_j(x; \Omega) = 0\}$. We define the distance from $x$ to the decision boundary $D_{ij}(\Omega)$ as $t(x; D_{ij}(\Omega)) = \min_{\Omega \in D_{ij}(\Omega)} q(x)$, where $q(x)$ can be any metric function measuring the distance between $x$ and $z$. In this work, Euclidean distance, $\|x - z\|$, is used. Moreover, if the label of $x$ is known as $i$, a new term $t(x, i; \Omega)$ is defined as

$$t(x, i; \Omega) = -\frac{1}{\eta} \log \left( \frac{1}{M-1} \sum_{j \neq i} \exp \left( -\eta t(x; D_{ij}(\Omega)) \right) \right) \tag{5}$$

where the hypothesis space $\mathcal{M}(x; i)$ is assumed to include all the possible pattern classes, except $i$. As the smoothing parameter $\eta$ approaches to $+\infty$, $t(x, i; \Omega)$ becomes the minimum distance from $x$ to the decision boundaries $\{D_{ij}(\Omega) \neq \Omega\}$. To allow classification error, we define SSM using $t(x, i; \Omega)$ with a sign to indicate whether $x$ is classified correctly by the decision rule $r(x; \Omega)$, i.e.,

$$\hat{t}(x, i; \Omega) = \begin{cases} t(x, i; \Omega) & \text{if } i = r(x; \Omega), \\ -t(x, i; \Omega) & \text{otherwise.} \end{cases} \tag{6}$$

Given the above SSM, we follow [6] to redefine the misclassification measure $d(x_r, i_r; \Omega)$ as $-\hat{t}(x_r, i_r; \Omega)$. The “per sample loss function” becomes

$$\hat{\ell}_{\text{SSM}}(x_r, i_r; \Omega) = \frac{1}{1 + \exp[\alpha t(x_r, i_r; \Omega) + \beta]} \tag{7}$$

and the empirical risk function on the training data set becomes

$$\hat{\ell}_{\text{SSM}}(X, \mathcal{I}, \Omega) = \frac{1}{R} \sum_{r=1}^{R} \hat{\ell}_{\text{SSM}}(x_r, i_r; \Omega), \tag{8}$$

which is referred to as SSM-based MCE objective function hereinafter.

The effect of minimizing SSM-based MCE objective function is two-fold: (1) decreasing the empirical risk; (2) pushing the training samples away from the decision boundaries. Therefore, our SSM-based MCE training is expected to generalize better than the conventional MCE training. To solve this new MCE training problem, following two issues arise:

- How to calculate the distance from $x$ to the decision boundary $D_{ij}(\Omega)$, i.e., to solve a nonlinear programming problem

$$f(\Omega, x) = \min_{h(x; \Omega) = 0} \|x - z\|^2, \tag{9}$$

where $h(z; \Omega) = g_i(z; \Omega) - g_j(z; \Omega)$;

- How to calculate the gradient of $f(\Omega, x)$ w.r.t. $\Omega$, such that gradient-based algorithms can be employed to optimize the SSM-based MCE objective function in Eq. (8). The second issue can be addressed by using a technique known as sensitivity analysis in nonlinear programming. By directly applying a theorem in [4], we have the following proposition:

**Proposition 1** Let $(z^*, \lambda^*)$ be the pair of local-minimum and Lagrangian multiplier of the problem in Eq. (9) with $\Omega = \Omega_0$, i.e.,

$$z^* = \arg\min_{\Omega = \Omega_0} \|x - z\|^2 + \lambda^* h(z; \Omega_0)$$

and $z^*$ is feasible. If $\nabla_{z^*} h(z^*; \Omega_0) \neq 0$, then there exists a neighborhood around $\Omega_0$ such that $f(\Omega, x)$ is a continuous function, and

$$\frac{\partial f(\Omega, x)}{\partial \Omega} \bigg|_{\Omega = \Omega_0} = \lambda^* \frac{\partial h(z^*; \Omega_0)}{\partial \Omega} \bigg|_{\Omega = \Omega_0}$$

It can be verified that $\nabla_{z^*} h(z^*; \Omega_0) \neq 0$ if and only if $x$ is not on the boundary, i.e., $h(x, \Omega_0) \neq 0$. In next section, we will give detailed formulas for gradient calculation and address the second issue in the context of SSM-based MCE training of QDF-based classifiers.

4. SSM-BASED MCE TRAINING OF QDF-BASED CLASSIFIERS

Assuming that feature vectors belonging to $i$-th class follow a Gaussian distribution with a mean vector $\mu_i$ and a full covariance matrix $\Sigma_i$, and taking the corresponding log-likelihood function as the $i$-th discriminant function $g_i(x; \Omega)$, we have

$$g_i(x; \Omega) = c_i + x^T \Sigma_i^{-1} \mu_i - \frac{1}{2} x^T \Sigma_i^{-1} x + \text{const}, \tag{10}$$

where $c_i = -\frac{1}{2} (\log \det \Sigma_i + \mu_i^T \Sigma_i^{-1} \mu_i)$, and $\Omega = \{\mu_i\} \cup \{\Sigma_i\}$. The discriminant function in Eq. (10) is known as “quadratic discriminant function (QDF)”. Given this QDF and the decision rule in Eq. (1), projecting $x$ onto the decision boundary $D_{ij}(\Omega)$ becomes the following constrained optimization problem:

$$\min_{z} \frac{1}{2} \|x - z\|^2 \tag{11}$$

subject to $\frac{1}{2} z^T P z + \mu^T z + c = 0$,

where $P = \Sigma_i^{-1} - \Sigma_j^{-1} \mu_i \Sigma_j^{-1} \mu_j^T$; $c = c_i - c_j$; and the following condition holds,

$$\min_{z} \frac{1}{2} z^T P z + z^T \mu + c < 0 < \max_{z} \frac{1}{2} z^T P z + z^T \mu + c.$$

If $P = 0$, it is easy to show that the optimal $z^* = x - \lambda^* \mu$ and its corresponding Lagrangian multiplier $\lambda^* = \frac{x^T P x + z^T \mu + c}{\lambda^*}$. If $P \neq 0$, problem in Eq. (11) is a quadratically constrained quadratic programming (QCQP) with a very special structure: a convex objective function with one quadratic equality constraint (but not necessarily convex). Similar problems have been studied in indefinite trust-region methods for nonlinear programming. For example, Stern and Wolkowicz [12] constructed an algorithm which can globally converge to the minimizer of the problem in Eq. (11) provided $\mu = 0$. Unfortunately, it is impossible to eliminate the linear term in the constraint function by methods of changing variable if $P$ is singular. Moré [11] extended this study to the problem with one arbitrary quadratic equality constraint. By summarizing the results in [11, 12], we derive the following proposition:

**Proposition 2** For the problem in Eq. (11), let $J$ denote the set of $\lambda$ such that $I + \lambda P \succeq 0$. If $x - \lambda \mu \notin \mathcal{R}(I + \lambda P)$ when $\lambda \in \mathcal{R}(P)$ stands for the range space of matrix $P$, and $\mathcal{B}(J)$ means the boundary of the set $J$, then there always exists a $\lambda^*$ together with an $z^*$, such that $(I + \lambda^* P) z^* = x - \lambda^* \mu$, $I + \lambda^* P \succeq 0$ and $z^*$ is the global minimizer of the problem in Eq. (11).
Although the conditions in the above proposition seem more restrictive, they can guarantee the uniqueness of the global minimizer. In practice, the chance of violating the conditions is extremely rare. In case that there exist $x$ and $D_r(x)$ which violate the conditions, we just ignore $x$ in calculating the objective function in Eq. (8).

From Proposition 2, it is easy to see $\sum^* = \sum(\lambda^*)$, where $\sum(\lambda^*) = (I + \lambda P)^{-1}(x - \lambda \mu)$. Therefore the problem in Eq. (11) reduces to a univariate root finding problem, i.e., to find a $\lambda^* \in J$, such that

$q(\lambda^*) = \frac{1}{2} \sum^T P \sum(\lambda^*) + \mu^T \sum(\lambda^*) + c = 0$.

Moreover, since $q'(\lambda) < 0$ for all $\lambda \in J$, there will be only one zero in $J$. Therefore a Newton’s algorithm with safeguard is used to solve the above problem.

In addition to solving the problem in Eq. (11) as described above, the following procedure is used to calculate the gradient of SSM-based MCE objective function w.r.t. mean vectors for a training sample $x_r$.

**Step 1:** For each $j$ in the alternative hypothesis space $M(x_r; i_r)$, form the decision boundary $D_{r,j}(\Omega)$. Consequently, we have $|M(x_r; i_r)|$ decision boundaries for each $x_r$.

**Step 2:** For each of these decision boundaries, solve the corresponding nonlinear constrained optimization problem in Eq. (11). Denote $z_{rj}$ the projection of $x_r$ onto the boundary $D_{r,j}(\Omega)$, and $\lambda_{rj}$ the corresponding Lagrangian multiplier.

**Step 3:** By using chain rule and Proposition 1, the gradient of the loss function in Eq. (7) w.r.t. mean vectors can be obtained as follows:

$$\frac{\partial \ell(x_r; i_r; \Omega)}{\partial \mu_i} = \sum_{\Omega \in M(x_r; i_r)} c_{\Omega} \Sigma^{-1} (z^*_{\Omega} - \mu_i),$$

$$\frac{\partial \ell(x_r; i_r; \Omega)}{\partial \mu_j} = c_{\Omega} \Sigma^{-1} (\mu_j - z^*_{\Omega}), \forall j \in M(x_r; i_r),$$

where

$$c_{\Omega} = \frac{\exp(-\eta(x_r; D_{r.n}(\Omega)))}{\sum_{\Omega \neq \Omega' \in M(x_r; i_r)} \exp(-\eta(x_r; D_{r.n}(\Omega)) \ell(x_r; D_{r.n}(\Omega)))} \lambda^*_{\Omega},$$

$$\nu_{\Omega} = \begin{cases} \ell(x_r; i_r; \Omega)(1 - \ell(x_r; i_r; \Omega)) & \text{if } r(x_r; i_r; \Omega) = i_r, \\ -\ell(x_r; i_r; \Omega)(1 - \ell(x_r; i_r; \Omega)) & \text{otherwise} \end{cases}.$$

After going through the above steps for all the training samples, the gradient of the MCE objective function in Eq. (8) w.r.t. mean vectors can be calculated. Although other options exist, we use Quickprop algorithm [3] in this study to update the mean vectors. The detailed procedure of using Quickprop to optimize MCE objective function in a similar context can be found in our previous work [13].

In Chinese handwriting recognition, the following “modified quadratic discriminant function (MQDF)” [9] is considered as the state-of-the-art:

$$g_i(x; \Omega) = -\frac{1}{2} \left( (x - \mu_i)^T U_i (A_i^{-1} - \frac{1}{\delta_i} I) U_i^T (x - \mu_i) + \frac{1}{\delta_i} ||x - \mu_i||^2 + \log \det \Lambda_i + (D - K) \log \delta_i \right),$$

where $U_i$ is a $D \times K$ matrix whose columns are the $K$ leading eigenvectors of $\Sigma_i$ in descending order; $A_i$ is a $K \times K$ diagonal matrix with $d$-th diagonal element being the $d$-th eigenvalue of $\Sigma_i$; $\delta_i$ is a class-dependent constant; and $K$ is a control parameter. The set of classifier parameters $\Omega$ becomes $\{\mu_i\} \cup \{U_i\} \cup \{A_i\} \cup \{\delta_i\}$. $\Sigma_i$ can be easily recovered from $U_i$, $\Lambda_i$ and $\delta_i$ as follows:

$$\Sigma_i = U_i (A_i - \delta_i I) U_i^T + \delta_i I .$$

Since we only update mean vectors in this study, the previous procedure can be directly applied to MCE training of MQDF-based classifiers by using Eq. (14) to calculate $\Sigma_i$ whenever necessary.

## 5. EXPERIMENTS AND RESULTS

### 5.1. Experimental Setup

In order to evaluate the effectiveness of the proposed SSM-based MCE training algorithm, we conduct a series of experiments on the task of the recognition of isolated online handwritten Chinese characters with a vocabulary of 6,763 Chinese characters in GB2312-80 standard. An in-house developed online handwritten Chinese character corpus is used in our experiments. The training set of this corpus contains 9,447,328 character samples. The number of training samples per character class ranges from 952 to 5,600. The testing set contains 614,369 handwriting samples from 6,763 character classes, which is further divided into two subsets according to the writing style (cursive or regular) of each testing sample as follows:

- **Cursive:** 431,546 samples from 3,863 character classes;
- **Regular:** 182,823 samples from 6,763 character classes.

As for feature extraction, a 512-dimensional raw feature vector $y$ is first extracted from each handwriting sample by using the procedure described in [2]. Then we use an LDA transformation matrix estimated from the training data to transform $y$ into a new feature vector $x$ of dimension 128. All of our experiments are conducted on these 128-dimensional feature vectors. To speed up the training process, a multiple-prototype-based pre-classifier [6] is used first to identify a short-list of 50 candidates for each training sample. Then during the MCE training, this short-list will be used as the hypothesis space instead of the whole character set.

In the first set of experiments, we train two MQDF-based classifiers with $K = 5$ and $K = 10$ respectively by using the penalized (or regularized) ML estimation as follows: $\mu_i$ is simply taken as $\mu_i$, which is the sample mean of the training feature vectors from the $i$-th class; the eigenvalues $\Lambda_i$ and eigenvectors $U_i$ are estimated from the $K$ leading eigen-pairs of the regularized sample covariance matrix $\Sigma_i = (1 - \gamma) \Sigma_i + \gamma \mu_i \mu_i^T I$, where $\Sigma_i$ is the sample covariance matrix of the $i$-th class; $\delta_i$ is set as the average of $D - K$ minor eigenvalues of $\Sigma_i$. The regularization coefficient $\gamma$ is set as $\gamma = 0.2$ in both cases.

Starting from the above ML-trained MQDF-based classifiers, we conduct SSM-based MCE training by updating the mean vectors $\mu_i$ only. For each training sample $x_r$, to further save computation, the alternative hypothesis space is set as the most competing class (i.e., with the maximum discriminant score). After computing SSM and SSM-based MCE loss function in Eq. (7) for $x_r$, the gradient of the loss function $\ell(x_r; i_r; \Omega)$ w.r.t. the mean vectors is obtained by Eqs. (12) and accumulated. After one pass of all the training samples (a.k.a. one epoch), the mean vectors are updated by Quickprop algorithm. The control parameters in this set of experiments are set as $\alpha = 5, \beta = 0$. Quickprop is stopped after 20 epochs.

For comparison, we also conduct conventional MCE training by starting from the above ML-trained MQDF-based classifiers. Different from the work in [10], we use Quickprop to update mean vectors only in MCE training. Unlike the procedure in calculating the SSM-based MCE objective function, we use all the competing classes in
Table 1. Comparison of recognition accuracies (in %) of several MQDF-based classifiers trained by different approaches: maximum likelihood training (ML), conventional MCE training (MCE), and SSM-based MCE training (SSM-MCE). “MQDF(K)” means that K eigenvectors are retained.

<table>
<thead>
<tr>
<th>Training Approaches</th>
<th>MQDF(5)</th>
<th>MQDF(10)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Regular</td>
<td>Cursive</td>
</tr>
<tr>
<td>ML</td>
<td>98.27</td>
<td>91.66</td>
</tr>
<tr>
<td>MCE</td>
<td>98.71</td>
<td>92.66</td>
</tr>
<tr>
<td>SSM-MCE</td>
<td>98.81</td>
<td>93.00</td>
</tr>
</tbody>
</table>

Fig. 1. Comparison of histograms of SSMs on training set using SSM-based MCE training vs. conventional MCE training.

5.2. Experimental Results

We summarize the comparison of our SSM-based MCE training with conventional MCE training in Table 1. Our SSM-based MCE training performs consistently better than the conventional MCE training. For example, compared with ML-trained classifiers, after conventional MCE training, MQDF(5) achieves a relative error reduction of 25.4%, whilst SSM-based MCE reduces the error rate by 31.2%. It is also noted that accuracy improvement on “Cursive” set is relatively smaller than on “Regular” set. This may be caused by the fact that there are much more regular samples than cursive samples in our training data set, therefore performance improvement may be biased toward “Regular” set.

To compare the SSMs after discriminative training, we calculate the SSMs of all training samples, using SSM-based MCE-trained and conventional MCE-trained MQDF(10) classifiers, respectively. The distributions (envelops of histograms) of SSMs of two different classifiers are plotted in Figure 1. It is noted that samples are pushed away from the decision boundaries after our SSM-based MCE training, therefore our algorithm has a better generalization capability.

6. SUMMARY

In this paper, we generalize the definition of sample separation margin (SSM) in [6] and apply the SSM-based MCE training to QDF-based classifiers. Computing SSM is formulated as a nonlinear programming problem and an efficient algorithm is provided to solve this problem for QDF-based classifiers. The experimental results confirm that the SSM-based MCE training leads to not only small classification error rate on training set, but also large separation among different classes, thus a better generalization can be expected. As future works, we will extend SSM-based MCE formulation to even more flexible classifiers based on data models such as Gaussian mixture model and hidden Markov model.

7. REFERENCES