Combining Evolution Strategy and Gradient Descent Method for Discriminative Learning of Bayesian Classifiers

Xuefeng Chen Xiabi Liu* Yunde Jia
Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology
Beijing Institute of Technology, Beijing 100081, P.R. China
{crocodel, liuxiabi, jiayunde}@bit.edu.cn

ABSTRACT

The optimization method is one of key issues in discriminative learning of pattern classifiers. This paper proposes a hybrid approach of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) and the gradient decent method for optimizing Bayesian classifiers under the SOFT target based Max-Min posterior Pseudo-probabilities (Soft-MMP) learning framework. In our hybrid optimization approach, the weighted mean of the parent population in the CMA-ES is adjusted by exploiting the gradient information of objective function, based on which the offspring is generated. As a result, the efficiency and the effectiveness of the CMA-ES are improved. We apply the Soft-MMP with the proposed hybrid optimization approach to handwritten digit recognition. The experiments on the CENPARMI database show that our handwritten digit classifier outperforms other state-of-the-art techniques. Furthermore, our hybrid optimization approach behaved better than not only the single gradient decent method but also the single CMA-ES in the experiments.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models – *statistical;* G.1.6 [**Numerical Analysis**]: Optimization – *Global optimization;* I.2.6 [**Artificial Intelligence**]: Learning – *parameter learning*

General Terms

Algorithms

Keywords

Discriminative learning, Evolution strategy, Gradient descent, Handwritten digit recognition, Max-Min Posterior pseudo-probabilities (MMP)

1. INTRODUCTION

In the last two decades, discriminative learning has become a major theme in statistical pattern recognition research and successfully applied in many areas, such as speech and language processing, document analysis and recognition, image retrieval, and so on. The development of discriminative learning methods for statistical pattern recognition involves two crucial issues: design of objective function for optimization and actual optimization method. The traditional optimization methods used in discriminative learning are gradient-based ones. It is well known that the optimization by gradient-based methods tends to get stuck in local optima. In contrast with gradient-based methods, evolutionary algorithms can quickly locate high performance regions in large and complex search spaces. Furthermore, evolutionary algorithms can escape from local optima with a high probability to reach the desired global optima for its multipoint search strategy. Therefore, the recent years have witnessed a growing interest in applying evolution algorithms for global optimization of statistical classifiers, such as neural networks and Support Vector Machines (SVMs). In the field of neural networks, the evolution algorithms have been employed to evolve the connection weights [3, 32], or the neural architecture [19, 20, 25], or the both of them [7, 13, 27]. The applications of evolution algorithms to SVMs include (1) the selection of optimal feature subsets [11], (2) the optimization of parameters [28, 31], (3) the creation of new kernel functions [14], and (4) the multi-objective optimization [18, 24].

In this paper, we explore the combination of the evolution strategy and the gradient decent method under a new discriminative learning framework of Bayesian classifiers, SOFT Target based Max-Min posterior Pseudo-probabilities (Soft-MMP), which was developed by us in the previous work [5]. Soft-MMP has shown its effectiveness in the application to character recognition [5], where the traditional gradient descent optimization algorithm is used. In this work, the Soft-MMP is strengthened by the evolution strategy which is a main branch of evolution algorithms and particularly well suited for real-valued optimization problem encountered in Soft-MMP. We consider the currently popular Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [14, 15, 17, 30] and combine it with the original gradient descent algorithm to improve the effectiveness and efficiency of Soft-MMP learning. The main idea is to adjust the weighted mean of the parent population in each generation of the CMA-ES by using the gradient descent. In this way, the advantages of CMA-ES and gradient descent are merged. On one hand, the local optimum can be escaped by multi-point stochastic search. On the other hand, the convergence speed can be improved by exploiting the gradient information. The combination of CMA-ES and gradient descent has been explored for function optimization by Auger et al [1]. To our best knowledge, the work discussed in this paper is the first one of applying the combination of CMA-ES and gradient descent to

* Corresponding Author. Tel.: +86 10 68913447; Fax: +86 10 86343158.

GECCO'09, July 8–12, 2009, Montréal, Québec, Canada.

discriminative learning. Furthermore, our combination method is different with that developed by Auger et al [1], where a dynamic criterion is designed to choose the covariance matrix update method between that based on a quadratic approximation of the target function and the original CMA. The covariance matrix update method could vary with different target functions. Oppositely, we integrate the gradient descent into the CMA-ES to get a single optimization process which is uniform for all the target functions.

We evaluate the Soft-MMP with the proposed hybrid optimization approach through the experiments of handwritten digit recognition, which were conducted on the well-known CENPARMI database of handwritten digits [29]. In the work of Liu et al [21], the state-of-the-art techniques of handwritten digit recognition, including features and classifiers, is thoroughly investigated on the CENPARMI database. They reported the best error rate of 0.95% on the test set of the CENPARMI database, which is obtained by using 8-direction gradient features (abbreviated to e-grg there) and the classifier of either SVM with RBF kernel or Discriminative Learning Quadratic Discriminant Function. Using the same features by courtesy of Liu under the Soft-MMP framework with the proposed hybrid optimization approach, we achieved the better error rate of 0.75% on the test set. This result also outperforms other up-to-date results reported on the CENPARMI database via various features and classifiers. Furthermore, we compared three optimization schemes for Soft-MMP in the experiments: (1) only gradient descent used, (2) only CMA-ES used, and (3) the hybrid of them. Experimental results show that the CMA-ES behaved better than the gradient descent, but worse than the hybrid one. Compared with the single gradient descent and the single CMA-ES, the hybrid scheme brings more effectiveness and more efficiency. The error rate on the test set is decreased from 1.05% (gradient descent) and 0.85% (CMA-ES) to 0.75% (hybrid), respectively. The average training time is decreased from 2133 seconds (gradient descent) and 2340 seconds (CMA-ES) to 1856 seconds (hybrid), respectively.

The rest of this paper is organized as follows. Section 2 briefly introduces the Soft-MMP learning framework of Bayesian classifiers. Section 3 presents the hybrid optimization approach for Soft-MMP. Section 4 discusses the application of our approach to handwritten digit recognition and the corresponding experimental results. We give our conclusions in Section 5.

2. Soft-MMP

Here we briefly introduce the Soft-MMP learning framework. The reader is referred to our papers for more details [4, 5, 6].

Soft-MMP is developed to learn posterior pseudo-probability based classifiers, a new kind of Bayesian classifiers. Posterior pseudo-probability based classifiers have been successfully applied to text extraction, image retrieval, and digit recognition [4, 5, 6, 23]. Let $\boldsymbol{x}$ be a 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

$$f(p(\boldsymbol{x}|C_i)) = 1 - \exp(-\kappa p^{\beta}(\boldsymbol{x}|C_i)), \tag{1}$$

where κ and β are positive numbers. 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.

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

According to Eq. 1, the posterior pseudo-probability is in direct proportion to class-conditional probability density, so the posterior pseudo-probabilities based classifier is consistent with traditional Bayesian counterpart. However, the values of posterior pseudo-probabilities for making classification decision are in $[0,1]$, by introducing which the discriminative learning approaches such as MMP [4] and Soft-MMP [5] can be developed for Bayesian classifiers. Furthermore, the posterior pseudo-probability is useful for (1) making rejection decision, (2) combining classifiers, and (3) assessing the performance of a classifier in a much more accurate way than that of counting the number of patterns classified correctly [4].

In Soft-MMP learning method [5], two adaptive soft targets of posterior pseudo-probabilities for positive samples and negative samples of each class are defined. Let $\hat{H}$ and $\overline{H}$ be two adaptive soft targets which take values in $[0,1]$, $\boldsymbol{\Lambda}$ be the set of unknown parameters learned by Soft-MMP, $\hat{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$ be the feature vector of arbitrary positive and negative sample of the i -th class, respectively, then the empirical loss of the classifier on positive samples and negative samples of the i -th class are measured as

$$\hat{l}(\hat{\boldsymbol{x}};\boldsymbol{\Lambda}) = \begin{cases} 0 & , \quad f(p(\hat{\boldsymbol{x}}|C_i)) > \hat{H} \\ \hat{H} - f(p(\hat{\boldsymbol{x}}|C_i)), & f(p(\hat{\boldsymbol{x}}|C_i)) \le \hat{H} \end{cases}, \tag{3}$$

and

$$\bar{l}(\overline{\boldsymbol{x}};\boldsymbol{\Lambda}) = \begin{cases} 0 & , \quad f(p(\overline{\boldsymbol{x}}|C_i)) < \overline{H} \\ f(p(\overline{\boldsymbol{x}}|C_i)) - \overline{H}, & f(p(\overline{\boldsymbol{x}}|C_i)) \ge \overline{H} \end{cases}, \tag{4}$$

respectively. 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

$$L(\boldsymbol{\Lambda}) = \frac{1}{m}\sum_{i=1}^{m} \hat{l}^2(\hat{\boldsymbol{x}}_i;\boldsymbol{\Lambda}) + \frac{1}{n}\sum_{i=1}^{n} \bar{l}^2(\overline{\boldsymbol{x}}_i;\boldsymbol{\Lambda}). \tag{5}$$

The objective of Soft-MMP is to minimize the empirical loss and maximize the difference between $\hat{H}$ and $\overline{H}$, which can be formally given by

$$F(\boldsymbol{\Lambda}) = \omega(1-d)^2 + (1-\omega)L(\boldsymbol{\Lambda}), \tag{6}$$

where $d = \hat{H} - \overline{H}$, and ω is a weight to control the tradeoff between the empirical loss and the difference between two soft targets.

Consequently, the task of Soft-MMP is to find out the optimum parameter set $\boldsymbol{\Lambda}^*$ by minimizing $F(\boldsymbol{\Lambda})$:

$$\boldsymbol{\Lambda}^* = \arg\min_{\Lambda} F(\boldsymbol{\Lambda}). \tag{7}$$

In this work, we insert a data selection component into original Soft-MMP learning process. The data is dynamically selected with soft targets to reduce the overfitting and accelerate the training speed. Actually, the data for which the posterior pseudo-probability has distinctly exceeded the corresponding soft target will be temporarily removed from the training set in certain times of training iterations.

3. Optimization Algorithms

3.1 Gradient Descent Algorithm

In gradient descent algorithm, the following iterative equation is used to update the parameters in Soft-MMP:

$$\boldsymbol{\Lambda}_{t+1} = \boldsymbol{\Lambda}_t - \alpha_t \nabla F(\boldsymbol{\Lambda}_t), \tag{8}$$

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

3.2 CMA Evolution Strategy

In this work, the widely used Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is introduced to optimize parameters in Soft-MMP. Different from traditional ES, CMA-ES update the parameters in ES by exploiting information conveyed by sequences of successful mutations. It has been shown that CMA-ES not only improve the convergence speed, but also reduce the required population size [14, 15, 17, 30].

In the following, we describe the CMA-ES proposed in [14, 15, 17, 30]. In essence, CMA-ES is a kind of real valued (μ, λ)-ES in which λ offspring are generated and μ best of λ offspring are selected to form the next parent population. Each individual represents a real valued object variable vector. The main feature of the CMA-ES is to generate the offspring by two characteristic variation operators, weighted recombination and additive Gaussian mutation. Let $\boldsymbol{y}_k^{(g)} \in \Re^s$ be the k-th offspring in generation g, then we have

$$\boldsymbol{y}_k^{(g+1)} = \langle \boldsymbol{y} \rangle_w^{(g)} + \sigma^{(g)} \boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} z_k^{(g)}, \tag{9}$$

where $\sigma^{(g)}$ is a global step size, $\langle \boldsymbol{y} \rangle_w^{(g)}$ is the weighted mean of the μ best offspring, i.e.

$$\langle \boldsymbol{y} \rangle_w^{(g)} = \sum_{i=1}^{\mu} u_i \boldsymbol{y}_{i:\lambda}^{(g)}; \tag{10}$$

and

$$\boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} z_k^{(g)} \sim \mathrm{N}\left(0, \boldsymbol{C}^{(g)}\right). \tag{11}$$

In Eq. 11, the $z_k^{(g)} \sim \mathrm{N}(0, \boldsymbol{I})$ are independent realizations of a normally distributed random vector with zero mean and covariance matrix equal to the identity matrix $\boldsymbol{I}$, and the covariance matrix $\boldsymbol{C}^{(g)}$ is a symmetric positive matrix with

$$\boldsymbol{C}^{(g)} = \boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)} \left(\boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)}\right)^T. \tag{12}$$

The columns of the orthogonal matrix $\boldsymbol{B}^{(g)}$ are the normalized eigenvectors of $\boldsymbol{C}^{(g)}$, and $\boldsymbol{D}^{(g)}$ is a diagonal matrix with the square roots of the corresponding eigenvalues.

After the offspring is generated by using Eq. 9 and the next parent population is formed by selection, the covariance matrix $\boldsymbol{C}^{(g)}$ and the global step size $\sigma^{(g)}$ are updated. For the covariance matrix, we have:

$$\boldsymbol{p}_c^{(g+1)} = (1 - c_c)\boldsymbol{p}_c^{(g)} + H_\sigma^{(g+1)} \sqrt{c_c(2 - c_c)} \frac{\sqrt{\mu_{eff}}}{\sigma^{(g)}} \left(\langle \boldsymbol{y} \rangle_w^{(g+1)} - \langle \boldsymbol{y} \rangle_w^{(g)}\right), \tag{13}$$

$$\begin{aligned} \boldsymbol{C}^{(g+1)} = {} & (1 - c_{\mathrm{cov}})\boldsymbol{C}^{(g)} + c_{\mathrm{cov}} \frac{1}{\mu_{\mathrm{cov}}} \boldsymbol{p}_c^{(g+1)} \left(\boldsymbol{p}_c^{(g+1)}\right)^T \\ & + c_{\mathrm{cov}} \left(1 - \frac{1}{\mu_{\mathrm{cov}}}\right) \sum_{i=1}^{\mu} \frac{u_i}{\sigma^{(g)2}} \left(\boldsymbol{y}_{i:\lambda}^{(g+1)} - \langle \boldsymbol{y} \rangle_w^{(g)}\right)\left(\boldsymbol{y}_{i:\lambda}^{(g+1)} - \langle \boldsymbol{y} \rangle_w^{(g)}\right)^T \end{aligned} \quad , (14$$

)

where $\boldsymbol{p}_c^{(g+1)}$ is the evolution path, c_c is the learning rate for the rank-one update, c_{cov} controls the update of $\boldsymbol{C}^{(g)}$, $H_\sigma^{(g+1)} = 1$ if $\left\|\boldsymbol{p}_\sigma^{(g+1)}\right\| < (1.5 + 1/n - 0.5)\sqrt{1 - (1 - c_\sigma)^{2(g+1)}} E(\|\mathrm{N}(0, I)\|)$, and 0 otherwise, μ_{cov} balance the rank-one and rand-μ updates.

For the global step size, we have:

$$\boldsymbol{p}_\sigma^{(g+1)} = (1 - c_\sigma)\boldsymbol{p}_\sigma^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)} \boldsymbol{B}^{(g)} \boldsymbol{D}^{(g)^{-1}} \boldsymbol{B}^{(g)^T} \frac{\sqrt{\mu_{eff}}}{\sigma^{(g)}} \left(\langle \boldsymbol{y} \rangle_w^{(g+1)} - \langle \boldsymbol{y} \rangle_w^{(g)}\right), \tag{15}$$

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\left\|\boldsymbol{p}_\sigma^{(g+1)}\right\|}{E(\|\mathrm{N}(0, I)\|)} - 1\right)\right), \tag{16}$$

where d_σ decouples the adaptation rate from the strength of the variation, $E(\|\mathrm{N}(0, I)\|) \approx \sqrt{s}\left(1 - 1/4s + 1/21s^2\right)$ is the expected length of a random variable distributed according to $\mathrm{N}(0, \boldsymbol{I})$, $\mu_{eff} = 1 \Big/ \sum_{i=1}^{\mu} u_i^2$ is the variance effective selection mass, and c_σ controls the update of $\boldsymbol{p}_\sigma$.

3.3 Hybrid of CMA-ES and Gradient Descent

The main advantage of evolution algorithms is the ability of escaping from the local optimum by multi-point stochastic search. However, the efficiency of evolution algorithms is not always satisfactory. It is well known that the direction of generating the offspring is directly related to the convergence speed of evolution algorithms. The gradient has an important property that at any point in search space, it always points to the direction of the maximal increase of the objective function. So it is possible to speed up the search process of evolution algorithms by integrating the gradient descent into the evolution algorithms. In this way, the advantages of evolution algorithms and gradient descent can be merged. On one hand, the local optimum can be escaped by multi-point stochastic search. On the other hand, the training speed can be accelerated by exploiting the gradient information of objective function. Based on this idea, we present a hybrid approach of CMA-ES and gradient descent for Soft-MMP learning in this section.

According to Eqs. 9-16, the offspring generation of CMA-ES is determined by three parameters: the weighted mean $\langle \boldsymbol{y}\rangle_w^{(g)}$, the covariance matrix $\boldsymbol{C}^{(g)}$, and the global step size $\sigma^{(g)}$. But both the covariance matrix and the global step size are adjusted based on the previous evolution path of the weighted mean. It means that the weighted mean is crucial for the efficiency of CMA-ES. Thus we adjust the weighted mean $\langle \boldsymbol{y}\rangle_w^{(g)}$ along the gradient direction of objective function to obtain

$$\langle \boldsymbol{q}\rangle_w^{(g)} = \langle \boldsymbol{y}\rangle_w^{(g)} - \alpha\nabla F\left(\langle \boldsymbol{y}\rangle_w^{(g)}\right), \qquad (17)$$

where α is a constant step size for balancing the influence of gradient and stochastic information. The appropriate α is selected by experiments in this work. $\langle \boldsymbol{q}\rangle_w^{(g)}$ is then used to generate the offspring and update the covariance matrix $\boldsymbol{C}^{(g)}$ and the global step size $\sigma^{(g)}$. That is to say, Eqs.9, 13-15 will be rewritten as

$$\boldsymbol{y}_k^{(g+1)} = \langle \boldsymbol{q}\rangle_w^{(g)} + \sigma^{(g)}\boldsymbol{B}^{(g)}\boldsymbol{D}^{(g)}z_k^{(g)}, \qquad (18)$$

$$\boldsymbol{p}_c^{(g+1)} = (1-c_c)\boldsymbol{p}_c^{(g)} + H_\sigma^{(g+1)}\sqrt{c_c(2-c_c)}\frac{\sqrt{\mu_{eff}}}{\sigma^{(g)}}\left(\langle \boldsymbol{q}\rangle_w^{(g+1)} - \langle \boldsymbol{q}\rangle_w^{(g)}\right), \qquad (19)$$

$$\begin{aligned}\boldsymbol{C}^{(g+1)} &= (1-c_{\mathrm{cov}})\boldsymbol{C}^{(g)} + c_{\mathrm{cov}}\frac{1}{\mu_{\mathrm{cov}}}\boldsymbol{p}_c^{(g+1)}\left(\boldsymbol{p}_c^{(g+1)}\right)^T \\ &+ c_{\mathrm{cov}}\left(1-\frac{1}{\mu_{\mathrm{cov}}}\right)\sum_{i=1}^{\mu}\frac{u_i}{\sigma^{(g)2}}\left(\boldsymbol{y}_{i:\lambda}^{(g+1)} - \langle \boldsymbol{q}\rangle_w^{(g)}\right)\left(\boldsymbol{y}_{i:\lambda}^{(g+1)} - \langle \boldsymbol{q}\rangle_w^{(g)}\right)^T\end{aligned}, \qquad (20)$$

$$\boldsymbol{p}_\sigma^{(g+1)} = (1-c_\sigma)\boldsymbol{p}_\sigma^{(g)} + \sqrt{c_\sigma(2-c_\sigma)}\boldsymbol{B}^{(g)}\boldsymbol{D}^{(g)-1}\boldsymbol{B}^{(g)T}\frac{\sqrt{\mu_{eff}}}{\sigma^{(g)}}\left(\langle \boldsymbol{q}\rangle_w^{(g+1)} - \langle \boldsymbol{q}\rangle_w^{(g)}\right), \qquad (21)$$

respectively.

According to Eqs. 17-21, the Soft-MMP learning algorithm with the proposed hybrid optimization approach is summarized in Table 1. The whole procedure of Soft-MMP learning is to perform this algorithm for all the classes under consideration.

Table 1. Soft-MMP Algorithm with Our Hybrid Optimization Approach

```
Initialization
Given an initial parameter set $\boldsymbol{\Lambda}$ of Soft-MMP, arrange parameters in $\boldsymbol{\Lambda}$ in fixed order to form the initial genotype.
Repeat
    1. Generate $\lambda$ offspring based on Eq. 18.
    2. Compute fitness of each offspring using Eq. 6.
    3. The $\mu$ individuals with best fitness were selected to form the new parent population.
    4. The weighted mean $\langle \boldsymbol{y}\rangle_w^{(g)}$ is computed from the new parent population by using Eq. 10 and adjusted according to Eq. 17.
    5. Update the covariance matrix $\boldsymbol{C}^{(g)}$ and global step size $\sigma^{(g)}$ using Eq. 20 and Eq. 16, respectively.
Until convergence or the maximum generation times is reached.
Let $\varepsilon$ be an infinitesimal, then the convergence condition is
□ $\max\{F(\boldsymbol{y}_i)\} - \min\{F(\boldsymbol{y}_i)\} \le \varepsilon, i = 1\cdots\lambda$.
```

4. Experimental Evaluation

In order to evaluate the Soft-MMP with the proposed hybrid optimization approach, we apply it to handwritten digit recognition and conducted experiments on the well-known CENPARMI database of handwritten digits [29].

4.1.1 Digit Modeling and Learning

The 8-direction gradient features of Liu et al. [22] (abbreviate e-grg there) are used to represent digits in the experiments. In this paper, the original 200-D e-grg is compressed to 100-D by the Principal Component Analysis (PCA) method to reduce the computation cost.

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, $\boldsymbol{\varphi_k}$, and $\boldsymbol{\Sigma_k}$ be the weight, the mean, and the covariance matrix of the k-th Gaussian component, respectively. $\sum_{k=1}^{K} w_k = 1$. Then we have

$$p(\boldsymbol{x}|C_i) = \sum_{k=1}^{K} w_k N(\boldsymbol{x}|\boldsymbol{\varphi}_k, \boldsymbol{\Sigma}_k) \qquad (22)$$

where

$$N\left(\boldsymbol{x}\middle|\boldsymbol{\varphi}_k, \boldsymbol{\Sigma}_k\right) = (2\pi)^{-50}\left|\boldsymbol{\Sigma}_k\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\varphi}_k\right)^T \boldsymbol{\Sigma}_k^{-1}\left(\boldsymbol{x}-\boldsymbol{\varphi}_k\right)\right) \quad (23)$$

$\boldsymbol{\Sigma_k}$ is further assumed to be diagonal for simplicity, i.e., $\boldsymbol{\Sigma_k} = \left[\gamma_{kj}\right]_{j=1}^{100}$. In order to make this assumption reasonable, we use the orthogonal GMM to decrease the correlation among the elements of the feature vector [33].

The unknown parameters in Soft-MMP learning of the classifier described above are

$$\boldsymbol{\Lambda} = \{\kappa, \beta, w_k, \boldsymbol{\varphi}_k, \boldsymbol{\Sigma}_k, \hat{H}, \overline{H}\}, k = 1, \cdots, K \,. \quad (24)$$

Some parameters in Eq. 24 must satisfy certain constraints, which are transformed to unconstrained domain for easier implementation. The constraints and transformation of parameters are listed in Table 2. In Table 2, τ is the preset minimum variance value in the covariance matrices of GMM for avoiding the estimation error caused by too small variance values. To sum up, the transformed parameter set is

$$\widetilde{\boldsymbol{\Lambda}} = \{\widetilde{\kappa}, \widetilde{\beta}, \widetilde{w}_k, \boldsymbol{\varphi}_k, \widetilde{\boldsymbol{\Sigma}}_k, h_1, h_2\}, k = 1, \cdots, K \,. \quad (25)$$

We use the Soft-MMP learning algorithm with the proposed hybrid optimization approach to estimate these parameters, and then transform them into the original ones.

Table 2. The constrains and transformation of parameters in Soft-MMP learning of digit classifier.

□ Original parameters and constrains	□ Transformation of parameters
$0 < \hat{H} < 1$; $0 < \overline{H} < 1$	$\hat{H} = \dfrac{1}{1+e^{-h_1}}$; $\overline{H} = \dfrac{1}{1+e^{-h_2}}$
$\kappa > 0$; $\beta > 0$	$\kappa = \exp\left(\widetilde{\kappa}\right)$; $\beta = \exp\left(\widetilde{\beta}\right)$
□ $\gamma_{kj} > \tau$	□ $\gamma_{kj} = \exp\left(\widetilde{\gamma}_{kj}\right) + \tau$
□ $\sum w_k = 1$	□ $w_k = \dfrac{e^{\widetilde{w}_k}}{\sum e^{\widetilde{w}_k}}$

4.1.2 Experimental results

We conducted the experiments of handwritten digit recognition on the CENPARMI database [29], which includes 4,000 training samples and 2,000 test samples.

The training process of our digit classifier includes three stages. In the first stage, the number of components in the GMM for each digit class was estimated by AutoClass, a clustering algorithm with automatic determination of cluster number [2].

In the second stage, we used the Expectation-Maximization (EM) algorithm on positive samples of each digit class to get the Maximum Likelihood Estimation (MLE) of parameters in the corresponding GMM model. We further set $\kappa = 10$, $\beta = 0.01$ in Eq. 1 and $\omega = 0.01$ in Eq. 6 by careful experiments. It should be noted that the EM algorithm was implemented using Torch machine learning library [34].

In the third stage, Soft-MMP with each of three optimization algorithms, i.e. gradient descent, CMA-ES, and the proposed hybrid algorithm, were used to find out unknown parameters in Eq. 24. As a result, three classifiers based on posterior pseudo-probabilities were obtained. We implemented closed and open tests of handwritten digit recognition by using each of these three classifiers. Table 3 displays the error rates on the training set (Train) and the test set (Test), as well as the reduction in the error rate for the test set (Reduction_test), which is brought by the hybrid approach compared with the single CMA-ES and the single gradient descent, respectively. It should be noted that the key parameters in CMA-ES were set to default values advised by Hansen [14], which are listed in Table 4. Furthermore, we reduce the computation cost of CMA-ES by updating $\boldsymbol{B}^{(g)}$ and $\boldsymbol{D}^{(g)}$ in each 50th generation instead of in each generation, as that also advised in [14]. The maximum number of iterations was set to 50000 for all the three optimization methods. For CMA-ES and our Hybrid approach, we tried ten times of optimization and selected the optimization result achieving the best recognition accuracy, respectively.

As shown in Table 3, our hybrid optimization algorithm obtained the better result than the single gradient descent or the single CMA-ES. Furthermore, the CMA-ES shows its superiority over gradient descent. In the work of Liu et al. [21], state-of-the-art techniques of handwritten digit recognition, including features and classifiers, are thoroughly investigated on the CENPARMI database. They reported the best error rate of 0.95% on the test set for e-grg features by using either SVM with RBF kernel or Discriminative Learning Quadratic Discriminant Function. Soft-MMP with the CMA-ES or proposed hybrid optimization approach both achieved better results on same features by courtesy of Liu. Furthermore, we also collected other up-to-date results on the CENPARMI database and compare them with ours in Table 5. It shows that the performance of the Soft-MMP with our hybrid optimization approach outperforms other state-of-the-art techniques. The experimental results reported above demonstrate that the combination of CMA-ES and gradient descent is promising to solve optimization problems in discriminative learning.

Table 3. Error rates from three optimization methods

Optimization Methods	Train (%)	Test (%)	Reduction_test(%)
Gradient Descent	0.200	1.05	28.57
CMA-ES	0.125	0.85	11.76
Hybrid	**0.075**	**0.75**	-

Table 4. CMA-ES Parameter Setting

For Recombination and Mutation:

$$\lambda = 4 + \lfloor 3\ln(s) \rfloor \quad , \quad \mu = \lfloor \lambda/2 \rfloor \quad , \quad u_i = u_i' \Big/ \sum_{i=1}^{\mu} u_i' \quad ,$$

$$u_i' = \ln(\mu+1) - \ln i \, , \; \boldsymbol{C}^{(0)} = \boldsymbol{I}$$

For Covariance Matrix Adjustment:

$$\boldsymbol{p}_c^{(0)} = 0 \, , \; c_c = \frac{4}{s+4} \, , \; \mu_{\mathrm{cov}} = \mu_{eff} \, ,$$

$$c_{\mathrm{cov}} = \frac{1}{\mu_{\mathrm{cov}}} \frac{2}{\left(s+\sqrt{2}\right)^2} + \left(1 - \frac{1}{\mu_{\mathrm{cov}}}\right) \min\left(1, \frac{2\mu_{eff} - 1}{(s+2)^2 + \mu_{eff}}\right)$$

For Global Step Size Adjustment:

$$d_\sigma = 1 + 2\max\left(0, \sqrt{\frac{\mu_{eff} - 1}{s+1}} - 1\right) + c_\sigma \quad , \quad c_\sigma = \frac{\mu_{eff} + 2}{s + \mu_{eff} + 3} \quad ,$$

$$\boldsymbol{p}_\sigma^{(0)} = 0 \, , \; \sigma^{(0)} = 0.5$$

Table 5. Error rates of different classification methods on CENPARMI database

Method	Feature	Test (%)
Modular Neural Network [26]	class dependent features	2.15
Local Learning Framework[9]	32 direction gradient features	1.90
Neural Network[12]	random features	1.70
Virtual SVM [10]	32 direction gradient features	1.30
SVC-rbf [22]	8 direction deslant chaincode features	0.85
CMA-ES	e-grg	0.85
Our Method	**e-grg**	**0.75**

In order to analyze the convergence speed improvement brought by the proposed hybrid approach, we recorded the optimization results of single CMA-ES or our hybrid approach during the first 250 generations for each class, where the optimization result is the best objective function value among the parent population. The results are plotted in Fig. 1, which show that the convergence of our hybrid approach is faster than the CMA-ES. By using our hybrid approach, the objective value goes to 0.01 after average 60 generations. However, the average 200 generations are needed to obtain the same objective value in the CMA-ES.

We also recorded the average training time consumed by each of three optimization methods, which is 2133 seconds, 2340 seconds, and 1856 seconds for gradient decent, CMA-ES, and hybrid one, respectively. It means that the proposed approach brought 12.99% and 20.68% reduction in training time, compared with gradient decent and CMA-ES, respectively.

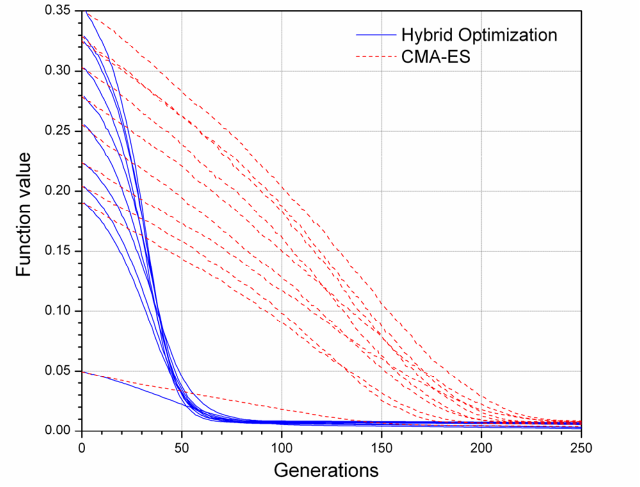

Figure 1. The runs on the objective function of Soft-MMP with CMS-ES or Our Hybrid Approach

5. CONCLUSIONS

Until now, gradient-based methods are main optimization means in the field of discriminative learning. Since the gradient-based methods tend to get stuck in local optimum, the result of discriminative learning is not always satisfactory. In this paper, we introduce the evolution strategy into a new discriminative learning framework of Bayesian classifiers, SOFT target based Max-Min posterior Pseudo-probabilities (Soft-MMP). Our main contributions are summarized as follows.

(1) A hybrid optimization approach of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) and the gradient decent have been proposed, which combines the advantages of CMA-ES and gradient descent. On one hand, the risk of getting stuck at local optimum is decreased by multi-point random global search. On the other hand, the convergence speed is accelerated by exploiting the gradient information of objective function in parameter evolution.

(2) The hybrid optimization approach has been applied to the Soft-MMP and tested by handwritten digit recognition. A corresponding handwritten digit classifier has been established, which shows its superiority over other state-of-the-art techniques in the experiments.

It should be noted that the power of evolution strategy has not been fully exploited in this work, since all the algorithms are implemented serially. In the future, we will investigate the efficiency improvement of evolution strategy based discriminative learning through parallel implementation. Another future research direction is the applications of the proposed hybrid optimization approach to other optimization problems, especially to other discriminative learning methods. Furthermore, as some reviewers suggested, we will perform statistical test to show the significance of advantages of our approach and discuss the impact and sensitivity of step size in Eq. 17 in the future.

6. ACKNOWLEDGEMENTS

The authors would like to thank Dr. Cheng-Lin Liu for providing the e-grg features of handwritten digits and the referees for their valuable comments and suggestions. This research was partially supported by 973 Program of China (No. 2006CB303103), Excellent Young Scholars Research Fund of Beijing Institute of Technology (No. 2008YS1203), Open Fund of the National Laboratory of Pattern Recognition, and MCE Theme Project Fund of Microsoft Research Asia (No. FY08-RES-THEME-158).

7. REFERENCES

[1] Auger, A., Schoenauer, M., and Vanhhaecke, N. 2004. LS-CMA-ES: A Second-Order Algorithm for Covariance Matrix Adaptation. In Proceedings of Eighth International Conference on Parallel Problem Solving from Nature PPSN VIII (2004). Springer, Berlin, 1611-3349.

[2] Cheeseman, P., Stutz, J. 1996. Bayesian Classification (AutoClass): theory and results, Advances in knowledge discovery and data mining, AAAI Press, Menlo Park, CA. USA, 153-180.

[3] Chellapilla, K., Fogel, D.B. 2001. Evolving an Expert Checkers Playing Program without Using Human Expertise. IEEE Trans. Evolutionary Computation, 5, 4 (2001), 422-428.

[4] Liu, X.B., Jia, Y.D., Chen, X.F., Deng, Y. and Fu, H. 2008 Image Classification Using the Max-Min Posterior Pseudo-Probabilities Method. Technical Report, BIT-CS-20080001. Beijing Institute of Technology. DOI=http://www.mcislab.org.cn/member/~xiabi/papers/2008_1.PDF

[5] Chen, X.F., Liu, X.B., and Jia, Y.D. 2008. A Soft Target Method of Learning Posterior Pseudo-probabilities based Classifiers with its Application to Handwritten Digit Recognition, 2008 11th International Conference on Frontiers in Handwriting Recognition (Montréal, Canada, August 2008). ICFHR'08.

[6] Chen, X.F., Liu, X.B., and Jia, Y.D. 2007. Learning Handwritten Digit Recognition by the Max-Min Posterior Pseudo-Probabilities Method. The 9th International Conference on Document Analysis and Recognition (Curitiba, Brazil, Sept. 2007). ICDAR'07. IEEE, 320-324.

[7] Cochenour, G., Simon, J., Das, S., Pahwa, A., Nag, S. 2005. A pareto archive evolutionary strategy based radial basis function neural network training algorithm for failure rate prediction in overhead feeders. In Proceedings of the Genetic and Evolutionary Computation Conference (Washington DC, USA, Jun., 2005). GECCO'05. ACM Press, New York, NY, 2005, 2127-2132.

[8] Deng, Y., Liu, X.B., Jia, Y.D. 2007. Learning Semantic Concepts for Image Retrieval Using the Max-min Posterior Pseudo-Probabilities Method. In Proceedings of 2007 IEEE International Conference on Multimedia and Expo (Beijing China, 2007). ICME'07. 1970-1973.

[9] Dong, J.X., Krzyzak, A., Suen, C.Y. 2002. Local Learning framework for handwritten character recognition. Engineering Applications of Artificial Intelligence, 15 (2002), 151-159.

[10] Dong, J.X., Krzyzk, A., Suen, C.Y. 2005. Fast SVM Training Algorithm with Decomposition on Very Large Datasets. IEEE Trans. Pattern Analysis Machine Intelligence, 27, 4 (April 2005), 603-618. DOI=http://www.cenparmi.concordia.ca/~jdong/HeroSvm.html

[11] Frőhlich, H., Chapelle, O., Schőlkopf, B. 2003. Feature selection for support vector machines by means of genetic algorithms. In Proceedings of the 15th IEEE International Conference on Tools with AI (Sacramento, CA, United states, Nov. 03-05, 2003). ICTAI'03. IEEE, 142-148.

[12] Gader, P.D., Khabou, M.A. 1996. Automatic feature generation for handwritten digit recognition. IEEE Trans. Pattern Analysis Machine Intelligence, 18, 12 (1996), 1256-1261.

[13] Goh, C.K., Teoh, E.J., Tan, K.C. 2008. Hybrid Multiobjective Evolutionary Design for Aritificial Neural Networks. IEEE Trans. Neural Networks, 19, 9 (Sep. 2008), 1531-1548.

[14] Hansen, N. 2006. An analysis of mutative σ-self-Adaptation on linear fitness functions. Evolutionary Computation, 14, 3 (2006), 255-275.

[15] Hansen, N., Niederberger, A. S. P., Guzzella, L., and Koumoutsakos, P. 2009. A Method for Handling Uncertainty in Evolutionary Optimization with an Application to Feedback Control of Combustion. IEEE Trans. Evolutionary Computation, 13, 1 (2009), 180-197.

[16] Hastie, T., Tibshirani, R., Friedman, J. 2001. The Elements of Statistical Learning, Data Mining, Inference, and Prediction. Springer Series in Statistica. Springer.

[17] Igel, C., Suttorp, T., and Hansen, N. 2006. A Computational Efficient Covariance Matrix Update and a (1+1)-CMA for Evolution Strategies. In Proceedings of the Genetic and Evolutionary Computation Conference (Seattle, Washington, USA, Jul., 2006). GECCO'06. ACM Press, New York, NY, 2006, 453-460.

[18] Igel, C. 2005. Multiobjective Model Selection for Support Vector Machines. In Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (Guanajuato, Mexico, March 9-11, 2005). EMO'05. Springer, Berlin, 534-546.

[19] Jung, J.Y., Reggia. 2006. J.A. Evolutionary Design of Neural Network Architectures Using a Descriptive Encoding Language. IEEE Trans. Evolutionary Computation, 10, 6 (Dec. 2006), 676-688.

[20] Leung, F., Lam, H., Ling S., and Tam, P. 2003. Tuning of the structure and parameters of a neural network using an improved genetic algorithm. IEEE Trans. Neural Network, 14, 1 (Jan. 2003), 79-88.

[21] Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H. 2003. Handwritten digit recognition: benchmarking of state-of-the-art techniques. Pattern Recognition, 36 (2003), 2271-2285.

[22] Liu, C.L., Nakashima, K., Sako, H., Fujisawa, H. 2004. Handwritten digit recognition: investigation of normalization and feature extraction techniques. Pattern Recognition, 37 (2004), 265-279.

[23] Liu, X.B., Fu, H., Jia, Y.D. 2008. Gaussian mixture modeling and learning of neighboring characters for multilingual text extraction in images, Pattern Recognition, 41 (2008), 484-493.

[24] Mierswa, I. 2007. Controlling Overfitting with Multi-Objective Support Vector Machines. In Proceedings of the Genetic and Evolutionary Computation Conference (London, England, Jul., 2007). GECCO'07. ACM Press, New York, NY, 1830-1837.

[25] Nikolaerv, N., lba, H. 2003. Learning polynomial feedforward neural networks by genetic programming and backpropagation. IEEE Trans. Neural Network, 14, 2 (Mar. 2003), 337-350.

[26] Oh, I.S., Lee, J.S., Suen, C.Y. 1999. Analysis of class separation and combination of class-dependent features for handwriting recognition. IEEE Trans. Pattern Analysis Machine Intelligence, 12, 10 (1999), 1089-1094.

[27] Palmes, P.P., Hayasaka, T., Usui, S. 2005. Mutation-Based Genetic Neural Network. IEEE Trans. Neural Networks, 16, 3 (May, 2005), 587-600.

[28] Phienthrakul, T., Kijsirikul, B. 2005. Evolutionary Strategies for Multi-Scale Radial Basis Function Kernels in Support Vector Machines. In Proceedings of the Genetic and Evolutionary Computation Conference (Washington DC, USA, Jun., 2005). GECCO'05. ACM Press, New York, NY, 905-911.

[29] Suen, C.Y., et al. 1992. Computer recognition of unconstrained handwritten numerals. Proc. IEEE, 80, 7 (1992), 1162-1180.

[30] Suttorp, T., Hansen, N., Igel, C. 2009. Efficient covariance matrix update for variable metric evolution strategies, Machine Learning, 75 (2009), 167-197.

[31] Wu, C.H, Tzeng, G.H., Goo, Y. J., Fang, W.C. 2007. A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. Expert Systems with Applications, 32 (2007), 397-408.

[32] Yao, X. 1999. Evolving Artificial Neural Networks. Proceedings of the IEEE, 87, 9 (1999), 1423-1447.

[33] Zhang, R., Ding, X.Q. 2001. Offline Handwritten Numeral Recognition Using Orthogonal Gaussian Mixture Model. In Proceedings 6th Int. Conference document Analysis and Recognition (Seattle, USA, 2001). ICDAR'01. IEEE, 1126-1129.

[34] Torch Machine Learning Library. DOI = http://www.torch.ch