published in the Proceedings of ICANN97, Stockholm, Sweden, June 1997, pp. 87-90
available at <URL:http://www.cis.hut.fi/~jorma/papers/>

Error-Corrective Feature Extraction in Handwritten Digit Recognition

Jorma Laaksonen and Erkki Oja

Helsinki University of Technology
Laboratory of Computer and Information Science
Rakentajanaukio 2C, FIN-02150 Espoo, FINLAND

Abstract

The idea of feedback and error-correction is central in neurally motivated classification algorithms. Most of the neural models, however, take the preceding feature extraction stage as given. Unfortunately, essential information may be unrecoverably lost in the feature extraction phase, leading to degraded classification accuracy. Enhanced overall classification performance would follow, if the neural adaptivity could be extended from the classifier to the feature extractor.

In our work, we have studied two neural feature extraction schemes which allow for adaptation of the feature extraction stage. The adaptation process is directed by errors occurring in the classifier. During the adaptation of the feature extractor, k-NN rule has been used as the classifier, due to its zero training time. More advanced classifiers can be used after the adaptation has ended. The usefulness of the proposed methods is demonstrated by a series of experiments carried out with handwritten digit data.

Keywords: feature extraction, subspace methods, statistical classification, handwritten digit recognition

1 Introduction

In this paper, error-corrective feature extraction is applied to recognition of handwritten digits. In many recognition systems and classifier comparisons, e.g., [1], [2], and [5], features of the Karhunen-Loève Transform (KLT) type have been used. Therefore, we have taken the KLT features as the starting point of our study.

In the traditional setting of pattern recognition, the system designer selects the features extracted and used in classification. Such an arrangement can hardly be optimal, because any information loss in the feature extraction stage will be irreversible and impair the classification accuracy. In this paper, we put forward the question, whether the feature extraction phase could be made adaptive and autonomously error-corrective in a manner familiar from neural network classifiers. Two such error-corrective feature extraction schemes are introduced here. In the first version, the extracted features are yet not truly adaptive, but the feature selection is controlled by the resulting overall classification accuracy. In the second formulation, the feature extraction stage is revised using each individual misclassified vector in a manner familiar from the learning subspace methods of classification. The feature extraction phase may thus be regarded genuinely neural because it is adaptive and able to learn from examples.

Given two high-dimensional input vectors $\mathbf{f_x}$ and $\mathbf{f_y} \in \mathbb{R}^D$ and a linear transform $\mathbf{K} \in \mathbb{R}^{D \times d}$ used in producing the lower-dimensional feature vectors $\mathbf{x}$ and $\mathbf{y} \in \mathbb{R}^d$, respectively, the squared Euclidean distance between the latter two may be expressed:

$$d^2(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^2 = (\mathbf{K}^T\mathbf{f_x}-\mathbf{K}^T\mathbf{f_y})^T(\mathbf{K}^T\mathbf{f_x}-\mathbf{K}^T\mathbf{f_y}) \quad (1)$$

$$= (\mathbf{f_x}-\mathbf{f_y})^T\mathbf{K}\mathbf{K}^T(\mathbf{f_x}-\mathbf{f_y}) \quad (2)$$

How faithfully the original metrics of the input space are transformed to the feature space, depends thus on the eigendirections of the $\mathbf{K}\mathbf{K}^T$ matrix. By affecting the $\mathbf{K}$ matrix, the resulting classification accuracy can be enhanced as will be explained in the following.

2 Method One

In the first formulation, the covariance matrix used in KLT is replaced by a sum matrix enhancing the directions between the classes and across the class borders. We define three matrices, the first of which, $\widehat{\mathbf{\Sigma}_\mathbf{f}}$, is formed from the original normalized pattern vectors $\{\mathbf{f}_1, \ldots, \mathbf{f}_n\}$ and their estimated mean $\widehat{\boldsymbol{\mu}_\mathbf{f}}$ just as in KLT:

$$\widehat{\mathbf{\Sigma}_\mathbf{f}} = \frac{\sum_{i=1}^{n} (\mathbf{f}_i - \widehat{\boldsymbol{\mu}_\mathbf{f}})(\mathbf{f}_i - \widehat{\boldsymbol{\mu}_\mathbf{f}})^T}{n} , \tag{3}$$

The second matrix, $\widehat{\mathbf{A}_\mathbf{f}}$, models the average directions from one class to another: every pair of vectors, $\mathbf{f}_{kj}$ and $\mathbf{f}_{li}$, which belong to separate classes, i and j, respectively, contributes to the sum. The number of training vectors in class j is denoted by n_j:

$$\widehat{\mathbf{A}_\mathbf{f}} = \frac{\sum_{j=1}^{c} \sum_{i=1, i\neq j}^{c} \sum_{k=1}^{n_j} \sum_{l=1}^{n_i} (\mathbf{f}_{kj} - \mathbf{f}_{li})(\mathbf{f}_{kj} - \mathbf{f}_{li})^T}{\sum_{j=1}^{c} \sum_{i=1, i\neq j}^{c} n_j n_i} , \tag{4}$$

The third matrix, $\widehat{\mathbf{B}_\mathbf{f}}$, goes even further: only the vector pairs which really are located in the areas of class borders are used:

$$\widehat{\mathbf{B}_\mathbf{f}} = \frac{\sum_{j=1}^{c} \sum_{i=1, i\neq j}^{c} \sum_{k=1}^{n_j} \sum_{l=1}^{n_i} s_p(\mathbf{f}_{kj}, \mathbf{f}_{li})\ (\mathbf{f}_{kj} - \mathbf{f}_{li})(\mathbf{f}_{kj} - \mathbf{f}_{li})^T}{\sum_{j=1}^{c} \sum_{i=1, i\neq j}^{c} \sum_{k=1}^{n_j} \sum_{l=1}^{n_i} s_p(\mathbf{f}_{kj}, \mathbf{f}_{li})} , \tag{5}$$

$$s_p(\mathbf{f}_{kj}, \mathbf{f}_{li}) = \begin{cases} 1 , & \text{iff } li = \underset{tu,\ tu\neq kj}{\operatorname{argmin}} |\mathbf{f}_{kj} - \mathbf{f}_{tu}| , \\ 0 , & \text{otherwise.} \end{cases} \tag{6}$$

The pattern vector misclassification function $s_p(\mathbf{f}_{kj}, \mathbf{f}_{li})$ attains a non-zero value actually only in the situations when the vector $\mathbf{f}_{kj}$ would be erroneously classified using the leave-one-out 1-NN classification in the normalized input pattern space. In the limiting case of an infinite training sample, $\widehat{\mathbf{B}_\mathbf{f}}$ is formed from vectors perpendicular to the Bayesian class border in $\mathbb{R}^D$.

A compound covariance-type matrix $\mathbf{S}$ is expressed as a weighted linear combination of the above three matrices and two coefficients α and β:

$$\mathbf{S} = \alpha\widehat{\mathbf{A}_\mathbf{f}} + \beta\widehat{\mathbf{B}_\mathbf{f}} + (1 - \alpha - \beta)\widehat{\mathbf{\Sigma}_\mathbf{f}} . \tag{7}$$

In this feature extraction scheme, the kernel matrix $\mathbf{K} = \mathbf{K}(\alpha, \beta)$ is formed from the eigenvectors of the $\mathbf{S}$ matrix in the fashion of KLT. As discussed above, the matrix $\mathbf{K}\mathbf{K}^T$ determines how faithfully the original metrics of the input space $\mathbb{R}^D$ are transformed to the feature space $\mathbb{R}^d$. By incorporating some amount of the $\widehat{\mathbf{A}_\mathbf{f}}$ and $\widehat{\mathbf{B}_\mathbf{f}}$ matrices in $\mathbf{S}$, the features can be made somewhat more sensitive to the differences between the classes than the overall shape of the input vector distribution. Optimal values for the multipliers α and β need to be found experimentally by using error cross-validation or some other method by observing the resulting classification performance.

3 Method Two

In the second formulation of error-corrective feature extraction, each individual misclassified vector is directly used to revise the feature extraction process. The misclassifications are observed in the feature vector space $\mathbf{x} = \mathbf{K}^T\mathbf{f}$, but the corrections to the feature extractor are made using the original normalized input images $\mathbf{f}$. The matrix $\mathbf{S}$ is now made adaptive and its initial value can be obtained using any of the earlier defined matrices $\widehat{\mathbf{\Sigma}_\mathbf{f}}$, $\widehat{\mathbf{A}_\mathbf{f}}$, and $\widehat{\mathbf{B}_\mathbf{f}}$, or the optimized linear combination thereof resulting from the above optimization process:

$$\mathbf{S}(0) = n\,(\alpha\widehat{\mathbf{A}_\mathbf{f}} + \beta\widehat{\mathbf{B}_\mathbf{f}} + (1-\alpha-\beta)\widehat{\mathbf{\Sigma}_\mathbf{f}})\ , \tag{8}$$

The iteration formula is:

$$\mathbf{S}(t+1) = \mathbf{S}(t) + \gamma(t)\sum_{j=1}^{c}\sum_{i=1,i\neq j}^{c}\sum_{k=1}^{n_j}\sum_{l=1}^{n_i} s_f(\mathbf{f}_{kj},\mathbf{f}_{li})\,(\mathbf{f}_{kj}-\mathbf{f}_{li})(\mathbf{f}_{kj}-\mathbf{f}_{li})^T\ , \tag{9}$$

$$s_f(\mathbf{f}_{kj},\mathbf{f}_{li}) = \begin{cases} 1\ , & \text{iff } g(\mathbf{K}(t)^T\mathbf{f}_{kj}) \neq j, \text{ i.e., } \mathbf{K}(t)^T\mathbf{f}_{kj} \text{ is misclassified} \\ & \text{and } il = \underset{tu,\ tu\neq kj}{\operatorname{argmin}} |\mathbf{K}(t)^T\mathbf{f}_{kj} - \mathbf{K}(t)^T\mathbf{f}_{tu}|\ , \\ 0\ , & \text{otherwise.} \end{cases} \tag{10}$$

The kernel matrix $\mathbf{K}(t+1)$ is again formed from the principal eigenvectors of the $\mathbf{S}(t+1)$ matrix after the training period t. During each training epoch, all the n feature vectors formed from the normalized input images $\mathbf{f}$ using the transform $\mathbf{K}(t)$ are tentatively classified. The difference between the pattern vector misclassification function $s_p(\cdot,\cdot)$ in (6) and the feature vector misclassification function $s_f(\cdot,\cdot)$ in (10) is that $s_p(\cdot,\cdot)$ is a function indicating incorrectness of 1-NN classification in the high-dimensional $\mathbb{R}^D$ pattern space, whereas $s_f(\cdot,\cdot)$ reports the misclassifications in the lower-dimensional $\mathbb{R}^d$ feature vector space. The latter error-corrective feature extraction version is thus not simply an iterative version of the former but, instead, models more accurately the actual operation of the combination of a feature extractor and a classifier. The coefficient $\gamma(t)$ in (9) needs to be given an appropriate constant or monotonically decreasing positive value which produces optimal final classification performance. The classification function $g(\mathbf{x})$ used in implementing (10) can be selected appropriately, k-NN classifier being an apparent choice.

4 Experiments

In the experiments, handwritten digit data were used. A set of 17880 binary digit images were normalized to the size of 32×32. This set was then divided to equally-sized training and testing sets each containing 894 samples of each of the ten digit classes. The same sets of data have earlier been used in experiments reported in [3]. In the current experiments, 3-NN classifier has been used and the optimal feature vector dimensionality d has been solved for each situation independently by observing the error rate attained with the independent testing sample.

First, the initial classification accuracy corresponding to $\alpha = \beta = 0$ in (7) was found to be 3.77% errors with $d = 28$. Then, some simple values were given to α and β, feature extraction was performed accordingly and classification error rate was evaluated. In these experiments, the best classification performance was attained with $\alpha = 0$ and $\beta = 1$. In the vicinity of that point, the values of α and β were varied and the classification error rates were calculated. The best accuracy, 3.45% errors, was attained with the combination $\alpha = 0.1667, \beta = 0.9167, d = 33$.

In the second set of experiments, the matrix $\mathbf{S}(t)$ was iteratively modified using (9) and (10) starting from the situation $\mathbf{S}(0) = n\widehat{\mathbf{\Sigma}_\mathbf{f}}$ in (8). The correction-rate parameter $\gamma(t)$ in (9) was given constant values in the range $[0.5, 5]$ with an increment of 0.5 and the development of the classification accuracy was observed. After each iteration the optimal feature vector dimensionality $d(t+1)$ was resolved from the range of $[d(t)-5, d(t)+5]$. The best classification accuracy was

attained with $\gamma(t) = 3$ after 33 iterations and was 3.33% errors. Then, the $\gamma(t)$ factor was made to decrease with time by setting $\gamma(t) = \frac{\gamma(0)}{t+1}$, $\gamma(0)$ varying in the range $[5, 50]$ with an increment of 5. The best accuracy obtained was 3.31% errors with $\gamma(0) = 35, d = 22$ after 8 iterations.

The classification performances collected in Table 1 show that both the proposed error-corrective feature extraction methods clearly lowered the overall error rate compared to the original accuracy obtained with the Karhunen-Loève transformed features.

features	ϵ-%	parameters
KLT features	3.77	$d = 28$
method one	3.45	$d = 33, \alpha = 0.1667, \beta = 0.9167$
method two – constant $\gamma(t)$	3.33	$d = 29, \gamma(t) = 3$, 33 iterations
method two – decreasing $\gamma(t)$	3.31	$d = 22, \gamma(0) = 35$, 8 iterations

Table 1: Final classification error percentages obtained using the two forms of error-corrective feature extraction. The classifier used in the experiments was a 3-NN classifier.

5 Discussion

The experiments performed showed that both of the introduced methods clearly increased the classification accuracy of a simple k-NN classifier compared to the original accuracy attained with Karhunen-Loève type features. The classification error rates of this paper are not as such comparable to those presented in [3]. Nevertheless, it can be stated that the improvement in the classification accuracy attained in the above described experiments is about half of the difference in classification accuracy between the k-NN and the best classifier in the comparison.

The relationship between the principles and formalisms of the second error-corrective feature extraction scheme and the ALSM classification method [4] should be noted. In both approaches, the outer products of all erroneously classified vectors are added to a scatter-type matrix which is after each training epoch re-eigenvectorized. It may therefore be stated that the proposed feature extraction scheme is related to KLT as its precedent in a similar manner as ALSM is related to CLAFIC as its own predecessor.

References

[1] J. L. Blue, G. T. Candela, P. J. Grother, R. Chellappa, and C. L. Wilson. Evaluation of pattern classifiers for fingerprint and OCR applications. *Pattern Recognition*, 27(4):485–501, 1994.

[2] J. Geist, R. A. Wilkinson, S. Janet, P. J. Grother, B. Hammond, N. W. Larsen, R. M. Klear, M. J. Matsko, C. J. C. Burges, R. Creecy, J. J. Hull, T. P. Vogl, and C. L. Wilson. The second census optical character recognition systems conference. Technical Report NISTIR 5452, National Institute of Standards and Technology, August 1992.

[3] L. Holmström, P. Koistinen, J. Laaksonen, and E. Oja. Neural and statistical classifiers – taxonomy and two case studies. *IEEE Transactions on Neural Networks*, 8(1):5–17, January 1997.

[4] E. Oja. *Subspace Methods of Pattern Recognition*. Research Studies Press Ltd., Letchworth, England, 1983.

[5] R. A. Wilkinson, J. Geist, S. Janet, P. J. Grother, C. J. C. Burges, R. Creecy, B. Hammond, J. J. Hull, N. W. Larsen, T. P. Vogl, and C. L. Wilson. The first census optical character recognition systems conference. Technical Report NISTIR 4912, National Institute of Standards and Technology, August 1991.