Principal Component Analysis for Online Handwritten Character Recognition

Deepu V.
Hewlett-Packard Labs India
Bangalore 560 030

Sriganesh M.
Hewlett-Packard Labs India
Bangalore 560 030

Ramakrishnan A. G.
Indian Institute of Science
Bangalore 560 012

Abstract

In this paper, Principal Component Analysis (PCA) is applied to the problem of Online Handwritten Character Recognition in the Tamil script. The input is a temporally ordered sequence of (x,y) pen coordinates corresponding to an isolated character obtained from a digitizer. The input is converted into a feature vector of constant dimensions following smoothing and normalization. PCA is used to find the basis vectors of each class subspace and the orthogonal distance to the subspaces used for classification. Pre-clustering of the training data and modification of distance measure are explored to overcome some common problems in the traditional subspace method. In empirical evaluation, these PCA-based classification schemes are found to compare favorably with nearest neighbour classification.

1. Introduction

Online Handwriting Recognition [1, 3] refers to the problem of interpretation of handwriting input captured as a stream of pen positions using a digitizer or other pen position sensor. Normally the input stream is segmented into smaller units ("characters") suitable for direct classification either as part of a divide-and-conquer strategy, or using constraints imposed by the application (e.g., hand-printing in boxes).

In this paper we address the problem of classification of these isolated handwritten characters of the Tamil script using Principal Component Analysis as the basic technique. Our formulation of the problem features 156 Tamil "characters" corresponding to vowels, consonants, vowel-consonant combinations and special symbols.

The paper is organised as follows. Section 2 describes the steps of pre-processing and feature extraction that result in a feature vector of fixed dimensions being derived from the input pen-position stream. Section 3 addresses pattern classification using PCA and introduces pre-clustering and a modified distance measure as means of overcoming some limitations of the classical technique. Experimental results from these techniques on real handwriting data are presented in Section 4. The paper concludes with a summary of findings.

2. Pre-processing and feature extraction

The input from the digitizer corresponding to a handwritten character is a sequence of points of the form $[x_i, y_i]$ with embedded pen-up and pen-down events when multiple strokes are involved. Pre-processing is required in order to compensate for variations in time and scale, and can be classified into two steps – smoothing and normalization. Normalization in turn generally includes compensating for variability in character size and pen velocity.

Smoothing is performed to reduce the amount of high-frequency noise in the input resulting from the digitizer or tremors in writing. Typically, most of the information content in the signal is in the low frequency range. In our scheme, each stroke is smoothed independently using a 5-tap Gaussian low-pass filter with coefficients:

$$w_n = \frac{e^{-\frac{n^2}{2\sigma^2}}}{\sum_{n=-(N_T-1)/2}^{(N_T-1)/2} e^{-\frac{n^2}{2\sigma^2}}} \quad (1)$$

where N_T is the number of taps of the filter (chosen as five) and σ is selected such that $3\sigma = (N_T - 1)/2$.

Special care is taken to preserve the end points. The end points are duplicated $(N_T - 1)$ times before applying the smoothing filter so that the end points in the input and smoothed output are the same. This is possible because the filter coefficients add to one.

To eliminate variability due to translation and compensate for size differences, the characters are centered and rescaled. For rescaling, the bounding box of the character is computed and transformed to a fixed size using the linear mapping

$$x_i^{rescaled} = \frac{(x_i - x_{min})}{(x_{max} - x_{min})} \quad (2)$$

$$y_i^{rescaled} = \frac{(y_i - y_{min})}{(y_{max} - y_{min})} \quad (3)$$

This maps both x and y coordinates to the [0,1] range.

Re-sampling is performed to obtain a constant number n_P of points for all characters that are *uniformly sampled in space*, whereas the input data is the result of uniform sampling in time. The total length of the trajectory is computed for each character by adding the Euclidean distances between successive points. This is divided by the number of intervals[1] required after re-sampling to find the desired spacing between successive points in the resampled data. The original points are replaced with a new set at this constant spacing using piece-wise linear interpolation.

When a character has multiple strokes, each stroke is re-sampled separately such that the total number of points = n_P, using the following technique. All training characters having the same number of strokes are considered as a set. The number of points in each stroke is made proportional to the average length of strokes obtained from the corresponding set.

The result of pre-processing is a new sequence of points $[x_i, y_i]$ of constant length n_P, centered at the origin, of constant scale and regularly spaced in arc length. A feature vector is constructed from this sequence as $[x_0, \ldots, x_{n_P}, y_0, \ldots, y_{n_P}]$ and used directly for classification. Note that the feature vector obtained preserves the order in which strokes are written, and that the x and y components are not explicitly distinguished.

3. Pattern classification

In this section, we describe the classification of the pre-processed character into one of M character classes using Principal Component Analysis. It is assumed that training data is available for each class and feature vectors (which we will call training vectors) are extracted from them using the techniques described in the previous section.

3.1. PCA-based classification [2]

Let the N training vectors of a particular class be $\{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$. The correlation matrix is defined as

$$\mathbf{R_x} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x_i} \mathbf{x_i'} \tag{4}$$

For finding the principal components, we solve the eigenvalue equation

$$\lambda \mathbf{v} = \mathbf{R_x} \mathbf{v} \tag{5}$$

In this fashion, the basis vectors for each class i are computed as a set of N eigenvectors $\mathbf{v_j^i}$, $j = 1, \ldots, N$. Each eigenvector is normalized so that the basis is orthonormal. The distance measure of a test vector $\mathbf{x_{test}}$ from a class i is its orthogonal distance $D_i^{\perp}$ from the subspace defined by the basis vectors of class i, and computed as

$$D_i^{\perp 2} = \|\mathbf{x_{test}}\|^2 - \|\mathbf{\hat{x}_{test}^i}\|^2 \tag{6}$$

where $\mathbf{\hat{x}_{test}}$ is the projection of $\mathbf{x_{test}}$ into the class subspace. Since the basis vectors (eigenvectors) are orthogonal, the projection into the subspace is the sum of projections along the basis vectors, which can be computed as the dot product of the corresponding basis vectors with the test vector:

$$\mathbf{\hat{x}_{test}^i} = \sum_{j=1}^{N} \mathbf{x'_{test}} \mathbf{v_j^i} \tag{7}$$

Finally, the test vector is assigned to the "nearest" class, i.e., the output class label is assigned as C if $D_i^{\perp 2}$ is minimum:

$$C = \arg\min_j \left(\|\mathbf{x_{test}}\|^2 - \|\mathbf{\hat{x}_{test}^j}\|^2 \right) \tag{8}$$

Since $\|\mathbf{x_{test}}\|^2$ does not depend on the subspace to which projection is done, we can write

$$C = \arg\max_j \|\mathbf{\hat{x}_{test}^j}\|^2 \tag{9}$$

In general, a subset of the full set of eigenvectors corresponding to the largest eigenvalues are used for classification. This amounts to approximating the feature vector in a lower-dimensional space. Normally the smallest eigenvalues correspond to spurious variations in the character. Therefore, selecting a subset of the original subspace increases the accuracy of classification. The choice of this subset for our problem was empirically made and is described further in Section 4.

3.2. PCA with pre-clustering

The robustness of the PCA based classification method arises from the modeling of a class as a subspace. A consequence of this is that whenever we have a core pattern and its variations, all the linear combinations of these patterns are treated as members of the class. This is equivalent to synthesizing patterns by taking linear combinations.

This concept of synthesizing patterns assumes a single writing style, which is not true in the general multiple writer scenario [**?**]. Of particular concern is variation in stroke order which has significant impact on the feature vector as currently defined. In order to address these issues, we performed clustering within each class to identify "sub-classes" within the class corresponding to different writing styles. k-means clustering is used, keeping k as a constant for all classes. Each sub-class obtained is treated as a separate class from the standpoint of applying PCA and the same distance measure is used for classification of the test pattern.

[1] Number of intervals will be one less than number of points = $(n_P - 1)$.

3.3. PCA with modified distance measure

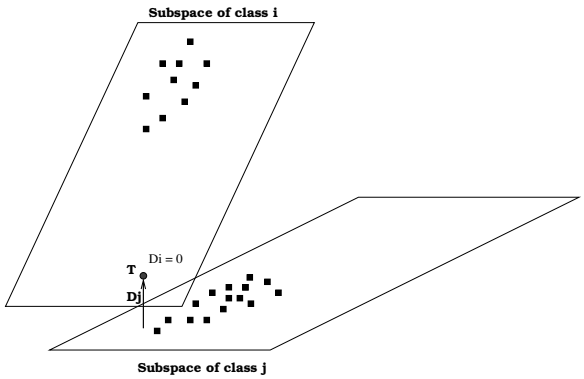

Figure 1. Misclassification with orthogonal distance measure

The major drawback with the orthogonal distance measure is that it does not consider any distortion within the subspace. This can lead to misclassification of some patterns, as illustrated by Figure 1. Although the test sample T is closer to all the training samples of class j than any of class i, it gets classified as class i since $D_i^{\perp} < D_j^{\perp}$. To overcome this, we define a new distance measure that takes into account the distortion within the subspace in addition to the orthogonal distortion.

Each class is modeled as a separable uniform distribution on the subspace. A rectangular distribution is assumed, and the parameters are found by their maximum likelihood estimates.The uniform distribution for a given class is given by

$$p(\mathbf{x}) = \prod_{i=1}^{d} \frac{1}{(x_H^i - x_L^i)}\, \mathcal{U}(x^i - x_L^i)\mathcal{U}(x_H^i - x^i) \qquad (10)$$

where d is the dimension of the vector, x^i is the i^{th} coordinate of the vector $\mathbf{x}$, x_L^i and x_H^i are the lower and upper boundaries of the distribution and $\mathcal{U}$ is the unit step function. These parameters are estimated by

$$x_L^i = \min_n(x_n^i) \qquad (11)$$

$$x_H^i = \max_n(x_n^i) \qquad (12)$$

$$n = 1, \ldots, N$$

N is the number of training vectors for the class.

The distortion within the subspace may be modeled as the distance of the projection of the test point from the boundary of the distribution. We then compute the "modified distance measure" of the test pattern from the subspace as the hypotenuse of the triangle illustrated in Fig. 2.

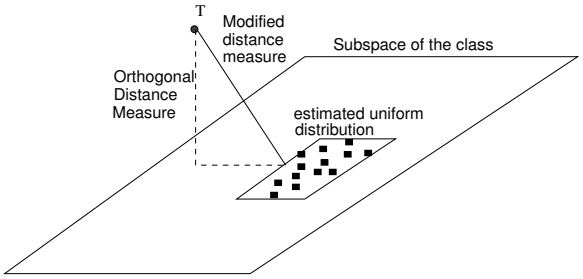

Figure 2. Modified Distance Measure

The test pattern is assigned to the class which minimizes the modified distance measure. Figure 3 illustrates how the use of this measure corrects the mis-classification seen earlier.

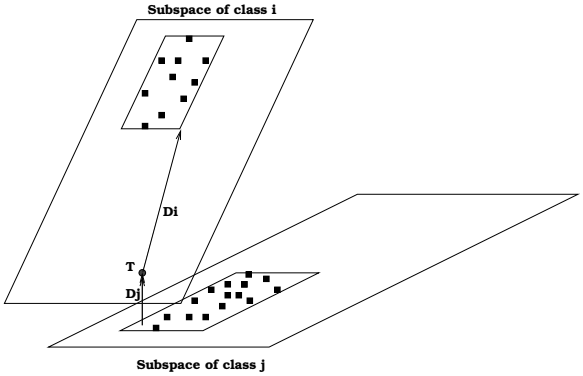

Figure 3. Classification with modified Distance Measure

It is clear that PCA with the modified distance measure can be used both with and without the pre-clustering in the original feature space described earlier.

4. Experimental evaluation

In this section, the different PCA-based schemes proposed in the earlier section are evaluated for the isolated Tamil handwritten character recognition task. A database of isolated Tamil characters were collected from a set of 15 writers using a custom application running on a PocketPC (Compaq iPAQ 3850). Ten samples of each symbol were collected from each writer for each of the 156 symbols under consideration.

In order to characterize writer-independent accuracy of the system (i.e., the performance of the system on the writing of a new writer), we used the following "leave-one-out" procedure. For each of the 15 writers, the accuracy for that

writer was computed by training on the $(14 \times 156 \times 10)$ samples from the other 14 writers, and testing on the (156×10) samples of the "target" writer. The mean accuracy of a given classification scheme was computed by averaging the accuracies computed as above across all 15 writers.

The different PCA-based classification schemes were evaluated and compared with the Nearest Neighbor classifier. Figure 4 shows the variation of mean accuracy of these schemes with change in dimensions of the modeling subspace (i.e., for different subsets of eigenvectors corresponding to the largest eigenvalues). In general, the mean accuracy peaks at a certain number of dimensions and then diminishes or saturates.

It is interesting to note that pre-clustering produces significant improvements in accuracy for smaller dimensions, but not for higher ones, perhaps because PCA is able to model different writing styles in these dimensions. The use of the modified distance measure appears to generally and significantly improve the base scheme.

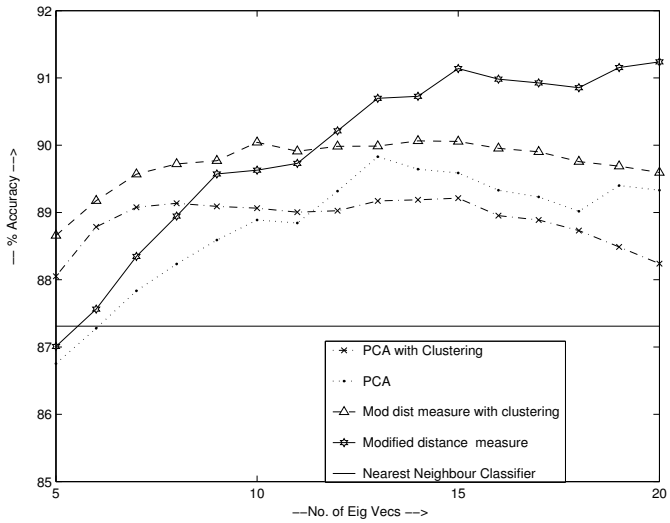

Figure 4. Accuracy of classification schemes vs number of basis vectors used

Table 1 compares the performance of the different schemes for different writers.

5. Conclusions

In this paper, PCA-based classification is applied to the problem of online recognition of isolated handwritten characters. The stream of points from the digitizer is smoothed and normalized and a feature vector extracted. Standard PCA is used to reduce dimensionality of each class and the orthogonal distance to the class subspace used for classification. Pre-clustering and modification of the distance measure are explored as ways of addressing specific problems with PCA. The proposed modifications are found empirically to improve recognition accuracy, and the resulting schemes found to compare favorably with the conventional Nearest Neighbor classifier.

A noteworthy aspect of these methods is that adding or replacing a training pattern of a specific class requires re-computation of the principal components of only that class. Another salient point is that although tested on Tamil, no language or script specific features are used in the pre-processing, feature extraction or classification, making these methods widely applicable for the recognition of other scripts.

Table 1. Comparison of different schemes(Number of Eigen vectors used = 15)

user	without clustering		with clustering		nearest neighbour classifier
	classical PCA	mod. dist. measure	classical PCA	mod. dist. measure	
1	94.87	95.30	96.41	95.71	80.64
2	95.09	95.72	96.02	96.60	92.63
3	95.30	95.51	95.83	96.35	85.58
4	92.09	92.73	94.68	95.25	82.24
5	80.34	83.55	80.00	81.60	88.33
6	91.67	93.59	90.06	90.58	94.62
7	82.26	85.04	85.06	84.29	81.60
8	87.39	88.88	89.94	91.22	90.45
9	91.67	93.16	86.28	86.73	92.05
10	92.74	94.23	91.54	93.14	90.58
11	89.32	92.74	90.71	92.44	94.87
12	93.38	93.80	92.31	93.84	90.26
13	82.05	83.33	76.03	77.63	76.79
14	86.32	88.88	84.74	86.03	81.99
15	89.32	90.60	88.59	89.42	87.05

References

[1] C. C.Tappert, C. Y.Suen, and T. Wakahara. The state of art in online handwriting recognition. *IEEE Transaftions on Pattern Analysis and Machine Intelligence*, 12(8):787–807, August 1990.

[2] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons INC., NY, 2000.

[3] T. Wakahara, H. Murase, and K. Odaka. On-line handwriting recognition. *Proceedings of IEEE*, 80(7):1181–1194, July 1992.