Two Dimensional Principal Component Analysis for Online Tamil Character Recognition

Suresh Sundaram , A G Ramakrishnan
Indian Institute of Science ,Bangalore, India
suresh@ee.iisc.ernet.in, ramkiag@ee.iisc.ernet.in

Abstract

This paper presents a new application of two dimensional Principal Component Analysis (2DPCA) to the problem of online character recognition in Tamil Script. A novel set of features employing polynomial fits and quartiles in combination with conventional features are derived for each sample point of the Tamil character obtained after smoothing and resampling. These are stacked to form a matrix, using which a covariance matrix is constructed. A subset of the eigenvectors of the covariance matrix is employed to get the features in the reduced sub space. Each character is modeled as a separate subspace and a modified form of the Mahalanobis distance is derived to classify a given test character. Results indicate that the recognition accuracy using the 2DPCA scheme shows an approximate 3% improvement over the conventional PCA technique.

Keywords: Principal Component Analysis (PCA), 2DPCA, Mahalanobis Distance.

1. Introduction

In an online handwriting recognition system, a methodology is developed to recognize the writing when a user writes on a pressure sensitive screen using a stylus that captures the temporal information. Online handwritten script recognition engines exist for languages like Latin [1], Chinese [2] and Japanese [3]. However, little attention has been devoted to develop similar engines for Indian languages.

In this paper, we attempt to evolve an online recognition system for Tamil characters using a technique called two dimensional Principal Component Analysis (2DPCA). Tamil is a classical South Indian language spoken by a segment of the population in countries such as Singapore, Malaysia and Sri Lanka apart from India. The Tamil alphabet comprises of 247 letters (consonants, vowels and consonant vowel combinations). Each letter is represented either as a separate symbol or as a combination of discrete symbols, which we refer to as 'characters' in this work. Only 156 distinct characters are sufficient to recognize all the 247 letters [4]. Samples of each of these characters form a separate class.

As far as the work on online handwriting recognition for Tamil is concerned, Niranjan et al. [5] have proposed elastic matching schemes. Dimensionality reduction techniques like Principal Component Analysis [6] have also been employed for recognition.

In this work, we propose an adaptation of the 2DPCA technique [7] for character feature extraction in a reduced subspace. Each of the 156 classes is separately modeled as a subspace. Contrary to the conventional PCA, the 2DPCA operates on matrices rather than 1D vectors. A set of local features (basically a novel set of features combined with conventional features) are derived for each sample point of the preprocessed character. The features corresponding to a sample point are stacked to form the rows of a matrix, referred to as the character matrix in this work. A covariance matrix of a significantly smaller size as compared to the one obtained in PCA is constructed from the character matrix. In order to represent the features in a reduced subspace, we project the character matrix onto a subset of the eigenvectors of the covariance matrix. For the classification of a test character, we have employed a modified form of the Mahalanobis / Euclidean distance.

To the best of our knowledge, there have been no attempts in the literature of applying the 2DPCA technique to the context of online character recognition till date. Most of the applications for which this technique has been proposed have been image-based such as face recognition [7].

2. Preprocessing

Prior to feature extraction and recognition, the input raw character is smoothened to minimize the effect of noise. The character is then resampled to obtain a constant number of points uniformly sampled in space following which it is normalized by centering and rescaling [6].

3. Feature Extraction

Let the number of sample points in the preprocessed character be N_p. At each sample point (x_i, y_i) for $1 \leq i \leq N_p$ of the resampled character, we extract a set of local features described in Section 3.1. Let F_j^i represent the j^{th} feature derived from the i^{th} sample point of the character. This notation has been adopted here merely to index the features and not to assign any weightage to them.

In case of multistroke characters, we concatenate the strokes into a single stroke, retaining the stroke order, before feature extraction.

3.1 Local Features

- **Normalized x-y coordinates**: The normalized x and y coordinates of the sample point are used as features and are denoted by F_1^i and F_2^i.
- **Radial Distance and Polar Angle**: The radial distance and angle in radians of the sample point with respect to the centroid of the character are computed to form two features F_3^i and F_4^i.
- **Radial distance and polar angle from the segment mean:** We find the length of the preprocessed character and divide it into 4 segments. Samples lying within a segment are used to compute the mean for that segment. The radial distance and polar angle of the sample point of the character under consideration is computed as follows: when it lies in segment k, $(1 \leq k \leq 4)$ its distance and angle from the mean of that segment is the feature F_5^i and $F_{6.}^i$
- **Polynomial fit coefficients:** At every sample point, we intend to relate its position with respect to its immediate neighbors. In order to exploit this local property, we take a sliding window of size M (M odd) centered on the sample point and perform an Nth order polynomial fit on the samples within the window using numerical techniques. We use the resulting N+1 polynomial coefficients as the features. For our work, we take M=3, N=2 (quadratic fit) and accordingly denote the features as F_7^i, F_8^i and F_9^i.
- **Autoregressive (AR) Coefficients**: We separately model the x and y coordinates of the sample point by two Nth order **autoregressive (AR) processes** and use the resultant AR coefficients also as features. We employ a 2nd order AR process and accordingly obtain the features F_{10}^i, F_{11}^i, F_{12}^i, F_{13}^i, F_{14}^i and F_{15}^i.

It is to be explicitly stated that for obtaining the polynomial and AR coefficients of the first and last sample points of the character, we assume that the last sample point of the last stroke is connected to the first sample point of the first stroke. Such a connection ensures that the notion of neighborhood is not lost while computing the polynomial fit features for the first and last sample point of the character.

The set of 15 features obtained at a sample point (x_i, y_i) are concatenated to form a feature vector FV^i of size 1 X 15.

$$FV^i = \left[F_1^i \; F_2^i \; \dots \; F_{15}^i \right] \tag{1}$$

We then construct a matrix C (referred to as the "character matrix" in this work) by stacking the feature vectors of the sample points of the preprocessed character.

$$C = \begin{bmatrix} FV^1 \\ FV^2 \\ \dots. \\ FV^{N_p} \end{bmatrix} \tag{2}$$

It can be observed that the i^{th} row of the character matrix C corresponds to the feature vector derived for the i^{th} sample point. Therefore the size of matrix C is $N_p \times 15$.

4. The 2DPCA Technique

The main principle behind the 2DPCA method lies in projecting the character matrix C onto an 15 dimensional projection vector X to yield a N_p dimensional feature vector Y. We refer to Y as the projected feature vector or the principal component vector.

$$Y = CX \tag{3}$$

The best projection vector X is the direction along which the total scatter of the projected samples is maximum. The total scatter of the projected samples can be characterized by the trace of the covariance matrix S_Y of the principal component vectors. Accordingly we seek to find the direction X for which the criterion $J(X)$ is maximized.

$$J(X) = trace(S_Y) \qquad (4)$$

It has been shown in [7] that the projection vector X that maximizes the criterion $J(X)$ is the eigenvector corresponding to the largest eigenvalue of the character covariance matrix G_t defined below.

$$G_t = \frac{1}{M}\sum_{j=1}^{M}(C_j - \bar{C})^T (C_j - \bar{C}) \qquad (5)$$

It is to be borne in mind that we attempt to model each character as a separate subspace. Accordingly, one can interpret $C_1, C_2, ..., C_M$ to be the M training character matrices of a particular class and $\bar{C}$ as the mean character matrix of that class. It can be easily verified that for our work, the size of the character covariance matrix G_t is 15 × 15.

However, in actual practice, we select a set of d projection axes $\{X_1, X_2, ..., X_d\}$ subject to being orthonormal to one another and maximizing the criterion $J(X)$. These projection axes turn out to be the orthonormal eigenvectors of G_t corresponding to the first largest d eigenvalues.

On applying the proposed 2DPCA technique to the character matrix C, we get a family of principal component analysis vectors $\{Y_1, Y_2 ..., Y_d\}$ as defined below

$$Y_p = CX_p, \quad p = 1, 2...d \qquad (6)$$

For the case where $d < 15$, a subset of the eigenvectors of the covariance matrix G_t is employed to get the features in the reduced subspace

The d principal component vectors can be stacked column-wise to form an $N_p \times d$ matrix B referred to as the character feature matrix.

$$B = [Y_1 \; Y_2 ...... \; Y_d] \qquad (7)$$

If instead of the 2DPCA technique, the PCA is used for feature extraction [6], we first concatenate the columns of matrix C to form an $15 \; N_p$ dimensional feature vector. We then use the eigenvectors corresponding to the d (where $d <= 15 \, N_p$) largest eigenvalues of the character covariance matrix as the projection axes. The size of the covariance matrix in the PCA is $(15 \; N_p) \times (15 \; N_p)$ which is very large compared to the 15×15 covariance matrix G_t in the 2DPCA method. The significantly smaller size of G_t in turn speeds up the feature extraction process in the 2DPCA technique compared to the PCA.

5. Classification Scheme

Assume that we have M training samples of a class (character) ω_c. After transformation by 2DPCA, we obtain M feature matrices of the form

$$B_c^k = [Y_{1c}{}^k \; Y_{2c}{}^k \; .... \; Y_{dc}^{\;k}] \qquad k = 1, 2, ..M \qquad (8)$$

From Eq. 8, we can interpret $\{Y_{ic}{}^k\}$ as the set of i^{th} principal component vectors corresponding to the M training samples of the class ω_c. These principal component vectors have been obtained by projecting the M character matrices $C_1, C_2, ..., C_M$ onto the eigenvector corresponding to the i^{th} largest eigenvalue of the character covariance matrix G_t defined in Eq. 5.

We assume that the set of N_p dimensional principal component vectors $\{Y_{ic}{}^k\}$ are drawn independently from a multivariate Gaussian probability distribution function of the form [8]:

$$p(Y_{ic}) = \frac{1}{(2\pi)^{\frac{N_p}{2}} \left|\sum_{ic}\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(Y_{ic} - \bar{Y}_{ic})^T \sum_{ic}^{-1} (Y_{ic} - \bar{Y}_{ic})} \qquad (9)$$

where $\bar{Y}_{ic} = \frac{1}{M} \sum_{k=1}^{M} Y_{ic}^k$

and $\Sigma_{ic} = \frac{1}{M}\sum_{k=1}^{M}(Y_{ic}^{k} - \overline{Y}_{ic})\ (Y_{ic}^{k} - \overline{Y}_{ic})^{T}$

are the estimated mean vector and covariance matrix of the i^{th} principal component vectors of the class ω_c. Eq. 9 gives the likelihood of the i^{th} principal component vector Y_{ic} for the given class ω_c.

For simplicity, we make an assumption that any set of principal component vectors of class ω_c, $\{Y_{mc}^{k}\}$ and $\{Y_{nc}^{k}\}$ $(m \neq n)$ are independent of each other. Therefore, using this we can write the likelihood of the principal component vectors in the subspaces in which they lie as:

$$p(B_c) = \prod_{i=1}^{d} p(Y_{ic}) \qquad (10)$$

Using Eq. 9 we can write

$$p(B_c) = c'\ e^{-\frac{1}{2}\sum_{i=1}^{d} (Y_{ic} - \overline{Y}_{ic})^{T}\ \Sigma_{ic}^{-1}(Y_{ic} - \overline{Y}_{ic})} \qquad (11)$$

where $c' = \dfrac{1}{(2\pi)^{\frac{N_p d}{2}} \prod_{i=1}^{d} (|\Sigma_{ic}|)^{\frac{1}{2}}}$

and $B_c = [Y_{1c}\ Y_{2c}\ ....Y_{dc}]$

Let $\omega_1, \omega_2 ... \omega_{156}$ be the labels of the classes corresponding to the 156 Tamil characters. Given a test character, we can now construct a feature matrix of the form

$$B_c^{test} = [Y_{1c}^{test}\ Y_{2c}^{test} ......\ Y_{dc}^{test}] \quad c = 1,2...156 \qquad (12)$$

by projecting it to each of the 156 subspaces using the 2DPCA. B_c^{test} refers to the feature matrix obtained by projecting the test character onto the subspace of class ω_c. Using Eq. 11 we see that

$$p(B_c^{test}) = c'\ e^{-\frac{1}{2}\sum_{i=1}^{d} (Y_{ic}^{test} - \overline{Y}_{ic})^{T}\ \Sigma_{ic}^{-1}(Y_{ic}^{test} - \overline{Y}_{ic})} \qquad (13)$$

The test character is assigned the class ω_{test} for which the following condition is satisfied.

$$\omega_{test} = \arg\max_c\ p(B_c^{test}) \qquad (14)$$

It can be readily verified from Eq. 13 that we assign the test character to the class for which the modified Mahalanobis distance is minimized.

Let

$$D_c = \sum_{i=1}^{d}(Y_{ic}^{test} - \overline{Y}_{ic})^{T}\ \Sigma_{ic}^{-1}(Y_{ic}^{test} - \overline{Y}_{ic}) + \log|\Sigma_{ic}| \qquad (15)$$

then we can write

$$\omega_{test} = \arg\ \min_c D_c \qquad (16)$$

On the other hand, if we employ a simple nearest neighbor Euclidean distance based approach [7] for the classification of the test data, we first compute the distance of the test character to its closest training sample in the subspace of class ω_c as

$$D_c = \min_j \sum_{m=1}^{d} \left\| Y_{mc}^{test} - Y_{mc}^{j} \right\|_2 \quad j = 1,2...M \qquad (17)$$

Given the set of distances $\{D_1, D_2, ..., D_{156}\}$, the test character is assigned the class ω_{test} for which the following condition is satisfied.

$$\omega_{test} = \arg\ \min_c D_c \qquad (18)$$

6. Experiments and Results

Data base of Tamil characters was collected from 15 native Tamil writers using a custom application running on a tablet PC. Each writer input 10 samples of each of the 156 distinct characters. To avoid the problem of segmentation, users wrote each character in a bounding box. We present our results for the writer independent scenario, with each class comprising of 150 samples. The characters are resampled to 60 points and normalized to [0, 1].

In the first experiment, we compute the features listed in Section 3 and construct the character matrices of size 60 × 15. We then extract the features in a lower dimensional subspace by performing the 2DPCA algorithm on the training samples of each class separately. The size of the character covariance matrix used to derive the projection axes is 15 × 15. On applying the algorithm, we get a different subspace for each Tamil character. Given a test character, we form its character matrix and project it on to each of the 156 subspaces. The test character is assigned to the subspace for which the modified Mahalanobis distance (Eq. 16) is least.

As our second experiment, to compare the performance of the 2DPCA algorithm against the existing PCA technique, all features obtained at each sample point of a training character are concatenated to form a 900 length feature vector. PCA is performed on the pooled training samples and a simple nearest neighbor classifier is used to classify the test character in the projected space.

It is to be noted that the size of the character covariance matrix

$$R_x = \frac{1}{N_T}\sum_{i=1}^{N_T}(x_i - \overline{x})(x_i - \overline{x})^T \qquad (19)$$

used to derive the projection axes in the PCA is 900 × 900, which is 60 times larger than the size of the covariance matrix G_t used in the 2DPCA algorithm. Herein lies the advantage of the 2DPCA algorithm over the PCA-the significantly smaller size of the character covariance matrix G_t in 2DPCA enables the extraction of features much faster while at the same time providing an improvement in recognition accuracy over the PCA. The set $\{x_1, x_2, .., x_{N_T}\}$ in Eq. 19 are the concatenated 900-dimensional feature vectors of the pooled training samples and $\overline{x}$ is the mean feature vector.

Tables 1 and 2 respectively present the recognition accuracy and time taken (in secs) for feature extraction by employing the 2DPCA and PCA techniques for varying number of training samples per class. The values in the parantheses in Table 1 indicate the number of eigenvectors required for retaining 98% of the total scatter of projected samples in the 2DPCA. The recognition accuracy using the 2DPCA is better than that of the conventional PCA for the case where eigenvectors of R_x are chosen so that at least 98% of the total variance is retained. (Table 1).

We also evaluated the performance of the 2DPCA algorithm on the IWFHR 2006 Tamil Competition Dataset [4] and found that top recognition accuracy using the 2DPCA shows an improvement of up to 3% as against the conventional PCA (Table 3). For this dataset, we used a nearest neighbor approach for classification of the test sample (Eq. 18).

Some misclassifications that occur in both the algorithms can be attributed to the visual similarities of a few characters and therefore in such cases, the distance metrics may not be powerful enough to track variations that make these characters distinct.

It is worth mentioning that though the 2DPCA method has been tested on Tamil characters in this work, no script specific features have been used in the feature extraction step, thereby

Table 1. Comparison of recognition accuracy of the 2DPCA versus the PCA. Value in parentheses indicate the number of eigenvectors used to retain 98% of the total scatter of projected samples in 2D PCA and 98% of the total variance in PCA.

#of Training Samples per class	40	80	120
2DPCA (Mahalanobis metric)	81.2% (8)	85.0% (8)	87.8% (8)
PCA	80.6% (248)	84.8% (250)	86.6% (248)

Table 2. Comparison of the feature extraction time (in secs) of the 2DPCA versus the PCA during training.

# of Training Samples per class	40	80	120
2DPCA	6.2	11.0	17.1
PCA	63.8	125.3	185.6

Table 3. Comparison of the top recognition accuracy of the 2DPCA versus the PCA on the IWFHR 06 Tamil Database [4].

2DPCA (Euclidean metric)	87.5% (8)
PCA	84.2% (267)

suggesting that the method can be applied to other scripts as well.

7. Conclusion and Future Work

In this paper we have attempted to apply the recently reported 2DPCA technique to the context of online character recognition. A set of local features is derived for each sample point of the preprocessed character to form the character matrix, using which a covariance matrix is constructed. The smaller size of the covariance matrix in 2DPCA makes the process of feature extraction much faster compared to the conventional PCA. Experimental results indicate that the recognition accuracy using the 2DPCA scheme shows an approximate 3% improvement over the conventional PCA technique while at the same time being more computationally efficient. Though the algorithm has been tested for recognizing Tamil characters, it can be widely used for the recognition of other scripts as well.

Further potential areas of research are to develop other dimensionality reduction schemes that take into account the class discriminatory characteristics for the online character recognition problem, to possibly improve the overall classification accuracy.

References

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

[2] Cheng-Lin Liu, Stefan Jaeger and Masaki Nakagawa. Recognition of Chinese Characters: The State-of-the-Art. *IEEE Trans.on Pattern Analysis and Machine Intelligence*, 26 (2), pp.188-213, 2004.

[3] S. Jaeger, C.-L. Liu and M. Nakagawa .The state of the art in Japanese online handwriting recognition compared to techniques in western handwriting recognition. *Intl Journal on Document Analysis and Recognition*, Springer Berlin 6 (2): pp. 75-88, October 2003.

[4] HP Labs Isolated Handwritten Tamil Character Dataset. http://www.hpl.hp.com/india/research/penhw-interfaces-1linguistics.html#datasets

[5] Niranjan Joshi, G Sita, A G Ramakrishnan and Sriganesh Madhavanath, Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition. *Proceedings of the 8th Intl Workshop on Frontiers in Handwriting Recognition* (IWFHR-8) pp 444-448, October 2004.

[6] Deepu V, Sriganesh Madhavanath and A G Ramakrishnan. Principal Component Analysis for Online Handwritten Character Recognition, *Proc. of the 17th Intl Conf .Pattern Recognition* (ICPR 2004) 2: pp 327-330, August 2004.

[7] Jiang Yang, David Zhang, Alejandro. F.Frangi and Jing yu Yang, Two Dimensional PCA: a New Approach to Appearance Based Face Representation and Recognition. *IEEE Trans. on Pattern. Analysis and Machine Intelligence*, 26 (1), pp.131-137, January 2004.

[8] Andrew Webb, *Statistical Pattern Recognition*. Second Edition. John Wiley and Sons Ltd, 2002