Learning Mixtures of Offline and Online features for Handwritten Stroke Recognition

Karteek Alahari Satya Lahari Putrevu C. V. Jawahar
Centre for Visual Information Technology, IIIT Hyderabad, INDIA.
jawahar@iiit.ac.in

Abstract

In this paper we propose a novel scheme to combine offline and online features of handwritten strokes. The state-of-the-art methods in handwritten stroke recognition have used a pre-determined combination of these features, which is not optimal in all situations. The proposed model addresses this issue by learning mixtures of offline and online characteristics from a set of exemplars. Each stroke is represented as a probabilistic sequence of substrokes with varying compositions of these features. The model adapts to any stroke and chooses the feature composition that best characterizes it. The superiority of the method is demonstrated on handwritten numeral and character strokes.

1. Introduction

Handwriting recognition finds its application in many situations like reading bank cheques, handwritten notes on PDAs, document retrieval, etc. [5, 6]. This problem has been addressed using offline [1, 4, 6] and online features [3, 5] independently, and also a combination of both features [7]. Offline features capture handwriting in the form of an image, while online features capture it as a time-sequential series of sensor positions [5]. Methods combining these features have shown considerable promise for recognizing handwritten strokes, but have a fundamental restriction. They assume that a pre-defined combination of offline and online features is appropriate for all the strokes in a dataset. This is not valid in general. For instance, when distinguishing numerals such as '0' and '6' offline features are more useful, while for distinguishing '5' and '6' online features are more useful.

We present an approach to address this issue; wherein the composition of offline and online features is learnt from a given set of strokes. Each stroke is represented as a probabilistic sequence of substrokes. The length of the substroke determines the composition of the two types of features, with the two extreme cases being: (i) the entire stroke representing the substroke, and (ii) each data point (*e.g.*, 2D coordinate) representing the substroke. The former case captures the offline nature of the stroke since the time-sequential characteristics are not captured, while the latter captures the online nature as a sequence of substrokes. The proposed method chooses the optimal combination between these two extreme cases, and represents each stroke using a set of probabilistic model components. Each component learns a mixture model of substrokes and determines an appropriate combination of the two features.

The remainder of the paper is organized as follows. Section 2 discusses the mixture of substrokes model and shows how it combines offline and online features. This mixture model is used to describe an adaptive scheme, which learns the feature composition, in Section 3. Section 4 presents the results on character and numeral data sets with a discussion. Concluding remarks are made in Section 5.

2. Mixture of Substrokes Model

The mixture of substrokes model represents each stroke as a probabilistic sequence of fundamental units called as *substrokes*. The model exploits the fact that many strokes in a given data set have common substrokes. As an example consider character strokes such as 'e', 'c', 'd', etc. It is evident that these strokes share a substroke which defines the curved segment in them. Similar observations can be made on other strokes. Following this observation, the mixture model represents the given data as a set of substrokes which are automatically learnt. Any stroke in the data set is characterized by a sequence of these substrokes probabilistically. Given multiple instances of strokes, the mixture model automatically extracts the substrokes, which constitute these strokes, and their sequencing information in order to generate the stroke.

This modeling is achieved using the Mixture of Factor Analyzers (MFA) model [2]. It is essentially a reduced dimension mixture of Gaussians, *i.e.* it identifies the commonalities in the data set (substrokes) as clusters in a low dimensional manifold. Once the substrokes are probabilistically estimated, a sequence of cluster transitions determines a stroke. To learn the substrokes and their sequencing, multiple features are extracted from each point in the stroke. They include chain codes computed using the position of a point with respect to its preceding point, the x_t and y_t coordinates of the point, and the angle. A feature vector x_t is constructed from all these features. The underlying generative model of the MFA model is given by $P(x_t) = \sum_{j=1}^{m} \int P(x_t|z_t, \omega_j) P(z_t|\omega_j) P(\omega_j) dz$, where

z_t is the low dimensional representation corresponding to x_t, ω_j denotes the j th mixture (substroke), and m is the number of mixtures. The low dimensional representation z_t is related to x_t as $x_t = \Lambda_j z_t + u$, for a given mixture j. The factor loading matrix Λ_j and the associated noise u, which is distributed according to $\mathcal{N}(0, \Psi)$, determine z_t. Given multiple instances (corresponding to handwritten data collected from multiple subjects) of the feature vector $x = [x_1, x_2, \ldots]^T$, the task is to determine the corresponding low dimensional vector $z = [z_1, z_2, \ldots]^T$, and the mixture each feature point belongs to. The generative process is inverted to estimate all the parameters $\{(\mu_j, \Lambda_j)_{j=1}^m, \pi, \Psi\}$, where π is the vector of adaptable mixing proportions, $\pi_j = P(\omega_j)$.

The parameters of the distribution are estimated using the Expectation Maximization (EM) algorithm [2]. It is a general method of finding the maximum likelihood estimate of the parameters of an underlying distribution from a given data set when the data has missing or unknown values. The EM algorithm has two stages, namely inference and learning, which are executed in succession till convergence. In these stages the algorithm alternates between inferring the expected values of the hidden variables, *i.e.* low dimensional representation and the substrokes keeping the parameters fixed, and estimating the parameters using the inferred values.

In the inference phase, the current estimates of the parameters are used to compute the expected values of the subspace representations and the substrokes. The expectations $E[\omega_j|x_t]$, $E[z_t|\omega_j, x_t]$ and $E[z_t z_t^T|\omega_j, x_t]$ are computed for all data points t and substrokes ω_j. These quantities are given by

$$\begin{aligned} E[\omega_j z_t|x_t] &= h_{tj}\beta_j(x_t - \mu_j), \\ E[\omega_j z_t z_t^T|x_t] &= h_{tj}(I - \beta_j\Lambda_j + \\ &\quad \Lambda_j(x_t - \mu_j)(x_t - \mu_j)^T\beta_j^T), \end{aligned} \tag{1}$$

where

$$\begin{aligned} h_{tj} &= E[\omega_j|x_t] = \pi_j\mathcal{N}(x_t - \mu_j, \Lambda_j\Lambda_j^T + \Psi) \\ \beta_j &= \Lambda_j^T(\Lambda_j\Lambda_j^T)^{-1}. \end{aligned} \tag{2}$$

Each $\mu_j, j = 1 \ldots m$, denotes the representative appearance of the corresponding substroke, $\Lambda_j, j = 1 \ldots m$, denotes the various subspace bases for the substrokes, π denotes the mixing proportions of substrokes in the stroke set, and Ψ is a measure of noise present in the data.

In the learning phase, the expected values of the subspace representations and the substrokes are used to get better estimates of the parameters. A linear system of equations is solved to compute the parameters $\pi_j, \Lambda_j, \mu_j, \Psi$. The exact equations can be easily derived from [2]. Each data point x_t is then assigned to the substroke c_t according to $c_t = \arg\max_j h_{tj}, j = 1 \ldots m$. Thus, each point is assigned to the substroke for which it has the maximum membership.

After the EM algorithm converges, a transition matrix T_k, which captures the sequencing of various substrokes, is constructed for each stroke k as follows

$$\tau_{pq}^k = \sum_{t=1}^{N-1} [c_t = p][c_{t+1} = q], \quad 1 \le p, q \le m. \tag{3}$$

The substroke transitions for successive points of the stroke k are represented by the entries in the transition matrix T_k. It encodes the temporal characteristics (online features) of the stroke. These matrices are normalized to denote the corresponding probability transition matrix. Given a new stroke, the learnt parameters are used to compute its probability transition matrix, and is assigned to the stroke that is most likely to generate this matrix.

The mixture of substrokes model represents each stroke as a sequence of substrokes. A substroke captures the offline characteristics of the stroke locally, while transitions between different substroke mixtures capture the online characteristics. The number of substrokes in the mixture model is fixed *apriori*. Thus, the mixture of substrokes model captures a *fixed* composition of offline and online features. We build on this model and describe an adaptive scheme that learns the feature composition.

3. Learning the feature composition

Consider the problem of recognizing handwritten English numerals as an example. In this set offline features are better suited for distinguishing numerals such as '0' and '6', and online features for numerals such as '5' and '6'. It is hard problem to determine a fixed composition of offline and online features that is appropriate for all the strokes in the data set. However, learning the composition in a stroke specific way addresses this issue. A collection of mixture models with varying compositions of these features is used to estimate the relevance of each component for a given stroke. To begin with, the offline and online features are extracted from the training set. Individual components, which use a mixture of these features, are then trained. A decision criterion is defined as a weighted combination of these individual components, as shown in Figure 1.

3.1. Estimating the component mixture models

Identification of substrokes is a critical step in finding the composition of online features for the recognition framework. The two extreme cases in finding the substrokes are: modeling with (i) a single substroke, and (ii) each point as a substroke. The problem is to define a model which chooses the appropriate combination of offline and online features for identifying handwriting strokes.

Let K denote the number of distinct strokes in the data set. To build a feature set based on completely offline features a mixture of substrokes model (refer Section 2) with a single mixture for each stroke, *i.e.* K mixtures, is used. Hence, no substrokes are identified in this case. Such purely

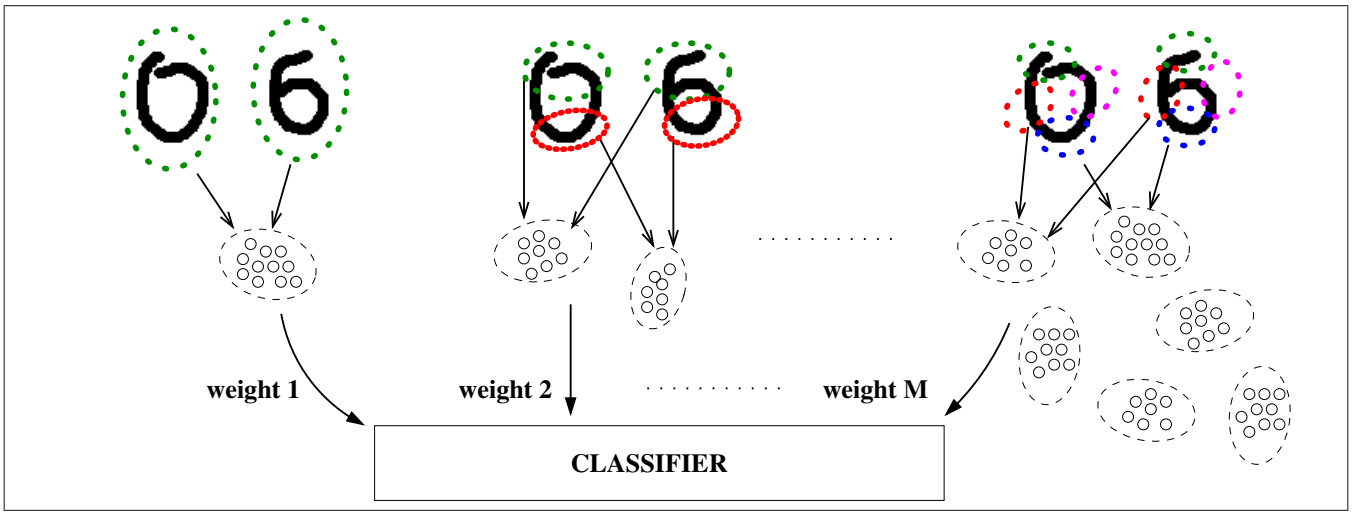

Figure 1. Summary of the proposed model. A mixture of MFA model components is used to let the model choose between offline, online and a combination of both features automatically. The contribution of each of these components in the decision making process is identified by its corresponding weight. The numerals '0' and '6' are used to illustrate the fact that they possess similar online, but different offline characteristics.

offline features are more useful when identifying numerals such as 0 and 6, which have similar curvature properties in the online feature space, as seen in Figure 1. Choosing a mixture model with $K+1$ mixtures leads to as many substrokes. The properties of these substrokes, such as curvature, length, transitions between the substroke clusters, etc., are captured in the feature set. In a similar fashion other mixture model components with increasing number of substrokes, which characterize different compositions of offline and online features, are learnt. Theoretically, one may define a single mixture for each data point in the stroke. However, such a scheme is impractical as due to the noisy estimates from a large number of transitions between substroke mixtures. The number of substrokes is typically decided by the nature of the data set and is much lower than the number of data points in a stroke.

At the end of this stage the feature sets corresponding varying compositions of offline and online features are learnt from the data set.

3.2. Relevance of mixture models

The contribution of the individual components, *i.e.* MFA components with varying number of substrokes, is weighed to obtain an optimal feature composition. The relevance computation is posed as an optimization problem. Let M be the number of MFAs trained for a set of N strokes. The objective function $J(.)$ is defined as

$$J(\Gamma) = \sum_{j=1}^{N}\sum_{i=1}^{M}(\gamma_{ij}d_{ij})^2,$$

where $\Gamma \in \mathbb{R}^{MN}$ is a matrix $[\gamma_{ij}]$. The weight γ_{ij} denotes the contribution of the i th MFA for the j th stroke in the data set, and d_{ij} is the distance metric signifying the cost of recognizing the j th sample with the i th MFA. This objective function is minimized over the space of γs using Lagrange multipliers with the constraint $\sum_{i=1}^{M}\gamma_{ij} = 1$.

Observing that the weights for each stroke are independent, the minimization can be done independently in each column. Thus, the Lagrangian is given by

$$\mathcal{J}(\lambda, \gamma_j) = \sum_{i=1}^{M}(\gamma_{ij}d_{ij})^2 - \lambda(\sum_{i=1}^{M}\gamma_{ij} - 1). \qquad (4)$$

Minimizing Equation 4 with respect to γ_{pq} gives $\gamma_{pq} = \lambda/2(d_{pq})^2$. Using the constraint $\sum_{r=1}^{M}\gamma_{rq} = 1$ with this equation eliminates λ, *i.e.*

$$\gamma_{pq} = 1 \Big/ (d_{pq})^2\sum_{r=1}^{M}(d_{rq})^2 \ . \qquad (5)$$

Equation 5 provides a method for estimating the weights, given the distance metric d_{ij}, which is chosen as the inverse of posterior probability $p(j|data, i)$. The posterior denotes the probability of identifying a sample $data$ as belonging to stroke j ($\in \{1, 2, \ldots, C\}$) for a given mixture model i.

3.3. Recognition

Once the weights $[\gamma_{ij}]$ are identified for all the classes, they are used in the recognition framework. Given a

new stroke S, the learnt parameters are used to compute the corresponding subspace representations, substroke assignments and probability transition matrices for each of the model components. A decision criteria based on the weighted sum of posterior probabilities, $p_j = \sum_{i=1}^{N} \gamma_{ij} p(j|S,i)$, is computed for each class of stroke j. The stroke S is labelled as j^* which maximizes the posterior according to $j^* = \arg\max_j p_j$.

4. Results

The dataset used in the experimentation consists of more than 1000 English numeral and character strokes collected from 3 different subjects using an IBM crosspad. The subjects' writing style is unconstrained. The database is divided into disjoint training (to estimate the substrokes and their corresponding weights) and testing (to evaluate the recognition performance) sets. The testing set comprised of over 250 strokes. The variability in the data due to translation of x_t and y_t coordinates is negated by computing a bounding box for each stroke and uniformly rescaling to the $0-1$ range. After this preprocessing, features such as chain codes (values ranging from one to eight), normalized x_t and y_t coordinates, angle, are extracted from each point in the stroke.

Character	Single MFA	Weighed MFA
u	90.00	96.67
w	96.00	96.00
c	90.91	95.45
e	92.86	96.43

Table 1. The average recognition accuracies (%) on a sample character set using a single mixture model and a weighted set of mixture models.

The model components which characterize the varying compositions of offline and online features were learnt from the training data set. The optimal number of substroke mixtures was found empirically for the numeral and character strokes independently, to determine the number of model components. However, it is to be noted that the recognition accuracy varied negligibly from the reported results beyond a certain number of components. The relevance of each component was estimated for all the stroke types. Strokes from the testing data set were recognized following the weighted decision criterion described in the previous section. Results of the experiments on some character and all the numeral strokes are summarized in Tables 1 and 2 respectively. The tables illustrate the average recognition accuracies using a single mixture model (Single MFA, refer Section 2), which captures a fixed offline and online feature composition, and an adaptive model (Weighted MFA), which learns the feature composition from the training set. It can be observed that in almost all the cases the Weighted MFA scheme outperforms the Single MFA scheme. Comparison with a standard HMM-based method also shows the superiority of our approach. The relatively low accuracies for certain character/numeral strokes is due to the fact that the datasets were neither fine tuned to achieve higher recognition nor excessively preprocessed to eliminate the noise. However, the relative improvement in accuracy is clearly evident in these cases as well.

Numeral	HMM	Single MFA	Weighed MFA
0	95.83	95.83	100.00
1	100.00	100.00	100.00
2	87.50	91.67	91.67
3	87.50	87.50	95.83
4	95.83	95.83	100.00
5	100.00	100.00	100.00
6	91.67	95.83	100.00
7	100.00	100.00	100.00
8	83.30	83.30	87.50
9	95.83	95.83	100.00

Table 2. The avg. recognition accuracies (%) using a HMM-based method, a single mixture model and a weighted set of mixture models for the numeral set.

5. Conclusion

The paper presents an adaptive scheme which learns the offline and online feature composition for a given set of strokes. It demonstrates the fact that a pre-determined combination of these features is not optimal in general. Furthermore, the model captures the features in a low dimensional manifold providing an efficient mechanism to store large collection of handwritten strokes. We believe this scheme finds many applications in building higher level handwriting recognition systems which work on words or sentences.

References

[1] A. Amin. Off-line arabic character recognition: Survey. *Proc. ICDAR*, pages 596–599, 1997.

[2] Z. Ghahramani and G. E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, Canada, 1996.

[3] K. Ishigaki and et al. Interactive Character recognition technology for pen-based computers. *Fujitsu Sci. Tech. Journal*, pages 191–201, 1999.

[4] L. Lam and C. Y. Suen. Structural classification and relaxation matching of totally unconstrained handwritten ZIP codes. *Pattern Recognition*, 19:15–19, 1986.

[5] R. Plamondon and S. N. Srihari. On-line and off-line handwriting recognition: comprehensive survey. *IEEE Trans. PAMI*, 22:63–84, 2000.

[6] O. Trier, A. K. Jain, and T. Taxt. Feature extraction methods for character recognition - A Survey. *Pattern Recognition*, 29(4):641–662, 1996.

[7] A. Vinciarelli and M. Perrone. Combining Online and Offline Handwriting Recognition. *ICDAR*, pages 844–848, 2003.