

Discriminant Substrokes for Online Handwriting Recognition

KartEEK Alahari Satya Lahari Putrevu C. V. Jawahar
Centre for Visual Information Technology
International Institute of Information Technology
Gachibowli, Hyderabad 500019. INDIA.
jawahar@iiit.ac.in

Abstract

A discriminant-based framework for automatic recognition of online handwriting data is presented in this paper. We identify the substrokes that are more useful in discriminating between two online strokes. A similarity/dissimilarity score is computed based on the discriminatory potential of various parts of the stroke for the classification task. The discriminatory potential is then converted to the relative importance of the substroke. Experimental verification on online data such as numerals, characters supports our claims. We achieve an average reduction of 41% in the classification error rate on many test sets of similar character pairs.

1. Introduction

With the widespread use of computers, the need for friendly man-machine interfaces is on the rise. Handwriting recognition forms an important component in building such interfaces [7]. In particular, online handwriting recognition approaches have received considerable research attention [4, 7, 9] recently. These approaches address the problem of interpreting the pen movement, which follows a sequential pattern over time. The success of online handwriting recognition schemes can be attributed to the availability of additional information such as the order of strokes.

Online handwriting recognition systems can be broadly divided into three categories: (1) Heuristic or Structure based methods (eg. Fuzzy rule-based schemes), (2) Template matching based methods (eg. DTW-based schemes), and (3) Statistical methods (eg. HMM, TDNN, SVM) [7]. In addition to the 2-dimensional point features available as a function of time, online systems may also use features such as velocity, pressure, etc., that are captured during writing. The temporal relations in online data are typically captured by mathematical models like HMMs, Linear Prediction, etc. [7, 9, 11], at the stroke or the substroke level.

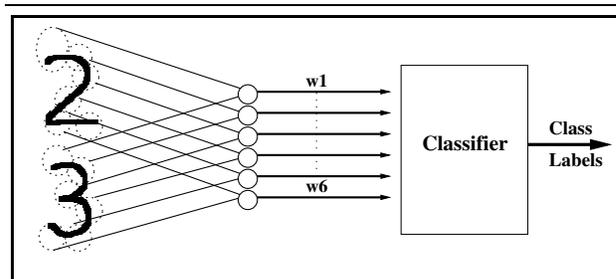


Figure 1. A summary of the discriminant-based classification framework. The substrokes (parts of the sequence circled with dotted lines) of the numerals are analyzed to find their corresponding discriminatory potential. These weights are combined with substroke matching scores to find a global decision criterion. For discriminating ‘2’ and ‘3’ the latter parts of the strokes are found to be more useful.

Most of these schemes also provide a compact representation for the online data. A recent trend has been to combine offline and online features for recognition, as they complement each other [6, 10]. Offline features (from images) mostly describe the spatial structure of the stroke, while online features provide the temporal ordering.

Popular online handwriting recognition approaches give equal importance to all parts of a stroke during matching, which may not be the best for all cases (Refer Figure 1). We need to detect the parts of a stroke (called substroke) that are more useful for the classification task. Our objective is to identify these substrokes and use this information for improving the performance of recognition schemes. Consider the problem of recognizing the numerals 2 and 3 (Figure 1). The two numerals appear to possess similar curvature properties at the beginning of the sequences. As the complete numbers begin to appear, their distinguishing characteristics unfold over time. In other words, the tail portion of the num-

bers is more useful for distinguishing them. We describe an approach to identify critical segments of the strokes. The individual parts are then weighed to obtain appropriate score for the final recognition.

2. Preliminaries

Online handwriting data needs efficient modelling schemes for building a compact representation by discarding the acceptable statistical variability. Often this is achieved by methods like Hidden Markov Models (HMMs), Linear Prediction, etc. Such modelling schemes exploit the inherent dynamism in online data. Of late, researchers have found modelling the substrokes in online data to be more useful for recognition [7, 9].

HMM is a doubly stochastic model characterized by the conditional probabilities of transitions between a set of hidden states. Online handwriting recognition schemes typically use a left-to-right HMM for each character. Substroke-based HMMs have also been proposed recently [9]. All these techniques recognize a given sequence by maximizing the a posteriori probability of the sequence.

Linear Prediction is another scheme for modelling sequential data [11]. For a sequence of N data points $\mathbf{X} = \{\mathbf{x}_i\}$, $i = 1, 2, \dots, N$, a p th order linear predictor relates a sample \mathbf{x}_i to its previous p samples as

$$\hat{\mathbf{x}}_i = a_1 \mathbf{x}_{i-1} + a_2 \mathbf{x}_{i-2} + \dots + a_p \mathbf{x}_{i-p}, \quad (1)$$

$i = (p + 1), (p + 2), \dots, N$, where $\hat{\mathbf{x}}_i$ denotes the prediction of \mathbf{x}_i . The coefficient vector $\mathbf{a} = [a_1, a_2, \dots, a_p]$ is estimated by minimizing the sum of squared errors $\sum_i \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|^2$. In the case of online handwriting data, the vector \mathbf{a} captures the temporal correlation among the samples of \mathbf{X} , which is an ordered sequence of (x, y) pen coordinates.

Principal Component Analysis (PCA) [3], a linear model based on eigenvectors corresponding to the dominant eigenvalues, is widely used for obtaining a low-dimension manifold of offline data [7]. In a mean squared error sense, PCA is an optimal dimensionality reduction method. Recently, it has been used for online handwriting data by extracting a feature set based on offline patterns (normalized 2-dimensional coordinates) [2]. Furthermore, PCA is believed to be suitable for representing the data, unlike discriminant based approaches which are appropriate for classification problem [1].

All these modelling schemes treat all parts of a sequence uniformly. To distinguish between the different parts, we need to weigh them appropriately. This is in the spirit of Discriminant Analysis and Statistical Pattern Recognition techniques.

3. A Weighted Measure for Online Strokes

Fisher Discriminant Analysis (FDA) is a commonly used variant of Discriminant Analysis techniques for 2-class problems [3]. It identifies an optimal direction ϕ along which the ratio of between class scatter and within class scatter is maximized. When the data points \mathbf{x}_i are projected onto this direction as $\phi^T \mathbf{x}_i$, each element of ϕ acts as a weight for the corresponding dimension of \mathbf{x}_i . In the lower dimension, the distance between two patterns \mathbf{x}_i and \mathbf{x}_j is expressed as a weighted linear combination of distances along each dimension, *i.e.*, $d(\phi^T \mathbf{x}_i, \phi^T \mathbf{x}_j) = \sum_k \phi_k d(x_i^k, x_j^k)$. To compute this vector ϕ , the criterion function for FDA, $J(\cdot)$, is defined as

$$J(\phi) = \frac{\phi^T \mathbf{S}_b \phi}{\phi^T \mathbf{S}_w \phi}, \quad (2)$$

where \mathbf{S}_w and \mathbf{S}_b are the within class and between class scatter matrices. It is shown that any vector ϕ which maximizes the Fisher criterion in Equation 2 satisfies $\mathbf{S}_b \phi = \lambda \mathbf{S}_w \phi$ for some constant λ [3]. This can be solved as an eigenvalue problem. The discriminant vector, ϕ is given by the eigenvector corresponding to the largest eigenvalue of $\mathbf{S}_w^{-1} \mathbf{S}_b$.

We present an approach, modelled on the similar lines of Fisher Analysis, to identify the substrokes with greater discriminatory potential. Given the strokes of two classes, our aim is to learn the representation for these classes and use it in a recognition framework as outlined below.

Training: Learn the substrokes with greater discrimination.

1. Align the online strokes and obtain equal number of substrokes in all samples.
2. Model the local continuity of the data points in each substroke to extract a set of features (say using HMM, LPC, splines, etc.).
3. In this feature space, identify the discriminant vector ϕ which provides an optimal weight with which the distinguishing characteristics of the two strokes are maximized.

Testing: Recognize an unknown stroke.

1. Align the test stroke in similar fashion as the training strokes and obtain the substrokes.
2. Model the substrokes according to the scheme used in training.
3. Using ϕ as a weight vector, which describes the discriminatory potential of each substroke, compute the dissimilarity score (or posterior probability, etc.) with respect to the two classes and recognize the test stroke as belonging to a class based on its score.

4. Discriminant Analysis of Online Data

Here we derive the technical details of the algorithm presented in the previous section. Consider two strokes \mathbf{A} and \mathbf{B} of classes \mathcal{A} and \mathcal{B} . Let $N_{\mathcal{A}}$ and $N_{\mathcal{B}}$ be the number of samples in these classes respectively. The training phase consists of three major steps – alignment, modelling and identification of relative weights of the sub-strokes.

4.1. Alignment of strokes

In most situations, identification of a simple model parameter is not valid for the entire stroke. Also, in recognition schemes (eg. HMM), where the model complexity is directly dependent on the number of distinct characters in the dataset, it is economical to define the model in terms of the basic repetitive units – sub-strokes [9].

We employ a DTW-based alignment for segmenting both the strokes to a fixed number of segments. DTW aligns a sequence of feature vectors using dynamic programming [5]. $D(p, q)$, the cost of aligning the sequences \mathbf{A} and \mathbf{B} , is computed using the recursive cost function $D(i, j) = \min\{D(i-1, j-1), D(1, j-1), D(i-1, j)\} + d(i, j)$, where $d(i, j)$ is the local cost in aligning the i th element of \mathbf{A} and the j th element of \mathbf{B} .

We compute the alignment score for all the strokes in the dataset with respect to a single stroke (called the template stroke). It is to be noted that alignment scheme and further processing steps are independent of the choice of the template stroke. To retrieve the alignment, we backtrack along the minimum cost path obtained for $D(p, q)$. At the end of the alignment scheme, we have a correspondence between the two strokes. We extract a fixed number of segments from the template stroke and pick the corresponding (aligned) sub-strokes from the others.

4.2. Modelling the sub-strokes

Let the k th segments of the two strokes be denoted by \mathbf{A}^k and \mathbf{B}^k , $k = 1, 2, \dots, s$, where s is the number of segments in \mathbf{A} and \mathbf{B} . Segmentation of aligned strokes provides a one-to-one correspondence between \mathbf{A}^k and \mathbf{B}^k , so that the global (total) recognition can be achieved with the help of individual ones across \mathbf{A}^k and \mathbf{B}^k and their sequencing information.

The sub-strokes are modelled appropriately to capture their temporal properties. This is done by mapping the substroke features into a new domain. We denote the model parameters of the k th sub-strokes as $\theta_{\mathcal{A}j}^k$ and $\theta_{\mathcal{B}j}^k$ for the j th sample. Thus the new feature set is given by $\Theta_{\mathcal{A}}^k = \{\theta_{\mathcal{A}j}^k\}_{j=1}^{N_{\mathcal{A}}}$ and $\Theta_{\mathcal{B}}^k = \{\theta_{\mathcal{B}j}^k\}_{j=1}^{N_{\mathcal{B}}}$. We describe the three modelling schemes used in our analysis as follows.

- One method of modelling sequences is by using Linear Prediction. For a linear predictor of p th order, $\theta_{\mathcal{A}j}^k$

given by $\theta_{\mathcal{A}j}^k = \mathbf{a}^k = [a_1^k, a_2^k, \dots, a_p^k]^T$ (from Equation 1).

- In DTW-based schemes, we do not consider any explicit modelling scheme, and use the features (in this case, a sequence of 2-dimensional points) directly for identifying substroke weights.
- In HMMs, we train the model using a chain-code representation of the online sub-strokes. After training, we get the model parameters $\{\Xi, A, B, \pi\}$, where A denotes the transition probabilities among the hidden states Ξ (chosen to be 5 in this case), B denotes the observation symbol probability for each state (with the observation symbols being the chain-code values 1 – 8), and π is the state distribution. Thus, the transition matrix A models the temporal nature of the sub-strokes.

4.3. Discriminating potential for Sub-strokes

We identify weights ϕ_k , $k = 1, 2, \dots, s$, for each substroke such that they have optimal distinguishing characteristics along the direction of the vector ϕ . We obtain this vector using a Fisher-like analysis, *i.e.*, we minimize the within class scatter and maximize the between class scatter for the online strokes. The scatter matrices are given by

$$\begin{aligned} \mathbf{S}_{\mathbf{w}} &= \sum_{i \in \{\mathcal{A}, \mathcal{B}\}} \sum_{j=1}^{N_i} (\theta_{ij} - \bar{\theta}_i)(\theta_{ij} - \bar{\theta}_i)^T \\ \mathbf{S}_{\mathbf{b}} &= (\bar{\theta}_{\mathcal{A}} - \bar{\theta}_{\mathcal{B}})(\bar{\theta}_{\mathcal{A}} - \bar{\theta}_{\mathcal{B}})^T, \end{aligned}$$

where the symbols without the superscript k denote the stroke features with sub-strokes stacked as rows and the mean over the samples of a class i is given by $\bar{\theta}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \theta_{ij}$. Also, $(\theta_{ij} - \bar{\theta}_i)$ is the distance measure defined in the representation space. Here, the $s \times s$ matrices $\mathbf{S}_{\mathbf{w}}$ and $\mathbf{S}_{\mathbf{b}}$ capture the within class and between class scatters at the substroke level. Each entry of $\mathbf{S}_{\mathbf{b}} = \{b_{ij}\}$ represents the variance between sub-strokes \mathbf{A}^i and \mathbf{B}^j over all samples. Maximizing the objective function in Equation 2 results in classes with large discriminating characteristics.

4.4. Recognition

Let \mathbf{T} be the stroke we are interested in recognizing. It is labelled as class i^* according to

$$i^* = \arg \min_{i \in \{\mathcal{A}, \mathcal{B}\}} D(\mathbf{T}, i), \quad (3)$$

where D defines the cost of recognizing the stroke \mathbf{T} as the stroke i . The matching cost $D(\mathbf{T}, \mathcal{A})$ is given by $D(\mathbf{T}, \mathcal{A}) = f(\phi_1, \dots, \phi_s, \theta_{\mathbf{T}}^1, \dots, \theta_{\mathbf{T}}^s, \theta_{\mathcal{A}}^1, \dots, \theta_{\mathcal{A}}^s)$. The function $f(\cdot)$ models the distance as a combination of the

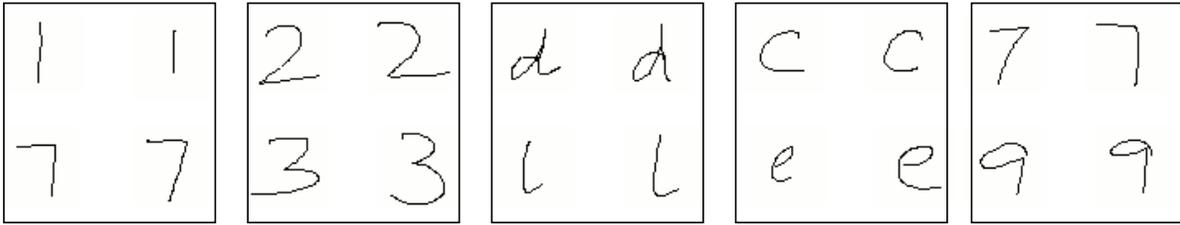


Figure 2. A few samples of similar numeral/character pairs used for experiments.

substroke-level matching costs and the weights ϕ_k discriminate between the substrokes. Naturally, $f(\cdot)$ depends on the modelling scheme used for the substrokes. For the three modelling schemes described above, the corresponding substroke matching score is defined as follows.

- When the substrokes are represented by linear prediction coefficients, we define $f(\cdot) = \sum_{k=1}^s \phi_k d(\theta_T^k, \theta_A^k)$, where $d(\cdot)$ is the Euclidean distance between the two coefficient vectors. This scheme does not model the sequencing information of substrokes explicitly, unlike the other two methods.
- In the case of DTW, we have the alignment cost for the substrokes as a matching criterion.
- Using HMMs for modelling the individual substrokes provides a probabilistic matching score – the posterior probability of a substroke belonging to a particular class – which is to be maximized. Thus, the cost measure is defined as $D(\mathbf{T}, \mathcal{A}) = -\sum_{k=1}^s \phi_k p(\theta_T^k | \mathcal{A})$, where $p(\theta_T^k | \mathcal{A})$ denotes the probability of the substroke θ_T^k belonging to class \mathcal{A} , and is computed from the training parameters $\{\Xi, \mathcal{A}, B, \pi\}$. Such modelling accounts for noise as well as variability in the class samples.

We illustrate the superiority of our approach, using the three modelling schemes discussed, in the next section.

5. Results and Discussion

The dataset consists of more than 1200 online numeral and character strokes collected from different people using an IBM CrossPad. A few samples of this data are shown in Figure 2. To demonstrate the applicability of our approach for discriminating two classes, we chose similar character/numeral pairs (e.g., (2, 3), (d, l), etc.). The database is divided into training set (to estimate the substrokes and their corresponding weights) and testing set (to evaluate the recognition performance). To account for the variability in the data due to translation, we normalize the features using a bounding box for the stroke and rescaling it to the 0 – 1 range.

After preprocessing the data, we identify the substrokes using a DTW alignment scheme. Then, we model the individual substrokes. Experiments are performed using three modelling schemes – DTW, HMM, LPC. The contribution of each substroke in the decision process is enhanced with a corresponding discriminatory weight to compute the final stroke-level matching score. We compare our results for each modelling scheme by assigning weights in two ways: (1) Equal weights for all the substrokes, (2) Weights computed using our approach. Equal weight assignment is equivalent to the existing recognition schemes. It is observed that our approach outperforms the equal weighting scheme. These results are summarized in Table 1.

Numerals: Results are shown on three pair-wise combinations of numerals – (2, 3), (1, 7), (1, 9). In general, HMM combined with our approach for weight assignment resulted in the best performance for all the pairs.

Characters: We present results on recognizing combinations of characters (u, w), (d, l) and (c, e). On average, the % accuracy improved by 3.2 using our weight assignment approach. Just as in the recognition of numerals, using HMM for modelling the substrokes and distinguishing them with discriminant weights gives the best average accuracy of nearly 96%.

On an average, we achieve a 41% reduction in classification error rate. We have observed that for a wide variety of parameters (like the states in HMM, order of the prediction for LPC, etc.), the percentage improvement is significant and consistent.

5.1. Discussion

In general, the accuracy of the classifiers reported here are much lower than the commercial or most of the reported recognition algorithms. This is due to the fact that (a) our datasets were not tuned to achieve higher recognition or pre-processed to suit a specific recognition scheme, and (b) our implementation of HMMs, DTW, etc. is not tuned for the online data case.

From the results in Table 1 it is evident that HMM is the best modelling scheme among the ones considered for online handwriting data. The underlying probabilistic framework in HMMs accounts for the noise and variations in

	LPC			DTW			HMM		
	Equal	Our Approach	% Red.	Equal	Our Approach	% Red.	Equal	Our Approach	% Red.
2, 3	89.0	92.0	27.3	90.0	96.0	60.0	94.4	98.0	64.3
1, 7	90.0	94.0	40.0	93.0	96.0	42.8	92.0	98.0	75.0
7, 9	89.0	93.0	36.4	96.0	98.0	50.0	100.0	100.0	0.0
<i>u, w</i>	88.0	93.4	45.0	93.4	94.0	9.1	90.0	96.0	60.0
<i>d, l</i>	91.0	93.0	22.2	92.6	98.0	73.0	94.0	96.0	33.3
<i>c, e</i>	93.8	95.0	19.4	94.0	96.0	33.3	92.0	96.0	50.0

Table 1. Results on the classification of 6 pair-wise combinations of numerals and characters. In each of the three modelling schemes (LPC, DTW, HMM), the average percentage accuracies achieved by using two different weighing schemes (equal weights, weights obtained by our approach) and the percentage of error reduction (% Red.) are shown. In almost all cases our weighing approach outperforms equal weight scheme.

the data, unlike DTW, which is a template matching technique. Linear Prediction Coefficients (LPC) are highly data-specific and are possibly unsuitable for modelling complex non-linear substrokes accurately. In such cases, it may be relevant to have higher order predictions (such as quadratic, cubic, etc.).

Although we presented the results using 2-dimensional points as the patterns obtained over time, other features available for online data such as pressure, velocity, etc. can be readily used in the framework by changing Θ^k appropriately.

A direct extension to the multiclass scenario can be achieved using Directed Acyclic Graphs (DAGs) [8]. A DAG comprises of many pairwise classifiers, which are connected to build a multiclass classifier. The test sample, which is presented at the root node, propagates through the graph until it reaches a leaf node (where it is labelled). To build a multiclass classifier for N classes, we need $N(N-1)/2$ pairwise classifiers. In the case of online handwritten numeral recognition, 45 pairwise classifiers (eg. (0, 1), (1, 2), (2, 3), etc.) built according to the scheme described in this paper are required. Other multiclass extensions on the lines of Multiple Discriminant Analysis may also be used [3].

6. Conclusion

The contribution of this paper is in presenting an approach to identify substrokes of online handwriting data which are more useful in the classification task. The usefulness of discriminating features is evident from the recognition accuracy achieved. Furthermore, the generic framework described in this paper allows replacement of individual components – substroke identification, substroke modelling, recognition criterion – in a data-specific way. We are currently working on building a multiclass recognition scheme for various characters.

References

- [1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Trans. on PAMI*, 19(7):711–720, 1997.
- [2] V. Deepu, S. Madhvanath, and A. G. Ramakrishnan. Principal Component Analysis for Online Handwritten Character Recognition. In *Proc. ICPR*, volume II, pages 327–330, 2004.
- [3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, New York, 2001.
- [4] S. Jaeger, S. Manke, J. Reichert, and A. Waibel. Online handwriting recognition: The NPen++ recognizer. *International Journal on Document Analysis and Recognition*, 3(3):169–180, 2001.
- [5] R. Martens and L. Claesen. On-Line Signature Verification by Dynamic Time-Warping. In *Proc. ICPR*, pages 38–42, 1996.
- [6] H. Nishimura and T. Timikawa. Off-line Character Recognition using On-line Character Writing Information. In *Proc. ICDAR*, volume 1, pages 168–172, 2003.
- [7] R. Plamondon and S. N. Srihari. On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Trans. on PAMI*, 22(1):63–84, 2000.
- [8] J. C. Platt, N. Cristianini, and J. Shawe-Taylor. Large Margin DAGs for Multiclass Classification. In *Advances in Neural Information Processing Systems 12*, pages 547–553. MIT Press, 2000.
- [9] H. Shimodaira, T. Sudo, M. Nakai, and S. Sagayama. On-line Overlaid-Handwriting Recognition Based on Substroke HMMs. In *Proc. ICDAR*, volume 2, pages 1043–1047, 2003.
- [10] A. Vinciarelli and M. Perrone. Combining Online and Off-line Handwriting Recognition. In *Proc. ICDAR*, volume 2, pages 844–848, 2003.
- [11] Q.-Z. Wu, I.-C. Jou, and S.-Y. Lee. On-Line Signature Verification using LPC Cepstrum and Neural Networks. *IEEE Trans. on Systems, Man and Cybernetics - Part B*, 27(1):148–153, 1997.