An Approach to Identify Unique Styles in Online Handwriting Recognition♦

A. Bharath, V. Deepu, Sriganesh Madhvanath
HP Laboratories India
HPL-2005-108
June 3, 2005*

clustering, stopping criteria, online handwriting, style identification

We describe a method for identifying different writing styles of online handwritten characters based on clustering. The motivation of this experiment is to develop automatic characterization of different writing styles that arise due to variation in stroke number or stroke ordering. An efficient agglomerative hierarchical clustering technique with the nearest neighbor approach was implemented to cluster strokes. The results obtained from our experiment indicate that the resulting prototypes are unique and essentially capture different writing styles.

* Internal Accession Date Only
♦ICDAR 2005 International Conference on Document Analysis and Recognition, 29 Aug – 1 Sep 2005, Seoul, Korea
Approved for External Publication

An Approach to Identify Unique Styles in Online Handwriting Recognition

Bharath A.
Hewlett-Packard Labs India
Bangalore 560 030
bharath.a@hp.com

Deepu V.
Hewlett-Packard Labs India
Bangalore 560 030
deepuv@hp.com

Sriganesh Madhvanath
Hewlett-Packard Labs India
Bangalore 560 030
srig@hp.com

Abstract

We describe a method for identifying different writing styles of online handwritten characters based on clustering. The motivation of this experiment is to develop automatic characterization of different writing styles that arise due to variation in stroke number or stroke ordering. An efficient agglomerative hierarchical clustering technique with the nearest neighbor approach was implemented to cluster strokes. The results obtained from our experiment indicate that the resulting prototypes are unique and essentially capture different writing styles.

1. Introduction

Online handwriting recognition refers to the problem of interpretation of handwriting input captured as a stream of pen positions using a digitizer or other pen position sensor [2, 4]. Many recognition algorithms are based on the use of prototypes that represent different writing styles for the same character. The recognition accuracy deteriorates if the prototypes do not represent the actual writing styles. The identification of different writing styles of the same character is useful for the training of such algorithms, besides being generally for the design of algorithms and features for handwriting recognition.

A common approach for detection of writing styles is to cluster entire entire character samples for each class [8]. But this approach of clustering loses commonality in strokes between different writing styles (i.e, same stroke occurring across different writing styles). In this effort, we focus on the stroke level to capture different styles of writing the same character. In particular, we use a hierarchical clustering method to cluster strokes of a character. Based on the cluster models, the system automatically builds models for each character.

The paper is organized as follows: Section 2 describes the preprocessing technique used. The clustering technique employed is described in Section 3. Section 4 explains the automatic modeling of different characters. Section 5 presents some experimental results. Some conclusions and future directions are presented in Section 6.

2. Pre-processing

Pre-processing is required in order to compensate for variations in time and scale, and can be broken down into the steps of smoothing normalization and resampling [7].

Smoothing is performed to reduce the amount of high-frequency noise in the input resulting from the digitizer or tremors in writing. A low-pass filter is employed for smoothing. In our scheme, each stroke is smoothed independently and care is taken to preserve the end points.

To eliminate variability due to translation and compensate for size differences, size normalization is carried out. The characters are centered and rescaled. In the process of rescaling, the bounding box of the character is computed. The character size is normalized to unit square. The aspect ratio is retained if the aspect ratio of original character is above a threshold.

Re-sampling is performed to obtain a constant number of points that are uniformly sampled in space, whereas the input data is the result of uniform sampling in time. In this process,the original points are replaced with a new set with constant spacing using piece-wise linear interpolation. The result of pre-processing is a new sequence of points $[x_i, y_i]$ of constant length, of constant scale and regularly spaced in arc length.

3. Clustering

For automatic characterization of writing styles, it is necessary to have an unsupervised classification technique. Clustering addresses this problem. Clustering at the character level is quite common [8] but it does not capture the commonality in strokes between writing styles. For instance, consider the character in Figure 1(a). If this character were written in two styles with different stroke order

as shown in Figure 1(b) and 1(c), the samples would be divided into two different clusters as the feature vectors would be different. The training data would contain separate clusters for each of these. But the second stroke of style 1 and first stroke of style 2 are the same, and similarly, the first stroke of style 1 and second stroke of style 2 are identical. This information is lost in character level clustering. To avoid this problem, we employ clustering at the stroke level.

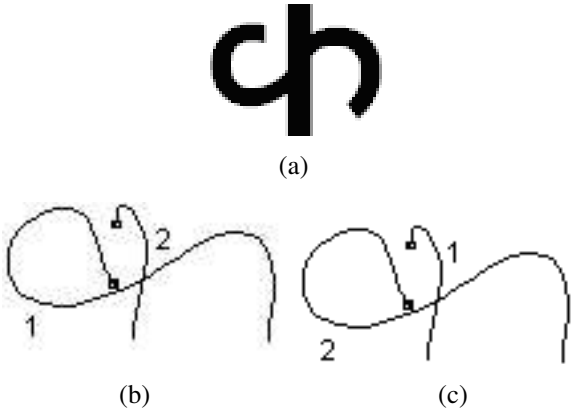


Figure 1. Two different writing orders for the same character *ka* from the Devanagari script

There are different clustering techniques proposed in the literature. In this effort, we experimented with the agglomerative hierarchical clustering technique [3] based on the nearest neighbor approach. The agglomerative hierarchical clustering starts by considering each object as a cluster and progresses by merging nearest neighbors. At each step, the two clusters which minimize the inter-cluster distances are merged. The inter-cluster distance is defined as the distance between the nearest neighbors of the two clusters as shown in Figure 2. At each level, the number of clusters decreases by one .

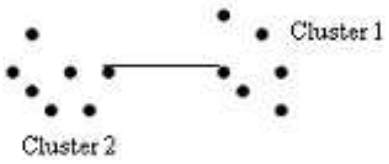


Figure 2. Inter cluster distance

3.1. Efficient Clustering

The classical hierarchical clustering technique has a high complexity of $O(n^3)$ for $c << n$ where c is the number of clusters required and n is the number of objects that need to be clustered. In our experiment we designed a more efficient method which reduced the complexity to $O(n^2 + nlog(n))$ (which is essentially $O(n^2)$). This resulted in an appreciable increase in the performance and reduction in clustering time.

The algorithm is as follows

BEGIN
Initialize n, c and $t = n$
Compute inter-object distances $[O(n^2)]$
Put them in a map (key = distance, value = pair)
Sort this map based on inter-object distances $[O(nlog(n))]$
Traverse the map as $[O(n^2)]$
do
    if(two data objects are in different clusters)
        Merge clusters
        Decrement t by 1
    else
    Continue
while($t! = c$)
END

3.2. Stopping Criteria

One of the most common problems encountered with clustering is deciding upon the number of clusters, i.e. the stopping criterion. We experimented with various methods like Bayesian Information Criterion(BIC) [1], Average Silhouette [5], Point of Maximum Second Derivative, L-method [6] etc.

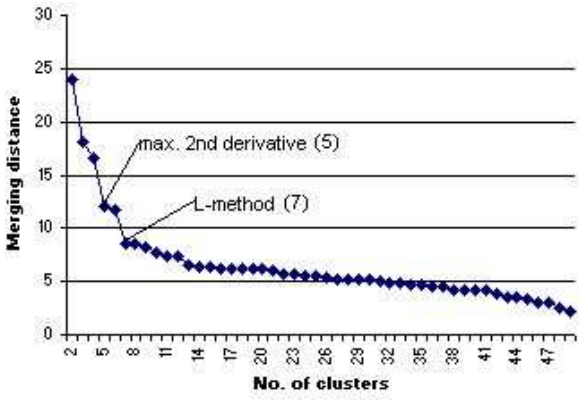


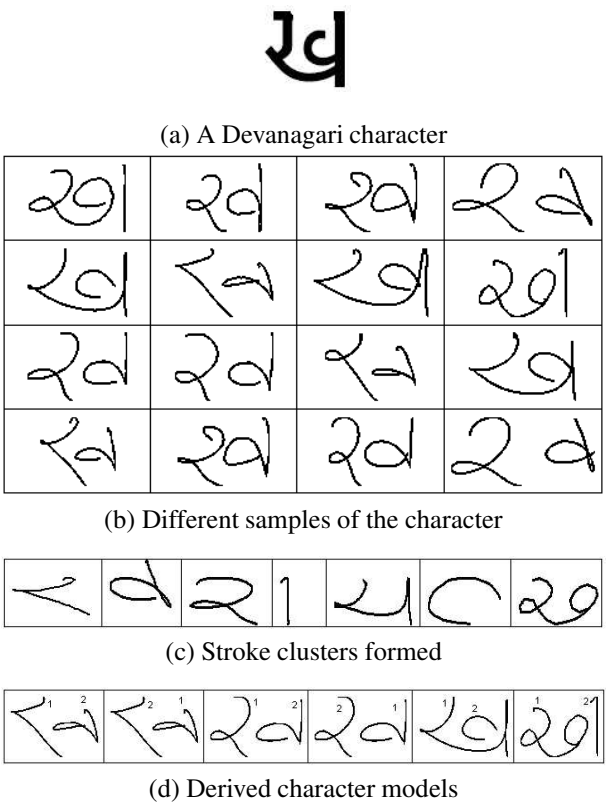
Figure 3. Number of clusters versus merging distance

In the BIC approach, the BIC value is evaluated for each cluster. In agglomerative clustering, two clusters are

merged only if the merging increases the BIC value. This approach assumes a multivariate Gaussian distribution for each cluster. Since we did not have enough samples in each cluster, we could not use BIC as the stopping criterion. The Average Silhouette is a function of intra-cluster and inter-cluster similarities. Being a global measure, it could not capture the within stroke variations between similar looking strokes.

The other two methods (Point of Maximum Second Derivative and L-method) are based on the plot of the number of clusters versus merging distance curve (see Figure 3). The first method determines the number of clusters by identifying the point that has the largest second derivative in the curve. The L-method finds the boundary between the pair of straight lines that most closely fit the curve.

The clustering is performed at the stroke level using Euclidean distance measure with L-method as stopping criterion. Singleton clusters are considered as outliers (they are either very specific styles or noisy data). The character samples containing outlier strokes are rejected. The clusters formed for the character in Figure 4(a) are shown in Figure 4(c).



(a) A Devanagari character

(b) Different samples of the character

(c) Stroke clusters formed

(d) Derived character models

Figure 4. Stroke level clustering and model generation

4. Character Modeling

The previous section described how the stroke clustering was carried out. In this section, we discuss the automatic characterization of different writing styles for a character class.

Once the stroke clusters have been formed, each valid sample (that does not contain an outlier stroke) of a character is taken. These samples contain the sequence of strokes that originally formed them. The strokes are now present in different clusters and hence each stroke is assigned its corresponding Cluster ID. This assignment of stroke to Cluster ID is done for all the samples. Now, each sample can be represented by a sequence of Cluster IDs. To determine the styles in writing a character, we determine the set of unique sequences of Cluster IDs. This set is essentially the character model. In other words, this set represents the ways in which the character could be written. Figure 4(d) shows the model for the character shown in Figure 4(a).

5. Experimental Evaluation

The experimental evaluation of the above techniques was carried out using word samples of the Devanagari script. The data was collected from six writers (60 words, 5 samples/word) and annotated at the character level using a set of 99 character labels that correspond to basic constituents of the script (vowels, consonants, modifiers and half consonants). The strokes from different samples of a character class were clustered using the efficient agglomerative hierarchical clustering described in Section 3. The L-method was able to determine the number of clusters relatively accurately whereas the Point of Maximum Second Derivative was sensitive to outlier strokes. Results using these methods are compared with the number of natural clusters(manually determined) in Table 1.

Singleton clusters generally represent unusual/outlier stroke samples and hence the character samples containing them were rejected. Once the clusters were formed, character models for each character were derived as explained in Section 4. Figure 5 shows character models derived for some of the characters.

6. Conclusions

In this paper, we investigated efficient agglomerative hierarchical clustering of handwriting samples at the stroke level for the identification of unique writing styles. The number of clusters determined using the L-method was comparable with the number of natural clusters. Once the strokes were clustered, the character models were automatically derived by extracting the unique Cluster ID sequences.

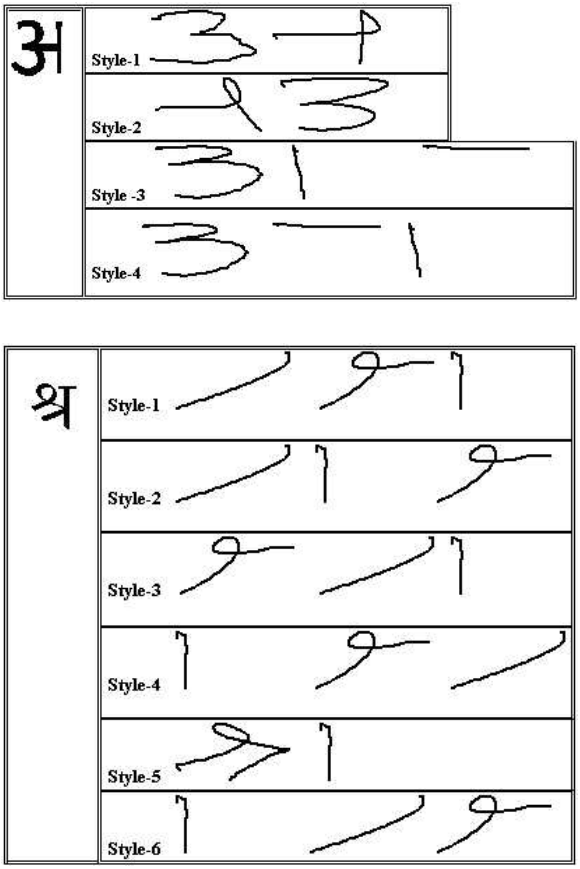


Figure 5. Character models for two classes from Devanagari

Table 1. Number of clusters determined by various methods

Symbol	Number of clusters			
	Manual inspection	Largest second derivative	Average Silhouette Method	L-method
ख	7	5	5	7
श्र	7	3	5	7
उ	3	3	4	3
झ	10	4	8	10
ट	3	1	1	4
॥	7	4	6	6
९	8	3	4	7

Although our experiments were carried out on character samples from the Devanagari script, it does not use any script-dependent features. Hence, the technique may be readily extended to other scripts.

Currently the clustering technique employs the Euclidean distance measure for determining inter cluster distance. Other distance measures like DTW (Dynamic Time Warping) distance can be experimented with.

We are currently developing a word recognition engine for the Devanagari script, which uses the character models derived using the techniques derived. The accuracy of the recognition engine trained on these character models may be used to benchmark the performance of our technique and the different stopping criteria in quantitative terms.

References

[1] S. S. Chen and P. S. Gopalakrishnan. Clustering via the bayesian information criterion with applications in speech recognition. *Proceedings of ICASSP*, 2:645–648, 1998.

[2] 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.

[3] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons Inc, New York, 2000.

[4] S. Jaeger, C. L. Liu, and M. Nakagawa. The state of art in japanese online handwriting recognition compared to techniques in western handwriting recognition. *International Journal on Document Analysis and Recognition*, 6:75–88, July 2003.

[5] K. S. Pollard and M. J. van der Laan. A method to identify significant clusters in gene expression data. *U.C. Berkeley Division of Biostatistics Working Paper Series (http://www.bepress.com/ucbbiostat/paper107)*, April 2002.

[6] S. Salvador and P. Chan. Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. *Proceedings of 16th IEEE International Conference on Tools with Artificial Intelligence*, 3:1852–1857, 2004.

[7] D. Vijayasenan and S. Madhvanath. Principal component analysis for online handwritten character recognition. *Proceedings of the 17th International Conference on Pattern Recognition*, 2:327–330, August 2004.

[8] V. Vuori and J. Laaksonen. A comparison of techniques for automatic clustering of handwritten characters. *Proceedings of the 16th International Conference on Pattern Recognition*, 3:168–171, 2002.