2009 10th International Conference on Document Analysis and Recognition

A Framework based on Semi-Supervised Clustering for Discovering Unique Writing Styles

Bharath A. and Sriganesh Madhvanath
Hewlett-Packard Laboratories, Bangalore, India
{bharath.a, srig}@hp.com

Abstract

An online multi-stroke character is often written in many ways. While some vary in the number of strokes they contain, others differ in the ordering of strokes. It is important for a writer-independent recognition system to learn these different styles of writing the character during the training phase in order to better model the training data. Typically, the samples of a character are clustered in an unsupervised manner and each cluster is modeled individually. In this paper, we describe an approach based on 'semi-supervised clustering' where basic domain knowledge can be incorporated for better clustering of strokes present across all the characters. Experimental results show improved recognition accuracy when compared to the baseline system.

1. Introduction

Online handwriting is captured as a time-ordered sequence of strokes where a stroke corresponds to the ink collected in between a pen-down event and the succeeding pen-up event. This temporal information proves to be a burden at times, since it captures variations in stroke number and order which do not change the identity of what is written. A character sample is said to follow a particular writing *style* when written according to a specific ordering of strokes where each stroke is of a particular shape. Being able to automatically discover unique writing styles from samples of handwriting has many applications in handwriting recognition, e.g. (i) deriving compact prototype sets for Nearest Neighbor classification schemes and (ii) building individual models representing different styles for sequential algorithms such as Hidden Markov Models (HMM). While most of the previous efforts employ clustering in an 'unsupervised' setting, in this work we explore the possibility of 'semi-supervised clustering'[1] - exploiting domain-specific constraints to derive clusters of stroke samples that are common across all the character classes. We have also attempted to develop a generic framework for identifying different writing styles when provided with a dataset of character samples.

2. Literature review

Prior work on identifying writing styles in online handwriting can broadly be categorized into two main strategies:

1. *Character-level clustering* - In this strategy the samples of a character class are clustered using a conventional clustering algorithm, following appropriate preprocessing and feature extraction. Each cluster would then represent a specific style of writing the character. Vuori and Lakksonen [1] evaluate the performance of four agglomerative hierarchical clustering algorithms on isolated digits, uppercase and lowercase characters. They note that cluster validity indices such as Davies-Bouldin and Calinski-Harabasz did not perform as one would expect and hence the number of clusters had to be specified manually. Connell and Jain [2] employ Mean Squared Error criterion for clustering isolated digits. The results show that there is only a marginal decrease in the Nearest Neighbor classification accuracy when the training set was reduced to 8.5% of its original size by clustering. This also results in 90% reduction in recognition time which may be regarded as significant for a real-time application. Character-level clustering has also been applied for HMM-based recognition of online Hangul characters [3]. The handwriting samples are first represented using direction codes. These samples are then clustered using an agglomerative technique and an HMM is built for each cluster.

2. *Stroke-level clustering* - This technique is typically applied for scripts that are overwhelmed with stroke order and number variations as in the case of Chinese, Japanese and Korean (CJK) scripts. In addition to identifying different writing styles, clustering at the stroke-level is also intended to discover the strokes that are common across different styles and/or character classes. Yamasaki [4] proposes a two-stage stroke clustering scheme for Japanese

[1] Unlike 'semi-supervised classification' which addresses the problem of using large amount of unlabeled data, together with the labeled data, to improve *classification*, in 'semi-supervised clustering', one has unlabeled data with some domain-specific pairwise constraints and the goal is *clustering*.

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.148

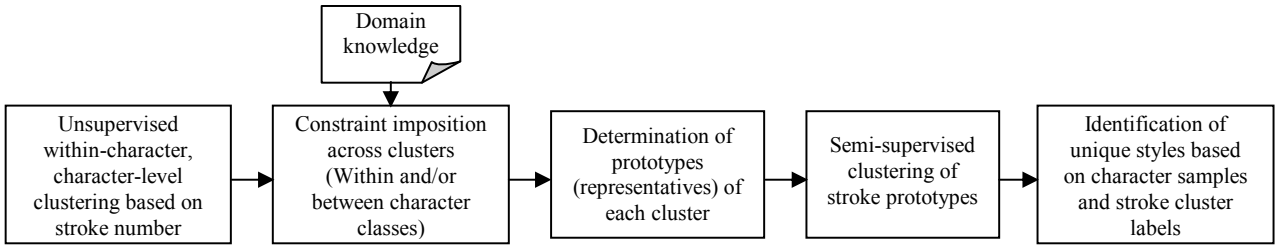

Figure 2. Framework for incorporating domain knowledge into stroke-level clustering

character recognition. Firstly, the samples of a character class are categorized according to the number of strokes they contain. The strokes having the same time index are considered as initial clusters. A top-down cluster splitting approach is adopted to divide clusters that have different stroke shapes. The decision to partition a cluster ensures that the farthest sample from the mean stroke is at a distance less than the threshold. The same criterion is also applied in the bottom-up cluster merging step where clusters of strokes with different time indices are grouped together. Finally, the stroke representatives of the clusters within a character class are clustered across other characters. The resulting stroke prototypes are used to determine character allographs that retain the original sequence and position information.

3. Using prior knowledge to improve stroke clustering

Most of the approaches presented in the previous section employ conventional clustering algorithms in an unsupervised setting. On the other hand, we would like to explore whether prior knowledge could be incorporated to obtain better clusters, especially in the case of style identification by clustering strokes. Stroke-level clustering has two main advantages over clustering entire character samples. Since the stroke clusters are common across all the character classes, significant data sharing is possible which would prove valuable for approaches such as HMM. Secondly, since one can know the strokes that are common across different characters, real-time text-entry applications such as QuickStroke [5] become realizable.

In a stroke clustering based approach, resulting styles are represented as sequences of stroke cluster labels. In our earlier work on style identification [6], when stroke clustering was attempted across all the Devanagari characters, we found that some of the resulting styles were invalid. Disregarding the ordering and considering each style merely as a collection of stroke cluster labels, we found that there are pairs of styles of the same character class where one style is a sub-collection of the other. For instance, in Fig. 1, capital letter 'B' is written in two different styles. Sample 1 contains two strokes whereas Sample 2 has three. When unsupervised stroke clustering technique is applied, it might correctly assign the first strokes of the two samples to the same cluster (with ID *a*) and wrongly cluster stroke 2 of Sample 1 and strokes 2 and 3 of Sample 2 also into one (with ID *b*). As a result, the style of Sample 1 i.e. {*a*, *b*} becomes a sub-collection of the style of Sample 2 i.e. {*a*, *b*, *b*}. Given that both the samples belong to the same character class 'B', therefore it indicates that one of them is not a true style of the character class. In the next section we describe how this handwriting domain specific knowledge can be incorporated for better clustering of strokes.

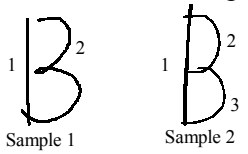

Figure 1. Two different styles of writing 'B'

4. Proposed framework

In this section, we propose a multi-stage framework (Fig. 2) based on stroke-level clustering for discovering unique writing styles of multi-stroke characters. The framework provides a mechanism to incorporate prior information in the form of constraints between stroke clusters.

4.1 Unsupervised within-character, character-level clustering

In the first stage of clustering, samples of each character class are categorized based on the number of strokes they contain and character-level clustering is carried out within the character class for each stroke number. These samples for clustering would differ either in their ordering of strokes or the shapes they contain. Since either of these variations would result in significant dissimilarity in feature values, one could expect the hidden clusters to be well separated. Any conventional clustering algorithm may be applied for this purpose. The strokes having the same time index in a character cluster now form initial stroke clusters.

4.2 Constraint imposition using domain knowledge

In order to detect strokes common across character classes, the stroke clusters obtained in the previous

stage have to be merged with other stroke clusters within the same or different character classes. Instead of clustering these stroke clusters in an unsupervised manner, one could introduce prior information to aid clustering. For instance, in order to avoid obtaining the invalid styles mentioned in Section 3, constraints could be imposed as follows. If we could determine that the first strokes of both the samples in Fig. 1 are similar, then we can conclude that the second stroke of Sample 1 cannot possibly fall into the clusters of stroke 2 and 3 of Sample 2 (Fig. 3). Such a constraint helps preempt the possibility of a style being a sub-collection of another of the same character class.

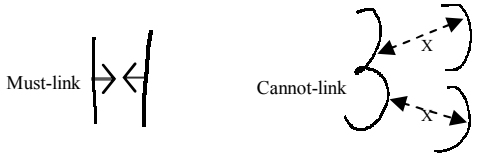

Figure 3. Imposing constraints on stroke clusters based on domain knowledge

In general, the constraint can be stated as follows. For a given character class, if there is an m-stroke sample S_m and an n-stroke sample S_n where $m < n$, and if one-to-one correspondence has been established between any m-1 strokes of S_m and m-1 strokes of S_n, then the remaining stroke of S_m cannot fall into any of the clusters of the remaining n-m-1 strokes of S_n. The same constraint can also be extended across different character classes. No two single-stroke samples of different character classes can be present in the same cluster. However, this constraint may not be applicable between characters which vary only in their positions but not in their shapes (e.g. hyphen '-' and underscore '_') and when features describing only their shapes are used for clustering.

4.3 Semi-supervised clustering

Semi-supervised clustering or 'clustering with constraints' [7-9] has been an active area of research over the last decade in the machine learning community. The constraints are often pair-wise relationships that indicate whether two data points should be in the same cluster (*must-link constraint*) or in different clusters (*cannot-link constraint*). The constraints aid and bias the clustering algorithm in order to obtain desired results. Based on the constraints, the algorithm could learn a new distance measure between the samples and/or ensure that the constraints are maximally respected. As one might expect, empirical results reported in the literature indicate that the quality of clusters obtained from semi-supervised clustering is often better than the unsupervised one [7].

In our scenario, the constraints on strokes described earlier may be modeled as must-link and cannot-link constraints, some examples of which are shown in Figure 3. The stroke prototypes from each cluster, along with these constraints, serve as inputs to a semi-supervised clustering algorithm, as described in detail in Section 5.5.

4.4 Identifying unique writing styles

Once each stroke is assigned to a cluster, identifying unique styles of writing a character is straightforward [6]. Each character sample is represented as a sequence of cluster labels to which the constituent strokes are assigned, and the unique sequences across all the samples of a character class then represent unique styles of writing the character.

5. Application to Devanagari character recognition

Devanagari, like any other Indic script, belongs to the family of syllabic alphabets. It is the most widely used script in India and contains 35 consonants, 11 vowels and 4 vowel modifiers. These basic characters combine to form more complex characters (or syllabic units). A majority of the characters are typically written in multiple strokes and hence stroke order and stroke number variations are considered to be significant in the script [6].

The utility of the proposed semi-supervised clustering framework was validated indirectly by measuring its impact on the recognition performance of 47 Devanagari characters including the basic consonants and vowels, as described below.

5.1 Preprocessing

In the entire experiment, preprocessing of character samples was carried out only once at the stroke-level. Irrespective of the position of the stroke in the character sample, each stroke is first translated to the origin and then the larger dimension (height or width) is rescaled to *10* while retaining the aspect ratio of the stroke. As a result, each stroke of the character sample appears to be written at the same location and of similar size. Stroke-level preprocessing was necessary for eyes-free handwriting recognition [10]; however, one may alternatively retain the positions of strokes (especially the vertical position 'y') for the purpose of recognizing normal handwriting. In order to remove writing speed variations and enable a vector representation, each stroke is also resampled to a fixed number (*30*) of points.

5.2 Feature extraction

The proposed approach was evaluated using two different sets of features. The first set is comprised of

the x-y values at each point after stroke-level preprocessing. The second set comprises angle-based features, writing direction, curvature and slope [11] in addition to the ones present in the first set. The values of the angle-based features are normalized such that the dynamic range of each feature is the same (*0* to *10*).

5.3 Unsupervised within-character, character-level clustering

Once the features are extracted, the first stage of clustering is carried out using the Agglomerative Hierarchical Clustering algorithm [6] with complete-linkage as the inter-cluster distance measure, and squared Euclidean distance as the inter-stroke distance measure. With complete-linkage as the inter-cluster distance measure one could expect that the algorithm would result in compact clusters. In order to determine the optimum number of clusters, two different stopping criteria were evaluated: L-method [12] and Longest Lifetime [13]. L-method formed fine clusters wherein even the relatively minor shape variations of the strokes present in the character samples were captured. As a result, even when samples were visually of the same style, they were placed in different clusters. On the other hand, the Longest Lifetime stopping criterion found coarse clusters that generally varied only in their writing order, and hence was preferred in our experiment. The character-level clusters are transformed into stroke-level ones by simply grouping the strokes that have the same time index within each character cluster.

5.4 Imposing constraints based on inter-cluster similarity

The stroke clusters formed in the previous stage are examined for constraint imposition. The constraints were applied according to the criterion explained in Section 4.2. In order to determine that the two stroke clusters are similar, we compute the similarity between their means (prototypes) μ_1 and μ_2 as described by Manor and Perona [14].

$$similarity(cluster_1, cluster_2) = similarity(\mu_1, \mu_2) = exp\left(\frac{-d^2(\mu_1, \mu_2)}{\sigma_1 \sigma_2}\right)$$

where $\sigma_i = d(\mu_i, s_K)$ is the Euclidean distance between the mean and the Kth nearest neighbor (S_K) in the stroke cluster. The value of K was set to *7* in our experiment. If the similarity between the cluster means exceeds a predefined threshold (*0.4*), the clusters are said to be similar (must-link constraint). Cannot-link constraints are imposed on the unmatched strokes when the criterion described in Section 4.2 is satisfied.

5.5 Semi-supervised clustering of stroke prototypes

The semi-supervised or constrained clustering algorithms presented in the literature invariably mandate the user specify the final number of clusters in advance. We implemented the Constrained Complete Link (CCL) algorithm proposed by Klein et al. [15] and determined the number of clusters based on the Longest Lifetime criterion [13]. The similarity measure described in the previous section was converted to a distance by subtracting it from 1. The algorithm imposes the must-link constraints by setting the corresponding distances to 0 and propagates them through the ALL-PAIRS-SHORTEST-PATHS (Floyd-Warshall) procedure which results in modified distance measure between the stroke prototypes. Cannot-link constraints are imposed by setting the corresponding pair-wise distances to 1 (maximum value) and they are propagated implicitly by the complete-linkage criterion. Figure 4 shows the cluster merging curve when constraints are imposed and propagated. At the end of semi-supervised clustering, the stroke clusters whose mean prototypes fall into the same cluster are merged to form the final clusters of strokes. The unique styles of writing are then identified as described in Section 4.4.

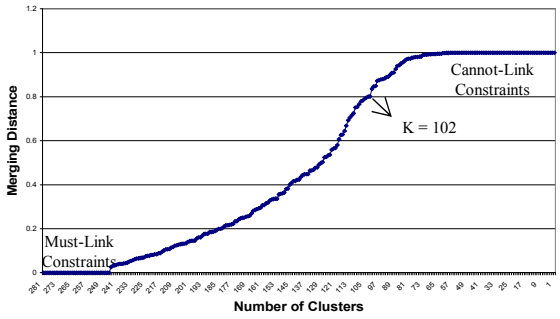

Figure 4. Semi-supervised clustering of stroke prototypes

6. Experimental evaluation

We evaluated our style identification framework indirectly using the character recognition accuracy of HMMs constructed for each character class. First, a left-to-right stroke HMM is built for each final stroke cluster. Second, based on each identified style of writing a particular character, the stroke HMMs are concatenated to form the character HMM for that character class and style. As a result, a character HMM has multiple parallel paths corresponding to each style of writing. The training set for style identification contained isolated character samples of 106 writers where each writer had contributed 2 samples per character class. An independent test set for evaluating the character recognition accuracy contained 50

samples per character class provided by a different set of writers. The preprocessing and feature extraction stages are the same as described before, except that the strokes are now resampled by fixing the distance between consecutive points. This results in a variable number of points for each stroke and is suitable for training HMMs. A single HMM modeling all the writing styles of a character class was used to obtain a baseline system.

Table 1. Comparison of character recognition accuracies

Approach	Features	Accuracy %
Single HMM per character class	x, y	70.0
	x, y, writing direction, curvature, slope	84.2
Character-level, unsupervised clustering	x, y	74.7
	x, y, writing direction, curvature, slope	86.7
Stroke-level, semi-supervised clustering	x, y	**77.0**
	x, y, writing direction, curvature, slope	**88.7**

Table 1. shows the accuracies obtained by three approaches, including unsupervised clustering at the character level. The results in the table show that the accuracy of our proposed approach for both sets of features is greater when compared to using a single HMM per character, or performing style identification by unsupervised character-level clustering.

7. Conclusions and future work

In this paper, we proposed a novel approach and framework for identifying unique writing styles based on semi-supervised clustering. The framework allows domain knowledge about the problem to be incorporated in order to aid and bias the clustering of strokes. The results indicate that the performance of the proposed approach is substantially better than those of the baseline single HMM per character class and unsupervised character-level clustering techniques. An important next step is to assess the informativeness of the constraints either by comparing recognition accuracy against that obtained by unsupervised stroke-based clustering, or in terms of prototype reduction for Nearest Neighbor classification. We would also like to explore whether the constraints can be exploited for determining the optimum number of clusters, a research problem which appears to be unaddressed in the semi-supervised clustering literature.

8. References

[1] V. Vuori and J. Laaksonen, "A Comparison of Techniques for Automatic Clustering of Handwritten Characters", *16th International Conference on Pattern Recognition*, Quebec, Canada, 11-15 Aug 2002, vol.3, pp. 168- 171.

[2] S. D. Connell and A. K. Jain, "Learning Prototypes for On-Line Handwritten Digits", *14th International Conference on Pattern Recognition,* Brisbane, Australia, 16-20 Aug 1998, vol. 1, pp. 182-184.

[3] J. J. Lee, J. Kim, and J. H. Kim, "Data-driven Design of HMM Topology for On-line Handwriting Recognition", *7th International Workshop on Frontiers in Handwriting Recognition*, Amsterdam, Netherlands, 11-13 Sep, 2000, pp. 107-121.

[4] K. Yamasaki, "Automatic Prototype Stroke Generation Based on Stroke Clustering for On-Line Handwritten Japanese Character Recognition", *5th International Conference on Document Analysis and Recognition,* Bangalore, India, 20–22 Sep, 1999, pp. 673-676.

[5] N. Matic, J. Platt, and T. Wang, "QuickStroke: An Incremental On-Line Chinese Handwriting Recognition System," *16th International Conference on Pattern Recognition*, Quebec, Canada, 11-15 Aug 2002, vol. 3, pp. 435-437.

[6] A. Bharath, V. Deepu, and M. Sriganesh, "An Approach to Identify Unique Styles in Online Handwriting Recognition", *8th International Conference on Document Analysis and Recognition*, Seoul, Korea, Aug 29 – Sept 1, 2005, vol. 2, pp. 775-778.

[7] S. Basu, "Semi-Supervised Clustering: Probabilistic Models, Algorithms and Experiments", *PhD thesis*, The University of Texas at Austin, 2005.

[8] S. Basu, M. Bilenko, A. Banerjee, and R. Mooney, "Probabilistic Semi-supervised Clustering with Constraints", *Semi-Supervised Learning*, Olivier Chapelle, Bernhard Scholkopf, & Alexander Zien (editors), Cambridge, MA. MIT Press, 2006.

[9] L. Yi, J. Rong, and A. K. Jain (2007), "BoostCluster: Boosting Clustering by Pairwise Constraints", *13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, USA, Aug 12-15, 2007, pp. 450-459.

[10] A. Bharath and M. Sriganesh, "Recognition of Eyes-free Handwriting Input for Pen and Touch Interfaces", *11th International Conference on Frontiers in Handwriting Recognition*, Montreal, Canada, Aug 19-21, 2008.

[11] S. Jaeger, S. Manke, J. Reichert, and A. Waibel, "Online Handwriting Recognition: the NPen++ Recognizer", *International Journal on Document Analysis and Recognition*, 2001, vol. 3, pp. 169-180.

[12] S. Salvador and P. Chan, "Determining the Number of Clusters/Segments in Hierarchical Clustering/Segmentation Algorithms", *16th IEEE International Conference on Tools with Artificial Intelligence*, Boca Raton, USA, Nov 15-17, 2004, vol. 3, pp. 1852–1857.

[13] A. L .N. Fred and A. K. Jain, "Combining Multiple Clusterings Using Evidence Accumulation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, vol. 27(6), pp. 835-850.

[14] L. Zelnik-Manor and P. Perona, "Self-Tuning Spectral Clustering", *Advances in Neural Information Processing Systems*, 2004, MIT Press, pp. 1601-1608.

[15] D. Klein, S. D. Kamvar, and C. D. Manning, "From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering", *19th International Conference on Machine Learning*, Sydney, Australia, Jul 8-12, 2002, pp. 307-314.