Prototype Learning Methods for Online Handwriting Recognition

Raghavendra B.S., Narayanan C.K., Sita G., Ramakrishnan A.G., Sriganesh M[2]
Dept. of Electrical Engineering, Indian Institute of Science, Bangalore, India
[2]Hewlett Packard Laboratories, Bangalore, India
r.bobbi@ece.iisc.ernet.in, {sita, ramkiag}@ee.iisc.ernet.in, srig@hp.com

Abstract

In this paper, we study different methods for prototype selection for recognizing handwritten characters of Tamil script. In the first method, cumulative pairwise- distances of the training samples of a given class are used to select prototypes. In the second method, cumulative distance to allographs of different orientation is used as a criterion to decide if the sample is representative of the group. The latter method is presumed to offset the possible orientation effect. This method still uses fixed number of prototypes for each of the classes. Finally, a prototype set growing algorithm is proposed, with a view to better model the differences in complexity of different character classes. The proposed algorithms are tested and compared for both writer independent and writer adaptation scenarios.

1. Introduction

Prototype learning plays an important role in the development of handwriting recognition algorithms and applications, especially for handheld devices constrained with limited resources. The quality of the prototypes influences the accuracy of recognition, while the number influences the computation time. This paper studies different algorithms for prototype selection for an online handwriting recognition application in the context of both writer independent and adaptation scenarios.

The proposed algorithms focus on reducing the size of the training set by selecting an appropriate subset as prototypes. In the first technique, the cumulative pair-wise distance is used as a measure to generate the prototype set for each character class. In the second method, similarity to an arbitrary symbol set of different orientations is used as the measure. Both methods generate fixed number of prototypes for each of the classes. The third algorithm aims at reducing the prototype set further by having variable numbers of prototypes for different classes, with complex characters having more prototypes. These algorithms are explained in detail in the following sections.

2. Database

Two different devices – HP iPAQ PocketPC (ARM processor running WinCE OS), and HP TabletPC TC1000 (Transmeta processor running WinXP TabletPC edition) were used for data collection. The sampling rates for these devices are 90 Hz and 120 Hz respectively. A database of 40 users, each writing all the 156 characters (of the Tamil script) ten times each, was collected. Twenty writers contributed data using the PocketPC, whereas the other twenty used the TabletPC. Each character was written in a separate box, which obviates the need for character segmentation. Figure 1 shows one sample each of the set of 156 distinct symbols in Tamil.

3. Methodology

The raw character, as obtained from the acquisition device, has variable number of points, depending upon the resolution of the device and the writing style of the user. The data was resampled to obtain constant number of points in space rather than in time. Uniform resampling resulted in 60 points for all character samples. To reduce the effect of noise, the x and y coordinate sequences were separately smoothed using a 5-tap Gaussian filter. The entire database of 40 users was divided into two categories. The training set consists of the writing samples of 30 users selected at random from the 40, and the test set, of the remaining 10 users. The training set as well as the test set contains data from different users and acquisition devices, with different sampling rates.

4. Prototype selection

4.1 Cumulative pair-wise distance

Connell [1] calculated the intra class distances for the samples of a particular character and clustered them based on this distance. We have modified this approach by selecting the prototypes without clustering. The pair-wise distances of a sample with respect to all the other samples are added to obtain the cumulative distance. The 50 samples corresponding to the 50 smallest cumulative

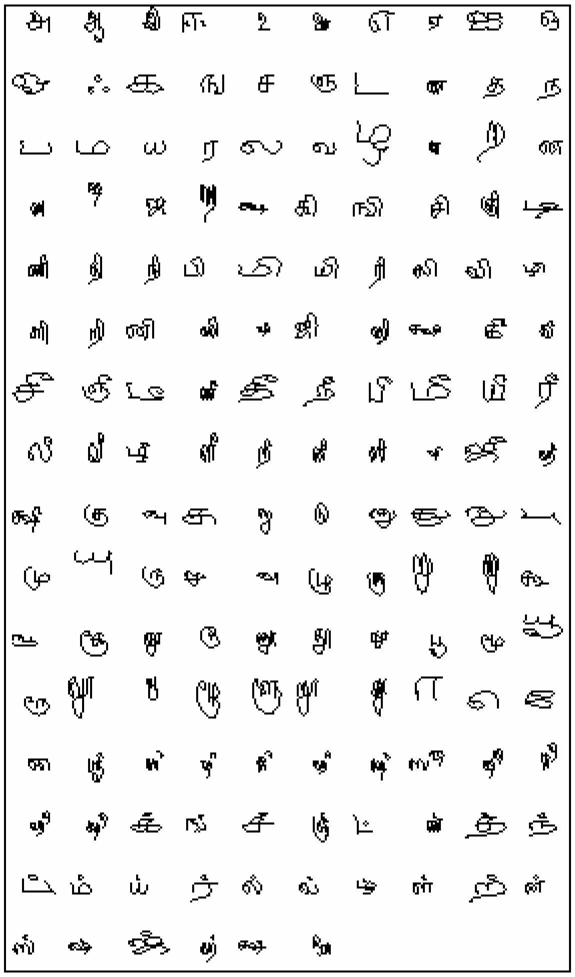

Figure 1. Sample of 156 distinct symbols in Tamil

distances are considered as prototypes. We refer to this approach as the cumulative pair-wise distance based method.

4.2. Orientation allographs

In this method, we define a set of arbitrary allographs, which is not a subset of the character set under consideration. The allograph set consists of four straight lines with angles 0^o, 45^o, 90^o and 135^o with respect to the horizontal, as shown in Figure 2.

Each of these lines is normalized to a length of 60 points. Each training sample is converted to a single stroke character by concatenating all the strokes. The process of selecting the prototypes is as follows: The cumulative DTW distance of each sample in the training set to the allograph set is considered as a measure for selecting the sample as a prototype. DTW algorithm matches two characters so that the sum of the squared Euclidean distances between the matched points is

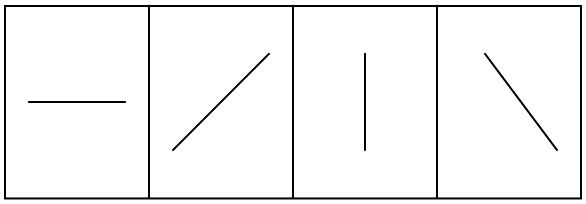

Figure 2. Oriented allographs

minimized. Thus for each sample, we get a distance that is the sum of the DTW distance of the sample with respect to each of the four lines. A motivating factor to consider these allographs is to account for various orientations of the character. If the character has a slant, then the distance of the character with respect to a vertical line may be affected. However the distance with respect to a line having a similar slant might be significantly less. Thus we consider the distances with respect to all the allographs before selecting the prototypes. Figure 3 shows the plot of this cumulative distance for 300 samples of a character.

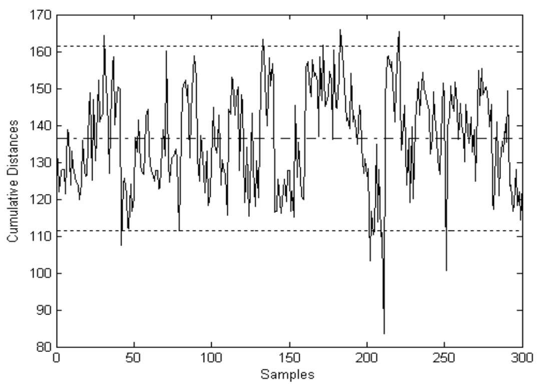

Figure 3. Cumulative distances of 300 character samples from the orientation allographs

In our implementation, 25 prototypes were chosen from either side of the mean distance value. The mean value is represented in the figure by the line in the middle (dashed and dotted). The dotted lines above and below the mean value represent the chosen upper and lower bounds of the distance value. The samples that have a cumulative distance lying within this range are selected as prototypes. Thus, an equal number of prototypes for each character are derived. With 50 prototypes per character, a total of 7800 prototypes are obtained, to match at the testing phase, which is a significant reduction compared to the entire set of around 45000 prototypes.

It is observed that each character has its own complexity in writing. Some characters are simple and thus the variation in writing them is less compared to complex characters. Having a fixed number of prototypes representing a character may increase the number of prototypes required to represent the character. This in turn increases the computation time for recognition. The third technique takes into account this factor and selects a variable number of prototypes for each character class from the training set and is explained in the next subsection.

4.3. Growing of the prototype set

In this approach, the prototype set evolves and grows from an initial set during the course of prototype selection. Initially we arbitrarily select one sample per class from the training set. Hence our initial prototype set has 156 samples corresponding to 156 character classes. Once the initial set of 156 samples is selected, the other samples from the training set are compared with this set, as follows. Consider the first character of this set. At the beginning of the prototype selection algorithm, there is only one prototype corresponding to this character. Then the next sample of this character from the training set is matched with this set of 156 samples (each belonging to a different character class). The best match is determined by considering the character that has the least DTW distance with the input sample. If this character corresponds to the correct class, the input sample is not added to the prototype set of that class. Otherwise, the input sample is added as a new prototype of that class. When the subsequent samples are considered, the original prototypes as well as all the ones that are added in the previous steps are considered for determining the match. As a result of this process, it can easily be observed that each class has a variable number of prototypes. Intuitively this is because a simple character has fewer variations in writing compared to a complex character. The block diagram of this process is shown in Figure 4.

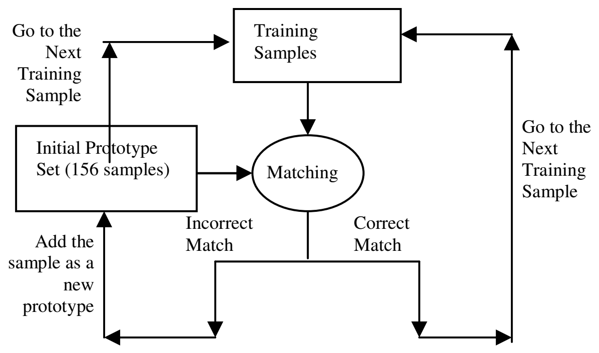
Go to the Next Training Sample
Training Samples
Initial Prototype Set (156 samples)
Matching
Go to the Next Training Sample
Incorrect Match
Correct Match
Add the sample as a new prototype

Figure 4. Block diagram of prototype set growing process

5. Recognition framework

The prototypes that have been selected are then used for recognizing the test samples. Two different recognition strategies have been used. The recognition strategy has been adopted from the techniques mentioned in [6]. The first technique is a two level technique wherein the Euclidean distance is computed at the first level. The Top 10 distinct classes are considered for matching at the second level. DTW distance is used as the distance metric at the second level.

The second technique employed is a two level DTW distance method. The characters are first down-sampled (by a factor of two) by extracting dominant points based on the quantized slope at every point [2, 3, 5]. To decide whether a point is dominant one or not, we calculate the slope angle of the segment between two consecutive points and the angle is quantized uniformly into eight levels. Dominant points of a character are those points where slope changes by more than one quantization step. At this level, each sample has a variable number of points. The DTW distance is obtained with respect to all the prototypes. As before, the Top 10 distinct classes are considered at the next level. Here, all the 60 points of the character are considered for calculating the DTW distance.

Both writer independent and writer adaptive scenarios were considered for recognition [4]. In the writer independent case, the training set consisted of the data of 30 users, from which the prototypes were selected. The test data consisted of the data of the remaining 10 users. In the writer adaptive case, we start with the above selected prototype set. Effect of adaptation is independently tested for each of the 10 users, where data was not used for the prototype selection. Each user has 10 trials, 7 out of which are used to adapt the prototype set for that user while the remaining 3 trials are used for testing the adapted prototype set.

6. Experimental results and discussion

In this section, the results of the experimentation with the different strategies described for prototype selection as well as recognition are presented. The parameters evaluated for each strategy are average recognition time and accuracy. The average recognition speed is computed by dividing the total number of recognized characters by the time taken to recognize them. The entire experimentation was performed using a C++ implementation on an Intel Pentium IV processor with 256 MB RAM.

First, the results of the different prototype selection algorithms are presented. The training set consisted of the data of 30 users. Using the cumulative pair-wise distance-based technique, a total of 7800 prototypes (50 prototypes for each class) was obtained. The method based on orientation allographs also yielded the same number of prototypes. Thus there is a reduction of about 82 percent in the number of training samples to be considered for recognition.

The technique of prototype set growing resulted in 7774 prototypes, which is quite close to the number obtained through the first two methods. The minimum number of prototypes obtained for a class was 7 (corresponding to class 120), while the maximum was 124 (corresponding to class 125). Each class, on an average, has 50 prototypes.

6.1. Results for writer independent

These prototypes were then used first for writer independent (WI) character recognition. Figure 5 summarizes the results obtained for the WI recognition strategy using the different prototype sets. The dominant point DTW distance measure was used. The plot shows the top choice accuracy for different test users for the same recognition scheme but for different prototype sets. It can be observed that all the three schemes yield similar recognition rates.

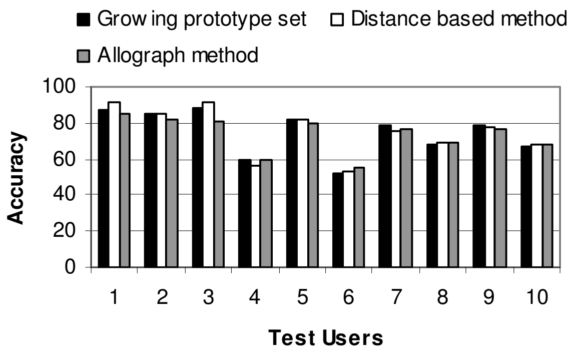

Figure 5. Performance of the different prototype sets for WI Recognition

Table 1 compares the performance of the three sets of prototypes for Writer Independent recognition. It presents the average Top 1, Top 5 and Top 10 accuracies. It also gives the figures for the number of characters classified per second. It can be observed that there is no significant variation in the Top 1 accuracies for the three sets. However there is a marginal improvement in the Top 5 and Top 10 accuracies for both orientation allograph and growing prototype set techniques. We also observe that, there is not much variation in time for recognition among the various approaches. This is because the recognition strategy is the same and the number of prototypes is also almost the same.

Two different distance measures are used for the purpose of writer independent recognition. The first one is based on the Euclidean distance measure while the second one was dominant point DTW (DDTW). The prototypes generated from the prototype set growing

Table 1. Performance of different prototype sets for Writer Independent Recognition

Selection method / Total No. of Prototypes	Average Accuracies			Speed (Chars/ Sec)
	Top 1	Top 5	Top 10	
Distance Method / 7800	74.9	91.0	93.6	1.5
Orientation Allographs / 7800	73.5	92.4	95.7	1.6
Growing prototype method / 7774	75.7	94.2	96.3	1.5

technique were used for this comparison. The Top 1 accuracy was obtained by finding the majority among the Top 5 distinct classes. The results are summarized in Figure 6.

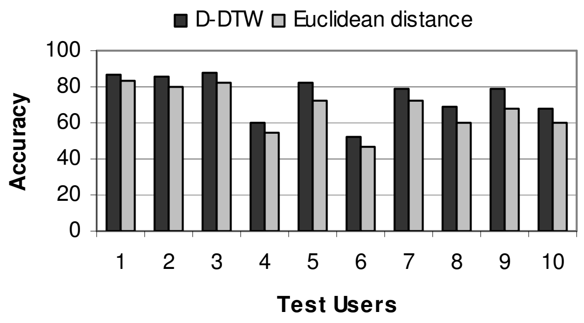

Figure 6. Recognition accuracies of DDTW and Euclidean methods

From this, it can be seen that the dominant point DTW recognition technique gives better recognition accuracy, though the Euclidean distance algorithm has the advantage of better time complexity. Both techniques feature two levels of recognition. At the first level, the dominant point DTW technique, computes the DTW distance between the input sample and the prototypes with one level of down sampling based on the dominant points. At the second level, the Top 10 distinct classes are considered for computing the DTW distance between the input sample and the prototypes without any down sampling. The second technique computes the Euclidean distance between the input sample and the prototypes, at the first level without any down sampling. The Top 10 distinct classes from this level are considered for computing the DTW distance at the second level. It was observed from the first technique that computing the DTW distance at the second level did not affect the accuracies at that level much. This can be seen from the graph shown in Figure 7. This implies that the second level of recognition is adding to the computational time while not really adding significantly to the recognition accuracy.

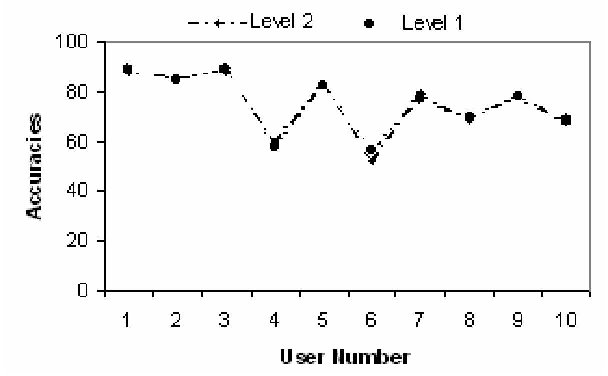

Figure 7. Accuracies at the two levels

6.2. Results for writer adaptation

The vast variety of writing styles is a challenge for character recognition systems. However, systems such as hand held devices are used in a limited user environment and to a large extent, are user specific. Hence, they can be tuned to a particular user. These devices possess limited memory and processing capabilities compared to normal sized computers. The writer adaptation which converts a writer independent recognition system into a writer dependent system, reduces the recognition time and memory consumption. The recognition accuracy also increases.

Table 2. Writer adaptation results

Method	Samples Added (out of 1092)	Average Recognition Accuracy	
		Top 1	Top 5
Distance Method	201	87.0	97.1
Orientation allographs	217	85.1	97.1
Growing Prototype Set	232	82.6	97.5

The writer adaptation results for the different users are presented here. For each user, out of his/her 10 trials for each character, 7 are considered for adapting while the remaining is used for testing. During training, the sample is added as a new prototype only if it is misclassified. For a given user, 1092 samples are used for training and the remaining 468 samples are used for testing the adapted prototype set. The results of this scheme with the prototypes generated using the three methods discussed are summarized in Table 2. By comparing the results in Table 2 with that in the Table 1, it is clear that the accuracy of recognition is improved in writer adaptation.

7. Conclusions

In this paper, different techniques for prototype selection for online handwritten character recognition are presented. These techniques were implemented and the results were compared on a database of isolated handwritten Tamil characters. Both writer independent and writer adaptive scenarios were investigated.

It is observed that for a multi-level classification scheme, the growing prototype set technique performs better than the other techniques. The dominant point DTW technique gives promising results for the classification at the first level. Subsequent stages of classification should be based on some other scheme.

Use of DTW distance for the first stage of recognition and including more features for the DTW in the second stage and finding a new recognition scheme for the second stage is the future work to be carried out.

8. References

[1] S. D. Connell and A. K. Jain, "Template-based online character recognition", *Pattern Recognition*, 34, 2001, pp. 1-14.

[2] X. Li and D. Y. Yeung, "Online handwritten alphanumeric character recognition using dominant points in strokes", *Pattern Recognition*, vol. 30, no. 1, 1997, pp. 31-44.

[3] V. Vuori and J. Laaksonen, "A comparison for automatic clustering of handwritten characters", Proc. International Conf. on Pattern Recognition, vol. 3, 2002, pp. 168-171.

[4] V. Vuori, J. Laaksonen, E. Oja, J. Kangas, "Experiments with adaptation strategies for a prototype based recognition system for isolated hand written characters", IJDAR, vol. 3, 2001, pp.150-159.

[5] V. Vuori, J. Laaksonen, E. Oja, J. Kangas, "Speeding up online recognition of handwritten characters by pruning the prototype Set", Proc. 6th ICDAR, 2001, pp. 501-505.

[6] Joshi N., Sita G., Ramakrishnan A. G., Madhavanath S. "Comparison of elastic matching algorithms for online Tamil handwriting recognition", Proc. 9th IWFHR, 2004, pp. 444 - 449.