Chapter 7

On-line recognition of handwritten characters

Vuokko Vuori, Matti Aksela, Jorma Laaksonen, Erkki Oja

7.1 Introduction

Automatic on-line recognition of handwritten text has been an on-going research problem for four decades. It has been gaining more interest lately due to the increasing popularity of hand-held computers, digital notebooks and advanced cellular phones. Traditionally, man-machine communication has been based on keyboard and pointing devices. These methods can be very inconvenient when the machine is only slightly bigger or same size as human palm. Therefore, handwriting recognition is a very attractive input method.

The most prominent problem in handwriting recognition is the vast variation in personal writing styles. There are also differences in one person's writing style depending on the context, mood of the writer and writing situation. The writing style may also evolve with time or practice. A recognition system should be insensitive to minor variations and still be able to distinguish different but sometimes very similar-looking characters. Recognition systems should, at least in the beginning, be able to recognize many writing styles. Such user-independent systems that allow free writing style usually have quite limited recognition accuracies. One way to increase performance is adaptation, which means that the system learns its user's personal writing style.

The goal of the On-line Recognition of Handwritten Characters project has been to develop adaptive methods for on-line recognition of handwritten characters. In this case, adaptation is to be understood in its most demanding sense, i.e. that the system is able to learn new writing styles during its normal use. Due to the learning, the user can use his own natural style of writing instead of some constrained style. Our work has concentrated on recognition of isolated alphanumeric characters and has been carried out in co-operation with Nokia Research Center in years 1997–2002.

The recognition is based on using a set of prototype characters stored in the memory of the system. The input characters are then classified on the basis of their Dynamic Time Warping (DTW) distances to the prototypes. A prototype-based recognition system can easily be adapted to a new writing style by modifying the prototype set: new prototypes can be added, existing prototypes can be reshaped so that they better represent the user's writing style, and prototypes which are not used or which cause more erroneous classifications than correct ones can be inactivated. According to our experiments, best results are obtained if all these three modes of adaptation are used together.

Lately, the character recognizer has been implemented in a Compaq iPAQ PDA device running Linux operating system. Additionally, support for recognition of entire words instead of single characters has been added. This mode is based on a simple language model that uses a dictionary of words. The adaptation of the character prototypes is then carried out after the user has accepted the written word from the given list. Figure 1 depicts this situation: The user has written the characters 'a', 'u', 't', 'o' and the system has recognized each of the characters correctly as shown on the top row of the pop-up list. Also, the corresponding word "auto" has been found in the dictionary along with other words with decreasing similarity.

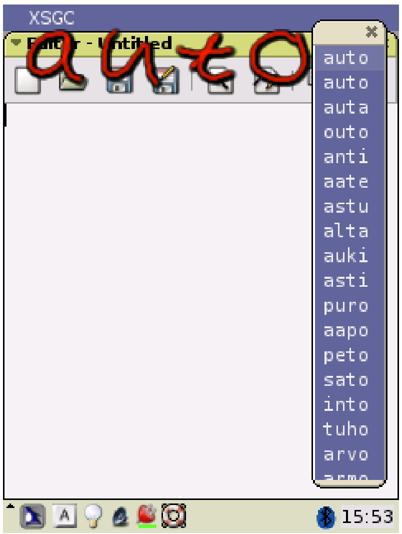

Figure 7.1: The user interface of the character recognition system running in a Linux PDA.

7.2 Adaptive prototype-based character classifiers

Vuokko Vuori, Jorma Laaksonen, Erkki Oja

With adaptive handwritten character recognition systems, it is essential to find a good initial recognition system which performs reasonably well, quickly and accurately, with all kinds of writers. The adaptation process has to be quick in the sense that the user does not have to input several character samples to teach the system a new writing style. In addition, the adaptation should be carried out in a self-supervised fashion during the normal use of the device, i.e. the correct classes of the input characters should be automatically deduced from the user's actions and responses to the recognition results. Naturally, the system should be robust against labelling errors of such training samples.

A prototype set which covers as many as possible alternative ways of writing characters is crucial for the initial recognition system to be able to work well with users using their natural writing styles. We have applied four hierarchical clustering algorithms to a large international database in order to create such a prototype set. In addition, we have experimented with two clustering indices to automatically determine the number of cluster, i.e. different prototypes. On the basis of the results of these experiments, we claim that a good set of prototypes can be formed from the combined results of the different clustering algorithms, but the number of clusters cannot be determined automatically and some human intervention is required [1].

One of the drawbacks of prototype-based classifiers is that the recognition time depends linearly on the size of the prototype set and on the complexity of the similarity measure defined for the prototypes and character samples. The computational complexity of the DTW algorithm depends quadratically on the average number of data points in the prototypes and character samples. We have designed a two-phase recognition scheme in which the prototype set is first pruned and ordered on the basis of a fast preclassification performed with heavily down-sampled character samples and prototypes. Then, the final classification is performed without down-sampling by using the reduced set of prototypes. Faster classification can also be achieved by posing stricter constraints for the nonlinear matching of the data points [1].

Another approach to speed up and enhance the recognition is to prune out those prototypes which are not used by the current user. We performed experiments in which writing styles of several writers were analyzed. The aim of the analysis was to find correlations in the usage of the prototypes and clusters of different writing styles. So the recognition system would be able to predict which prototypes could be pruned on the basis of character samples collected from the user and the estimation of the cluster in which user belongs to. The clustering analysis for the writing styles was performed with a Self-Organizing Map (SOM). The experiments showed that clusters of writing styles can be found, but the writers cannot be reliably assigned to them on the basis of a small set of arbitrary character samples [1].

When prototypes are always added in the adaptation, the recognition rate improves quickly, but the size of the prototype set tends to grow considerably. Therefore the prototype reshaping mode should be utilized too, as the recognition rates will then improve and the size of the prototype set remain the same. However, reshaping is not sufficient when used alone if the user's character samples and the prototypes are too different. According to our experiments, only two new prototypes per class would be enough for adapting the recognition system to a writing style. A prototype inactivation scheme is necessary if some of the samples are incorrectly labeled. Otherwise, the adaptation will be more harmful than useful if the probability of labeling errors is more than approximately 3-4 percent [2].

7.3 Adaptive committee techniques

Matti Aksela, Jorma Laaksonen, Erkki Oja

Combining the results of several classifiers can improve performance because in the outputs of the individual classifiers the errors are not necessarily overlapping. In addition, the combination method can be adaptive. The two most important features of the member classifiers that affect the committee's performance are their individual error rates and the diversity of the errors. The more different the mistakes made by the classifiers, the more beneficial the combination of the classifiers can be.

Selecting member classifiers is not necessarily simple. Several methods for classifier diversity have been presented to solve this problem. In [3] a scheme weighting similar errors made in an exponential fashion, the Exponential Error Count method, was found to provide good results. Still, the best selection of member classifiers is highly dependent on the combination method used.

We have experimented with several adaptive committee structures. Two effective methods have been the Dynamically Expanding Context (DEC) and Class-Confidence Critic Combining (CCCC) schemes [4]. The DEC algorithm was originally developed for speech recognition purposes. The main idea is to determine just a sufficient amount of context for each individual segment so that all conflicts in classification results can be resolved. In the DEC committee, the classifiers are initialized and ranked in the order of decreasing performance. Results of the member classifiers are used as a one-sided context for the creation of the DEC rules. Each time a character is input to the system, the existing rules are searched through. If no applicable rule is found, the default decision is applied. If the recognition was incorrect, a new rule is created.

In our CCCC approach the main idea is to try to produce as good as possible an estimate on the classifier's correctness based on its prior behavior for the same character class. This is accomplished by the use of critics that assign a confidence value to each classification. The confidence value is obtained through constructing and updating distributions of distance values from the classifier for each class in every critic. The committee then uses a decision mechanism to produce the final output from the input label information and critic confidence values. The adaptive committee structures have been shown to be able to improve significantly on their members' results [4].

References

[1] Vuokko Vuori *Adaptive Methods for On-Line Recognition of Isolated Handwritten Characters.* Doctoral Thesis, Helsinki University of Technology, 2002.

[2] Vuokko Vuori, Jorma Laaksonen, and Jari Kangas. Influence of erroneous learning samples on adaptation in on-line handwriting recognition. *Pattern Recognition*, 35(4):915–925, 2002.

[3] Matti Aksela. Comparison of classifier selection methods for improving committee performance. In *Proceedings of MCS2003*, pages 84–93, 2003.

[4] Matti Aksela, Ramūnas Girdziušas, Jorma Laaksonen, Erkki Oja, and Jari Kangas. Methods for adaptive combination of classifiers with application to recognition of handwritten characters. *International Journal of Document Analysis and Recognition*, 6(1):23–41, 2003.