Chapter 8

On-line recognition of handwritten characters

Vuokko Vuori, Matti Aksela, Ramūnas Girdziušas, Jorma Laaksonen, Erkki Oja

8.1 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, since its beginning in late 1997, 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. The project has earlier been a part of the TEKES technology programme Adaptive and Intelligent Systems Applications (AISA) and a subproject of the research project IMPRESS – Intelligent Methods for Processing and Exploration of Signal and Systems. The work is carried out in co-operation with Nokia Research Center.

We have performed series of experiments both with laboratory's large-scale computers and real hand-held PDA (personal digital assistant) devices. So far, we have performed genuine on-line user experiments with the PDA and the adaptive character recognizer by using a specially designed questionnaire program. The results have been promising and shown that users find the autonomous adaptation of the system comfortable. Currently, we are implementing our character recognizer to a new PDA with Linux operating system so that it could be used for textual input with all kind of applications. This way, we could experiment the adaptive recognition methods with more realistic tasks such as note making and e-mail composing.

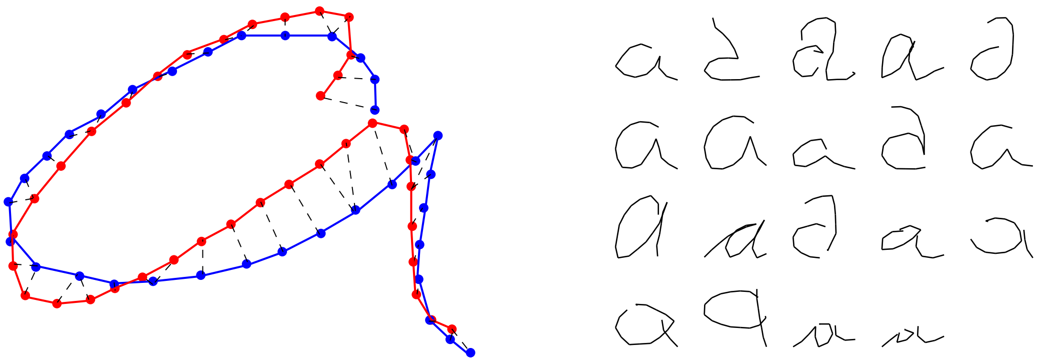

Figure 8.1: On the left: two handwritten characters 'a' are matched. On the right: a selection of different writing styles found for handwritten character 'a'.

8.2 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 perform reasonably well, quickly and accurately, with all kind of writers. The adaptation process of the recognition system 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. Ideally, the system would be able to learn a new style from a single sample. In addition, 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 deduced from the user's actions and responses to the recognition results by the system itself, instead of being specified by the user in some training mode. Naturally, the system should be robust against labelling errors of the 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.

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. In our work, we have applied similarity measures based on the Dynamic Time Warping (DTW) algorithm [1]. The computational complexity of the DTW algorithm depends quadratically on the average number of data points in the prototypes and character samples. Therefore, we 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. This way, the recognition time could be decreased by 76% while no new errors were introduced [2]. A faster classification can also be achieved by modifying the similarity measure by posing stricter constraints for the nonlinear matching of the data points performed by the DTW algorithm [3].

Another approach to speed up 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) [4]. These experiments showed that clusters of writing styles can be found but the writers cannot be realiably assigned to them on the basis of a small set of arbitrary character samples [5].

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 these three modes of adaptation are used together. If the first mode of adaptation is used alone, the recognition rate improves quickly but the size of the prototype set tends to grow considerably. The second mode of adaptation is also useful, the recognition rates improve and the size of the prototype set remains the same. However, it is not sufficient when used alone if the user's character samples and the prototypes are too different. For example in our work, character samples and prototypes are compared stroke-wise and a new prototype is needed if the number of strokes in the character matches with none of existing correct prototypes [6]. According to our experiments, only two new prototypes per class would be enough for adapting the recognition system to a writing style [7]. 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 [8].

References

[1] D. Sankoff and J. B. Kruskal. *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison.* Addison-Wesley, 1983.

[2] Vuokko Vuori, Jorma Laaksonen, Erkki Oja, and Jari Kangas. Speeding up on-line recognition of handwritten characters by pruning the prototype set. In *Proceedings of 6th International Conference on Document Analysis and Recognition*, Seattle, Washington, USA, pages 501–505, September 2001.

[3] Vuokko Vuori, Matti Aksela, Jorma Laaksonen, Erkki Oja, and Jari Kangas. Adaptive character recognizer for a hand-held device: implementation and evaluation setup. In L. R. B. Schomaker and L. G. Vuurpijl, editors, *Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, pages 13–22, Amsterdam, September 2000. Nijmegen: International Unipen Foundation. ISBN 90-76942-01-3.

[4] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, 1997. Second Extended Edition.

[5] Vuokko Vuori and Erkki Oja. Analysis of different writing styles with the self-organizing map. In *Proceedings of the 7th International Conference on Neural Information Processing*, volume 2, pages 1243–1247, November 2000.

[6] Vuokko Vuori, Jorma Laaksonen, Erkki Oja, and Jari Kangas. Experiments with adaptation strategies for a prototype-based recognition system for isolated handwritten characters. *International Journal on Document Analysis and Recognition*, 3(3):150–159, March 2001.

[7] Vuokko Vuori, Jorma Laaksonen, Erkki Oja, and Jari Kangas. Controlling on-line adaptation of a prototype-based classifier for handwritten characters. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume 2, pages 331–334, September 2000.

[8] 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.

8.3 Adaptive committee techniques

Matti Aksela, Jorma Laaksonen, Erkki Oja

Combining several different classifiers in a committee form is another way to reach for the best attainable recognition performance. Combining the results of several classifiers can improve performance because in the outputs of the individual classifiers the errors are not necessarily overlapping. Committee methods generally require more than one member classifier to recognize the input. In the near future, this will not be computationally too complex for even the smallest platforms for on-line handwritten character recognition, due to the continuous increase in available computational power.

The objective of a committee classifier is to combine the results of a set of member classifiers in a way that improves the overall performance. The two most important features of the member classifiers that affect the committee's performance are their individual error rates and the correlatedness of the errors. The more different the mistakes made by the classifiers, the more beneficial the combination of the classifiers can be.

We have experimented with several adaptive committee structures. Two of the most effective methods examined so far have been the Dynamically Expanding Context (DEC) [1] and the novel Class-Confidence Critic Combining (CCCC) scheme. The DEC algorithm was originally developed to create transformation rules that would correct typical coarticulation effects in phonemic speech recognition. The main idea behind the approach is to determine just a sufficient amount of context for each individual segment so that all conflicts in the set of training samples will be resolved. The DEC principle has been slightly modified to suit the setting of isolated handwritten character recognition [2]. In the DEC committee, the classifiers are initialized and ranked in the order of decreasing performance. The primary and the second-ranking results of every member classifier 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 performed by that particular member classifier. The confidence value is obtained through constructing and updating distance distributions of each class in every critic. The distance distributions model the distance from the input sample to its nearest prototype. 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 [3].

References

[1] T. Kohonen. Dynamically expanding context. *Journal of Intelligent Systems*, 1(1):79–95, 1987.

[2] M. Aksela. Handwritten character recognition: A palm-top implementation and adaptive committee experiments. Master's thesis, Helsinki University of Technology, 2000.

[3] M. Aksela, J. Laaksonen, E. Oja, and J. Kangas. Application of adaptive committee classifiers in on-line character recognition. In *Proceedings of International Conference on Advances in Pattern Recognition*, pages 270–279, 2001.

8.4 Discriminative classifiers

Ramūnas Girdziušas, Jorma Laaksonen, Erkki Oja

Hidden Markov Model (HMM) classifiers [1] have been considered as an alternative to the DTW distance classifiers in order to produce recognizers with uncorrelated recognition errors. HMMs are probabilistic generative models whose parameters are estimated by optimizing a cost criterion. Usually HMM classifiers provide a maximum log-likelihood (ML) estimate C_{ML} for sequences of a particular class. Alternatively, an HMM can be trained to discriminate between the classes by maximizing the mutual information (MMI) measure C_{MMI} between the observed sequences and their models. The ML approach is numerically efficient, but often results in classifiers with over-fitted parameters. On the other hand, the MMI alternative allows attaining lower recognition error rates, but it requires computational resources that are not available when one considers palm-top device implementations. Both approaches are well justified on statistical grounds, but they yield sub-optimal recognition machines when confronted with practical requirements.

An important observation is that the criteria C_{ML} and C_{MMI} are functions of the HMM scaling factors [1]. We have shown [2] that under certain assumptions, these quantities can be regarded as the average mutual information between the generated sequence and the model's state at a particular time instant. The discriminant functions of the classifiers obtained by optimizing C_{ML} and C_{MMI} have been replaced by a kernel function expansion in the scaling factors, designed akin to the support vector machine (SVM) methodology [3]. The classifier functions in such a way that the character sequence is first mapped into very high-dimensional feature space. In that space, each dimension represents the mutual information between the generated sequence and the state occurrence at a fixed time instant, averaged over the states of the prototype model. Linear binary separating hyperplanes are then built to maximize the class-separating margin.

The created HMM-SVM classifier was shown to have certain advantages over the traditional HMM approaches. The method decreases the error rate by 12 %-units compared with the ML-HMM classifier, to the level of 25%. The developed classifier attains the same error rates as the SVM classifier working on character bitmaps. However, it requires less than 300 character prototypes for writer-independent task rather than roughly 6000 support vectors extracted from the character bitmaps. Compared to the MMI approach, the HMM-SVM does not require gradient ascent optimization in the high-dimensional parameter space. The within-kernel parameters are determined by the ML design and the coefficients in the kernel function expansion are determined by the quadratic programming algorithm. Its dual formulation has computational complexity independent of the number of prototype models and the length of the character sequences. Also, the training phase does not have the local minima problem inherent in the MMI approach.

References

[1] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[2] R. Girdziušas. Discriminative on-line recognition of isolated handwritten characters. Master's thesis, Helsinki University of Technology, November 2001.

[3] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science. Springer, 2000.