published in the proceedings of IWFHR98, Taejon, Korea, August 1998, pp. 475-484
available at <URL:http://www.cis.hut.fi/jorma/papers/>

EXPERIMENTS WITH A SELF-SUPERVISED ADAPTIVE CLASSIFICATION STRATEGY IN ON-LINE RECOGNITION OF ISOLATED HANDWRITTEN LATIN CHARACTERS

Jorma LAAKSONEN, Jarmo HURRI, and Erkki OJA
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O. BOX 2200, FIN-02015 HUT, Finland
E-mail: {jorma.laaksonen,jarmo.hurri,erkki.oja}@hut.fi

Jari KANGAS
Nokia Research Center
P.O. Box 100
FIN-33721 Tampere, Finland
E-mail: jari.kangas@research.nokia.com

Results on a comparison of recognition techniques for on-line recognition of handwritten Latin alphabets are presented. The emphasis is on an adaptive classification strategy introduced in this paper. The classification strategy is based on compressing or distilling a large database of handwritten characters to a small set of character prototypes. The distillation is performed as a clustering procedure to which the number of clusters are given for each character class. Each user's personal characters are then added to this set as they become available. The classification decision uses the 1-Nearest Neighbor (1-NN) rule for the distances between the unknown character and the stored prototypes. The distances are calculated using Dynamic Time Warping (DTW). The benefits of the proposed method include the automatic unsupervised learning from user input simultaneously with the normal mode of operation. The presented experiments show that the recognition system is able to adapt to the user's writing style with only a very few – say some tens of – handwritten characters.

Keywords:
on-line recognition; adaptive classification; handwriting recognition; character recognition; handprinted character recognition; memory based learning; incremental learning

1 Introduction

We present the results of a series of experiments which assessed the development of an adaptive classifier for on-line recognition of handwritten Latin characters. We have had a series of work hypotheses on our mind. First, that information about the typical ways characters are written can be automatically extracted – or distilled – from a database containing data from numerous writers. Second, that an efficient classification system can be built from this distilled data. And third, that the performance of the resulting classifier can be enhanced by making it adapt to each writer's specific writing style.

The aim of the study has been to evaluate the potential of a totally *unsupervised adaptive handwriting recognizer*. The system should be able to utilize the user input to enhance the classification accuracy simultaneously when it is used. No separate training period should therefore be necessary. The system uses its own recognitions in a *self-supervised* fashion: the recognitions which are not indicated as failures by the user are regarded as correct. Similarly, the user-reported misclassifications are used to revise the system.

In the experiments, we have compared the recognition accuracy achieved by our proposed adaptive classifier against the accuracies of four other classifiers we have formed from the same data. The results show very promising performance for the adaptive classifier.

2 Used Database

In order to get data for our experiments we developed a data collecting program and collected a set of about 350 separate lowercase Latin characters 'a'-'z' from each of 22 writers, resulting in an approximate total of 7750 sample characters. Because of the preliminary nature of this first phase of research the writers were chosen from a relatively narrow cultural group: the set of laboratory employees.

The main features of the data collecting approach were as follows.

- Users were guided with voice.
- After a short training period the user was unable to see his / her handwriting result.
- Users wrote single printed characters one at a time.
- Successive characters could be written on top of each other.
- Users had to write both single characters as such and words separated into single characters.
- The alphabet consisted of letters used in the Finnish language augmented with some typical special characters, which, however, were not considered in this particular series of experiments.

2.1 Data Acquisition Equipment

The data collection system consisted of a Silicon Graphics Indy workstation and an attached Wacom II drawing tablet. Because the users had to write only single characters the input area of the tablet was constrained to a small rectangle with size 2cm$\times$4cm. The drawing tablet and its driver provided both pressure and timing information. The former was used in locating the beginnings and ends of the drawn strokes but the latter information was in these experiments ignored. The pen movements were recorded with approximately 20ms spacing between samples.

The data recordings were stored in UNIPEN[7] version 1.0 files.

2.2 Data Preprocessing

The pressure information provided by the tablet was utilized when the individual strokes were extracted from the measurement data. The detection of the *pen down* and *pen up* events were detected by thresholding the pressure values with a three-parameter hysteresis model. In this model, the first parameter specified the minimum pressure for the *pen down* event. The second parameter determined the minimum value the pressure should reach so that the stroke would be accepted. The third parameter specified the pressure limit falling below of which triggered the *pen up* event. All these three parameters were selected independently for each writer with an iterative optimization process. The result of the preprocessing stage were characters with varying number of strokes, each stroke given as a series of xy-coordinates. Each character was finally normalized by subtracting the mean of all points in the character. No size normalization was performed.

3 Types of Classifiers

A novel adaptive classification strategy will be proposed in this section. Before that, an intermediate recognizer used by the adaptive classifier and here called the *distilled classifier* is explained. The accuracy of the proposed technique will be compared against the attainable accuracies of various other kinds of classifiers in Section 5. In order to facilitate a comparison, three more classifiers were produced from the same database. These rival techniques are also explained in this section.

All the presented classifiers are based on the same prototype-based architecture and the same database of handwritten characters, only the way the samples are processed is different between the classifiers.

3.1 Distilled Classifier

We call *distilled classifier* the recognition system upon which the adaptive classifier is built. The word *distilled* originates from the principle of minimizing the amount of memory used in storing the prototypes of the characters. We believe that even though every writer has her own way of writing, each character class can be described accurately enough with a handful of samples.

In our implementation, the hypothetical handwriting recognition system is shipped with a small precompiled set of prototype characters which together form the distilled classifier. The actual generation of this set is elaborated in the following section. Let it suffice for now that the system is able to recognize say 90% of its input correctly when used by an average writer.

3.2 Adaptive Classifier

As a *self-supervised adaptive classifier* we mean a recognition system which is ready to operate as such but which is able to benefit from user input. As such, the adaptive classifier is thus built upon the distilled classifier.

When the system starts to receive input, the input characters are stored in the memory of the system as new prototypes. The correct classes of these new prototypes can be deduced from the user actions: if the user accepts the tentative recognition made by the system, then the tentative classification is correct. In the opposite case it is assumed that the user provides the correct class of the input character, which then can be utilized. This process gives rise to the attribute *self-supervised* in the name of the classifier.

The number of the user's prototypes stored in the memory of the system is affected by the available memory and computational resources. It is assumed that the classification rate would increase when the number of prototypes is increased until it saturates to some final accuracy level.

3.3 Universal Classifier

We call here *universal classifier* a classification system which is equipped with data from a multitude of writers. After the classifier is created and the system shipped, no modifications can take place. Therefore, the universal classifier is not adaptive. The performance of the system is highly dependent on the amount of memory available for storing the character prototypes available during the manufacturing. Also, it depends on the comprehensiveness of the samples in the database.

We have made the assumption that the number of handwritten lowercase letters we have in total gives a rough estimate of the potential of such a classification system.

3.4 Personal Classifier

We call here *personal classifier* a classification system which is trained for each user separately. Initially, the system has no classification capability at all. After receiving a pre-fixed number of pattern characters from one writer, the system starts to operate as a recognizer dedicated to that particular user.

We have assumed that the number of handwritten lowercase letters we have for each writer in our database – an average of 14 per character class – is enough to produce a personal classifier. About the same quantity of prototypes has also been used in other recognition systems.[4]

3.5 Utopian Classifier

The name *utopian classifier* is given to a classification system which is initially created as the universal classifier from potentially infinite training data. After the user has received the system, the system has then been adapted to her personal writing style during a training period during which potentially infinite amount of data has again been available. As such, the utopian classifier would be optimal in any use. Of course, such a system can never be realizable. As a model of an ideal case of recognition system design, however, the concept of the utopian classifier is useful.

We have approximated the utopian classifier by adapting the universal classifier with the same amount of data that is used in the creation of the personal classifier.

4 Implementation Details

This section further elaborates some details in the implementation of the self-supervised adaptive classifier. These include the calculation of distances between the characters, the creation of the distilled classifier, and the classification rules for the recognition of the characters.

4.1 Distances Between Characters

A key factor in the implementation of on-line character recognition systems is the definition of the distance function between two stroked characters. In our experiments, the distance was defined to be infinite if the number of strokes is different in the two characters. Otherwise, the distance between two characters was calculated as the sum of the distances between the corresponding strokes. The order and direction of the strokes were always kept fixed.

The distance between two strokes was calculated as point-wise elastic matching[6] using the standard Dynamic Time Warping (DTW) technique[5] with the continuity constraint:

$$(\Delta\mathbf{i}_0, \Delta\mathbf{j}_0) = \begin{cases} (1,0) & \text{or} \\ (1,1) & \text{or} \\ (0,1)\ . \end{cases} \tag{1}$$

These three permissible steps were all treated equally and the cost associated with each point pair in the sequences was the squared Euclidean distance between the corresponding points in the xy-plane. In this sense, the implementation of DTW differs from those versions in which the strokes are presented not with xy-coordinates but with strings of symbols corresponding to quantized directions of pen movement.[3]

4.2 Automated Prototype Set Distillation

The distilled prototype set was formed using a clustering procedure in which the number of different styles of writing a particular character was used as *a priori* information. The numbers of writing styles present in our database were enumerated manually before the distillation of the prototype set. These numbers were then iteratively refined after observing the results of the clustering.

In the clustering procedure, the Dynamic Time Warping distances between all pairs of characters in each class were calculated. First, the character prototype which yielded minimum sum of distances to all other prototypes was selected as the cluster center. Next, the characters furthest away from the center were tentatively used to form a new cluster in a similar manner. The distance from the center of the new cluster to the furthest prototype in that cluster was recorded. This distance was then summed with the maximum distance between the original cluster center and the prototypes left in that cluster. The resulting value is roughly a u-shaped function of the number of prototypes transferred from the original cluster to the new one. The minimum location of this function was then used to actually split the characters to two clusters. The centers of the clusters were located again and each prototype was assigned to that cluster the center of which was nearest to it. This iteration was continued until a stable settlement was reached.

The above described cluster splitting and reorganization procedure was repeated so many times that the preset number of clusters was reached. The center character of each cluster was then used as a representative of that cluster which, hopefully, corresponded to one distinctive writing style. As a result of the clustering procedure, a distilled set of 104 prototypes was created.

4.3 Classification

In the classification of the characters, k-Nearest Neighbor (k-NN) classification rule[1] was used. Values one to six for k were tested and it was found out that with every classifier, the value $k = 1$ produced the best classification accuracy.

In the universal, personal and utopian classifiers, the prototype itself was never used when the nearest neighbors were searched for. If it were used, 100% classification accuracy would have resulted in 1-NN classification. However, the extracted 104 prototypes were also used as testing samples in both the distilled and adaptive classifiers. This produces slight bias towards lower error rate in the results. It may be argued, though, that the extracted prototypes would have been correctly classified anyhow because of their unambiguousness. Likewise, it is a bit questionable that every user's data was used in the distillation process even though these same writers later appeared as test subjects, which, of course, should have been independent.

5 Results of the Experiments

First, we created and used the four reference recognizers, the distilled, universal, personal, and utopian classifiers. The results are summarized in Table 1. In the table, the numbers of used prototypes are given to every type of classifier. The prototypes are characterized either as system or user prototypes on the basis of their origin.

Table 1: Classification accuracies of the four reference classifiers described in the text. The error percentages shown are averages of the results of the 22 writers involved. Likewise, the numbers of the used prototypes are averages of the figures of the 22 writers.

	Prototypes used			
Classifier	system	user	total	**Error %**
distilled	104	–	104	8.33
universal	7396	–	7396	7.51
personal	–	352	352	2.10
utopian	7396	352	7748	1.74

It seems that the amount of data for the construction of the universal classifier has not been adequate. This is reflected by the fact that the accuracy of the personal classifier is so clearly superior to the result of the universal classifier. Also, the distilled classifier performs almost as well as the universal classifier even though the number of used prototypes is even less than 1/70 of the prototypes in the universal classifier. Comparing the results of the personal and utopian classifiers, it may be stated that the latter is only slightly better than the former. This, again, is an indication of the insufficient number of writers in the database used.

Next, we created adaptive classifiers for each of the 22 subjects with increasing number of character prototypes added to the classifier. Each classifier was implemented 30 times and the recognition results were averaged in order to get representative figures for the accuracies. The original distilled prototype set, to which each user's personal samples were added, was of the size 104 samples. The results are shown in Table 2. The decrease of the classification error percentage when the number of added prototypes was increased is also seen in Figure 1.

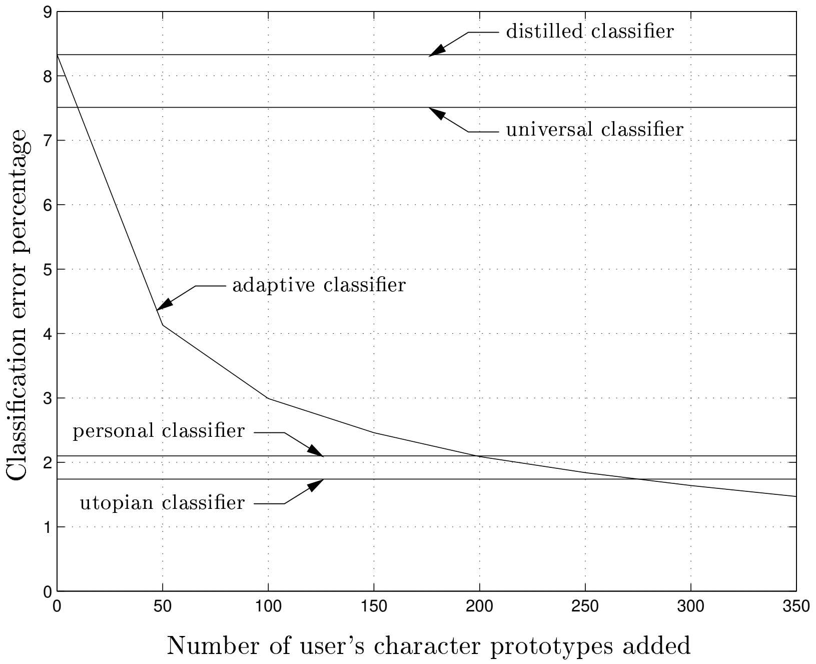
Table 2: Classification accuracies of the adaptive classifier with increasing number of user prototypes. The results are averages of 30 random repetitions for each of the 22 writers. The result in the first column is the result of the "distilled classifier" in Table 1.

Prototypes								
user	0	50	100	150	200	250	300	350
total	104	154	204	254	304	354	404	454
Error %	8.33	4.13	2.99	2.46	2.09	1.84	1.64	1.47

It seems that the classification error rate decreases monotonically as the number of user's samples increases. It even seems that the number of prototypes in our database for each writer – approximately 350 – does not suffice to saturate the enhancement of the accuracy. It might thus be argued that at least, say, the average of 20 prototypes per character class would be needed to enable the creation of a good adaptive classifier.

The nonintuitive result that the classification accuracy of the adaptive classifier with 300 user samples and more is better than the accuracies of the personal and utopian classifiers needs some clarification. First explanation might be that the adaptive classifier only contains very "pure" character prototypes from writers other than the actual user of the system. Therefore, these prototypes hardly ever cause misclassifications but give reliable information about the typical ways of writing the characters. If the number of user's own samples in the cases of the personal and adaptive classifiers were distinctively larger, the order of the two accuracies would presumably be reversed. The second explanation for the success of the adaptive classifier might lie in the fact, that even if the distilled prototypes may in some occasions produce misclassifications, these errors become rarer and rarer when the number of user's own characters increases.

Figure 1: The decay of the recognition error rate when the number of added user prototypes is increased. The horizontal lines depict the recognition accuracies of the four reference recognizer, i.e., the distilled, universal, personal, and utopian classifiers, from top to bottom.



6 Conclusions and Future Directions

One objective of our study was to evaluate whether the number of character prototypes we have in our database is adequate for the creation of the different classifier types or not. Interpreted in this sense, the results indicate that the number of prototypes given by each writer – approximately 350 – is almost enough to model the writer's writing style. On the other hand, the number of writers – 22 – is certainly too small to give a good representation of all the possible models of writing, even for this presumably homogeneous group of subjects. The fact that the best classification results for the universal and utopian classifiers were obtained with a 1-NN and not with a k-NN classifier with $k \geq 3$, may also be interpreted as an indication of the insufficient total number of prototypes in the database.

The small number of user's own prototypes needed in order to reach good classification accuracies with the adaptive classification strategy should be noted. It might even suffice to include to the set of classification prototypes those user's characters that were erroneously classified by the distilled classifier. By using this or some other variation of the classical Condensed k-NN algorithm,[2] the total number of prototypes used in the recognizer might not increase as fast as with the current method, when viewed as a function of the resulting classification accuracy.

As a future direction of this research project, we have planned including characters other than the lower case alphabets to the experiments. These additional characters would then include at least the upper case alphabets and numbers. Another perspective could be opened with a series of user experiments in which the users' impressions of the on-line improvement of the classification accuracy would be recorded.

References

1. E. Fix and J. L. Hodges. Discriminatory analysis—nonparametric discrimination: Consistency properties. Technical Report Number 4, Project Number 21-49-004, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.
2. P. E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(3):515–516, May 1968.
3. X. Li and D.-Y. Yeung. On-line handwritten alphanumeric character recognition using dominant points in strokes. *Pattern Recognition*, 30(1):31–44, 1997.
4. F. Nouboud and R. Plamondon. A structural approach to on-line character recognition: System design and applications. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(1&2):311–335, 1991.
5. D. Sankoff and J. B. Kruskal, editors. *Time warps, string edits, and macromolecules: The theory and practice of sequency comparison*. Addison-Wesley, 1983.
6. C. C. Tappert. Speed, accuracy, and flexibility trade-offs in on-line character recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 5(1&2):79–95, 1991.
7. The UNIPEN project. <http://hwr.nici.kun.nl/unipen/>, 1998.