published in the Proceedings of ICDAR'99, Bangalore, India, September 1999, pp. 792-795

On-line Adaptation in Recognition of Handwritten Alphanumeric Characters

Vuokko Vuori, Jorma Laaksonen, and Erkki Oja
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O Box 5400, FIN-02015 HUT, Finland
{vuokko.vuori,jorma.laaksonen,erkki.oja}@hut.fi

Jari Kangas
Nokia Research center
P.O. Box 100
FIN-33721 Tampere, Finland
jari.kangas@research.nokia.com

Abstract

We have developed an adaptive on-line recognizer suitable for recognizing isolated alphanumeric characters. It is based on the k nearest neighbor rule. Various dissimilarity measures, all based on dynamic time warping (DTW), have been studied. The main focus of the work is on on-line adaptation. The adaptation is performed by modifying the prototype set of the classifier according to its recognition performance and the user's writing style. These adaptations include: 1) adding new prototypes, 2) inactivating confusing prototypes, and 3) reshaping existing prototypes. The reshaping algorithm is based on Learning Vector Quantization (LVQ). The writers are allowed to use their own natural style of writing, and the adaptation is carried out during normal use in a self-supervised fashion and thus remains otherwise unnoticed by the user.

1. Introduction

Recognition accuracy is the key factor in determining the acceptability of a handwriting recognition system and the whole application in which it is implemented. In general, the required recognition rate for the recognizer is very high. Tests with keyboard typing have shown that the writers tolerate random errors up to 1% while 0.5% is unnoticeable and 2% is intolerable [1]. These error rates are lower than even human readers can attain. The average human recognition error rate for isolated, handwritten, and context free characters is approximately 4% [6, 7].

In the light of the results mentioned above, the required accuracy seems to be so high that it is very unlikely to be attained with a multi-writer system and without any constrains on the writing style. The work described in this paper concentrates on adaptation methods which are carried out in a self-supervised fashion during normal use. The purpose of the adaptations is to increase the recognition performance by turning a writer-independent system into a writer-dependent one. At the beginning of the use, the recognition performance of the system is relatively low as the users are allowed to use their own natural styles of writing. However, after entering few samples per each character class, the error rate will be considerably decreased. We study six dissimilarity measures and four adaptation strategies. First, a writer-independent recognition system was formed by finding the best combination of the preprocessing and normalization methods and the dissimilarity measure. Next, the capabilities of the adaptation strategies to turn the writer-independent system into a writer-dependent one and to increase the recognition performance are examined.

2. Adaptive elastic matching classifier

The present classifier is based on template matching, namely on the Condensed Nearest Neighbor (CNN) rule [2]. The CNN rule classifies unknown patterns in a similar manner as the k-NN rule but only a consistent subset of the training samples – not the whole training set – is used as the prototype set. In this work, the prototypes are selected by a clustering algorithm instead of the originally proposed condensing algorithm.

The clustering algorithm [5] is semiautomatic. This means that the number of clusters, or prototypes, must be predefined. The number of different writing styles per character class and stroke number variation was manually examined. In the current system, the same number of prototypes (7) is used for each class. The number of prototypes per stroke number variation is selected so that it roughly corresponds to the respective share of all the writing styles of that character. The prototypes are the centermost items of the clusters.

In this work, connected parts of the drawn curve in which the pressure between the pen and writing surface exceeds a given value are considered as strokes. Prior to the matching phase, the prototypes are pruned according to their number of strokes – only characters with the same number of strokes are matched. In addition to pruning, the prototypes are ordered according to their categories based on the locations of the starting and ending point of the first stroke [9]. The ordering of the prototypes saves computation time as the matching of an unknown character and a prototype is interrupted when the dissimilarity measure exceeds the kth smallest dissimilarity value evaluated so far.

As the pen position is sampled with a constant sampling frequency, while the size of the characters, writing speed and style vary, the number of data points per character is not constant. This is true even if the pen point movements are resampled in order to get spatially equidistant data points.

The six dissimilarity measures experimented with the recognition system are based on dynamic time warping (DTW) and thus suitable for nonlinear curve matching [8]. The main difference between the dissimilarity measures is the associated cost of matching a data point. The dissimilarity measures are defined on stroke basis and the data points are matched in the same order as they have been produced.

The dissimilarity measures are called *Point-to-point* (*PP*), *Normalized point-to-point* (*NPP*), *Point-to-line* (*PL*), *Normalized point-to-line* (*NPL*), *Kind-of-area* (*KA*) and *Simple-area* (*SA*) distances. *PP*-distance uses the squared Euclidean distance between the data points as a matching cost. In the case of *PL*-distance, the data points are matched to lines interpolated between the data points. *NPP*- and *NPL*-distances are otherwise similar to *PP*- and *PL*-distances, respectively, but the sums of the matching costs are divided stroke-wise by the number of matchings. *KA*-distance also matches data points against data points but the Euclidean distances from the matched data points to their neighboring data points are considered too. *SA*-distance uses the area between the strokes approximated with triangles or quadrilaterals as the matching cost. These dissimilarity measures are explained in more detail in [9].

3. Adaptation strategies

Adaptation strategy named *Add* examines the classes of the k prototypes nearest to the input character. The classification is carried out according to the k-NN rule. The input character is added to the prototype set if any of these prototypes belongs to a wrong class, even if the classification was correct.

Adaptation strategy *Inactivate* is used for inactivating those prototypes which are more harmful than useful. After each recognition, *Inactivate*-strategy checks if the prototype nearest to the input character has been the nearest one sufficiently many times and whether its class has been incorrect more often than correct. In that case, the prototype is removed from the set of active prototypes.

When a character written by the user is basically similar to a prototype of the correct class, for example it has the same number and order of strokes, but of slightly different shape, the existing prototype can be reshaped instead of adding the input character to the prototype set. This can be performed with an adaptation strategy called *Lvq* based on a modified version of Learning Vector Quantization (LVQ) [3, 4].

Adaptation strategy *Hybrid* combines adaptation strategies *Add* and *Lvq*. The k nearest prototypes are examined. If any one of them belongs to the same class as the input character, the nearest prototype is modified with *Lvq*. Otherwise, the input character is added to the prototype set.

4 Data used in the experiments

All the characters were collected with a Wacom ArtPad II tablet attached to a Silicon Graphics workstation. The resolution of the tablet is 100 lines per millimeter and the sampling rate is at maximum 205 data points per second. The loci of the pen point movements consist of the x- and y-coordinates and pen's pressure against the writing surface. The characters were written one at a time. Writers were advised to use their natural handwriting style.

The first database (Database 1) consists of about 10 400 characters which were written by 22 subjects without any visual feedback. The distribution of the classes (a-z, å, ä, ö, A-Z, Å, Ä, Ö, 0-9, and some special characters), was somewhat similar to that of the Finnish language. The rest of the databases (Databases 2-4) were collected with a program which showed the pen trace on the screen and recognized the characters on-line. After each misrecognition, the subjects were asked to confirm the correctness of the input character. The characters were written in same random order by all the subjects, and the distribution of the classes (a-z, å, ä, ö, A-Z, Å, Ä, Ö, 0-9) was nearly even. Databases 2-4 contain approximately 13 200, 21 200, and 8 100 characters from 8, 16, and 8 writers, respectively. None of the writers of Database 1 appeared in any of the Databases 2-4.

5. Preprocessing and normalization operations

The characters were collected with a system whose properties, such as sampling rate and resolution, are beyond the capabilities of the existing hand-held devices. Therefore, it is important to examine how sensitive the recognition method is to the amount of data for each character. The sampling frequency and method could be altered with operations called *Decimate*(n), *Interpolate*(n), or *EvenlySpacedPoints*(n). *Decimate*(n) keeps every $(n+1)$th data point and abandons the intermediate ones. *Interpolate*(n) does just the opposite – it interpolates n equally spaced points between every original data point pair. The data points can be made spatially equidistant with *EvenlySpacedPoints*(d)-operation, where d is the relative distance between the adjacent points.

The unknown character and the prototypes are shifted into the same location by moving their center points to the origin of the coordinate system. The normalization method *MassCenter* moves the mass center of the character to the origin and the normalization method *BoundingBoxCenter* does the same to the center of the bounding box.

The size variations in the characters are normalized in all the experiments by scaling the size of the character so that the length of the longer side of the bounding box is set to 1 000 units and the aspect ratio remains unchanged.

6. Experiments

In the first experiments, the six dissimilarity measures were compared with each other. The most suitable preprocessing and normalization operations were selected separately for each of them. Lower case letters and digits were used in the experiment. The initial prototype sets were formed by using Database 1, and Database 2 was used as a test set. *MassCenter*-normalization was found to be the most suitable centering method for all the dissimilarity measures. The total error rate and the best preprocessing method for *PP*-, *PL*-, *NPP*-, *NPL*-, *SA*-, and *KA*-distances were: 15.93% with *Decimate*(2), 17.07% with no preprocessing at all, 16.13% with *Decimate*(1), 17.07% with *Decimate*(2), 29.75% with no preprocessing at all, 18.25% with *Decimate*(2), respectively.

After choosing the best combination of the dissimilarity measure, preprocessing and normalization operations, several parameter values were tested for each adaptation strategy. Again, all the experiments were carried out by using lower case letters and digits, and the initial prototype set was created by using Database 1. Database 3 was used as a design set and the characters of each writer were recognized one by one in the same order as they were collected. The adaptation was performed after each classification. The parameter values which yielded the lowest total error rates were selected. Finally, the results were verified with Database 4. The error rates (E) for both the design and verification sets are presented in Table 1. Subscripts indicate the character sets in question. Superscript *tot* means that all the characters were used in the evaluation of the error percentage whereas *final* means that only the last 200 characters per each subject were observed.

Table 1. Recognition error percentages of different adaptation strategies. *INP* **shows the growth percentage of the prototype count.**

Adaptation strategy	E^{tot}_{design}	$E^{tot}_{verif.}$	$E^{final}_{verif.}$	*INP*
None	14.33	14.13	14.06	0
Add	2.93	3.12	1.81	66
Lvq	6.81	9.89	8.63	0
Add & Inactivate	2.95	3.04	1.56	65
Hybrid	3.06	4.18	2.50	2
Hybrid & Inactivate	3.33	4.26	2.75	2

From Table 1 it can be seen that all the adaptation strategies can improve the recognition accuracy significantly. Adaptation strategy *Add* combined with *Inactivate* yields the lowest final error rate. However, the size of the prototype set increases considerably. The total error obtained with *Hybrid*-strategy is also quite good. Clearly, *Lvq*-strategy is not sufficient when used alone, but it is truly useful as good recognition results can be obtained by using *Hybrid* which adds only few new prototypes. The probabilities that a new sample will be added to the prototype set were also examined. In the case of *Add*-strategy, the initial probability is nearly 0.7. At the end of the experiment, the probability is still remarkably high, approximately 0.15. For *Hybrid*-strategy, the initial probability is about 0.04 and the final is less than 0.01.

In the collection of Database 4, adaptation strategy *Add* was applied. This can be considered as a genuine on-line test of that adaptation strategy. To evaluate the effects of the adaptation, the characters were also recognized off-line with a nonadaptive system. As these results were obtained with all the character classes, not only with digits and lower case letters, they cannot be directly compared with the previous results.

On the basis of these experiments, the following observations were done: 1) A low final error rate for the adaptive experiment does not necessarily follow from a relatively good total error rate for the nonadaptive experiment. The initial accuracy of the recognizer is determined by the prototype set. It is high for those writers whose writing style is covered by the prototype set. On the other hand, the benefits of the adaptation, and thus the final accuracy, depend on the consistency and nature of the writing style. On the average, adaptation was able to drop the error rate from 23% down to 4.0%. 2) The initial rate of learning is high if the initial accuracy is low. 3) The increase in the size of the prototype set is more prominent if the final error rate is high. Due to the adaptation, the average final number of prototypes was 1.8-fold compared to the initial number. 4) The writers cannot change their writing styles so that the characters would more closely resemble the initial writer-independent prototypes and thus be better recognized. On the contrary, in the nonadaptive simulation, the error percentage for the last 200 characters was on the average slightly higher than the total error rate for all the characters.

The errors which occurred during the true collection of Database 4 and its nonadaptive simulation were analyzed and it was found out that the relative proportions of mis-

Figure 1. Evolution of the error rate in the simulated collections of Database 4. Separate prototype sets have been used for digits and lower and upper case letters. The lower and upper plot are obtained with the adaptative and nonadaptive systems, respectively.

recognitions in the three character groups (digits, upper and lower case letters) were roughly same for the two experiments. Therefore, the initial prototypes of the groups can be considered to be equally representative. In both experiments, the most probable type of error was a confusion between the lower and upper case versions of same letter. In addition, a letter was quite often mixed up with a digit, and vice versa. In the adaptive experiment, approximately 45% of all the errors were of the former and nearly 20% of the latter type.

In the last experiment, the collection of the Database 4 was simulated with the same adaptive and nonadaptive systems as in the previous experiment but the group of the input character was assumed to be known in advance. The results on the experiment are shown in Figure 1. It is evident that the use of separate prototype sets for the character groups is beneficial: the total error rate of the nonadaptive system drops from 23% to 9.8% and the final error rate of the adaptive system drops from 4.0% to 0.63%. In addition, the final number of prototypes is significantly lower – only 1.5-fold compared to the initial number. With the adaptive system, the total error rates for digits, lower and upper case letters were 0.42%, 2.93%, and 1.95%, respectively. More detailed information on all the experiments can be found in [9].

7. Conclusions

In this work, adaptation methods for handwriting recognizer based on elastic matching have been studied. First, a writer-independent classifier was constructed by choosing the combination of preprocessing and normalization methods and the dissimilarity measure most suitable for a set of writers. The major conclusion made in this phase of the work was that the implicit dynamic information which is contained in data points sampled with a constant frequency is beneficial. The dissimilarity measure and normalization method most sensitive to the data point distribution, namely *PP*-distance and *MassCenter*-normalization, were found to be better than those insensitive to the point density.

The main goal of the research was to study the adaptation of the system to the individual user. For this, several prototype set adaptation strategies were used. After adaptation, a recognition accuracy high enough to be acceptable for real-world applications was attained for most of the writers. Due to the adaptation, the recognition error rate for digits and upper and lower case letters was reduced from 23 % down to 4%. In addition, if separate prototype sets were maintained for the three character groups, the final error rate was less than 1%. It should be noticed, however, that the data used in the experiments is collected from a fairly limited set of writers and the recognition results varied considerably between the writers.

References

[1] I. Guyon and C. Warwick. Joint EC-US survey of the state-of-the-art in human language technology. http://www.cse.ogi.edu/CSLU/HLTsurvey.htm, 1996.

[2] P. E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(3):515–516, May 1968.

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

[4] J. Laaksonen, J. Hurri, E. Oja, and J. Kangas. Comparison of adaptive strategies for on-line character recognition. In *Proceedings of International Conference on Artificial Neural Networks*, 1998.

[5] J. Laaksonen, V. Vuori, E. Oja, and J. Kangas. *Advances in Handwriting Recognition*, chapter Adaptation of prototype sets in on-line recognition of isolated handwritten Latin characters, pages 489–497. World Scientific Publishing, 1999.

[6] U. Neisser and P. Weene. A note on human recognition of hand-printed characters. *Information and Control*, (3):191–196, 1960.

[7] M. Parizeau and R. Plamondon. Machine vs humans in a cursive script reading experiment without linguistic knowledge. In *Proceedings of International Conference on Pattern Recognition*, volume 2, pages 93–98, 1994.

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

[9] V. Vuori. Adaptation in on-line recognition of handwriting. Master's thesis, Helsinki University of Technology, 1999.