Testing the LAESA Algorithm in Handwritten Character Recognition Tasks *

Eva Gómez, Luisa Micó and José Oncina
Departamento de Tecnología informática y computación
Universidad de Alicante
E-03071 Alicante (Spain)
e-mail: {eva,mico,joncina}@dtic.ua.es

April 4, 1996

Abstract

Recently, the LAESA (Linear Approximating Eliminating Search Algorithm) was proposed for finding nearest neighbour in metric spaces. In a large series of synthetic experiments this algorithm showed to compute in average only a constant number of distances. In this work the algorithm is tested on some handwritten character recognition tasks. A Syntactical Pattern Recognition approximation is chosen due to its good performance and the large temporal cost in computing each distance. The LAESA showed to reduce the number of distance computation significantly.

1 Introduction

One of the more widely used technique in pattern recognition tasks is the the nearest neighbour technique. This technique is based in the definition of a similarity measure (distance) between patterns and the classification is performed comparing each test pattern with a large enough set of training patterns. Obviously, the error rate will decrease as the similarity measure is more accurate as the training set is larger. Unfortunately, the bests similarity measures are usually very time consuming. This, with the need of very large training sets, makes that the trivial nearest neighbour algorithm impractical.

Since the firsts times of the Pattern Recognition history, a large set of fasts algorithms were proposed [FN, 75] [FBF, 77] [KM, 83] [RP, 91] (see [Das, 91] fo a large survey). The aim of these algorithms was avoiding to calculate all the distances. Recently, the LAESA algorithm was proposed [MO, 92] [MOV, 92] [MOV, 94]. This algorithm is able of finding the nearest neighbour in an amount of time (in average) independent of the size of the training set and without making any assumption on the structure of the patterns nor the distance. It has also a linear space complexity unlike the AESA algorithm that has similar characteristics but has a cuadratic spatial complexity [Vid, 86] [Vid, 94] [VL, 88] [VRCB, 88] (see [FLL, 93] for a theoretical justification of this characteristics)

*Work partially supported under grant TIC93-0633-C02-02 from the Spanish CICYT

. Unfortunately, all these fast algorithms degrade their characteristics as the dimensionality of the representation space increase. In particular, the AESA and LAESA algorithms has shown that the constant that bounds the average execution time depends exponentially of the dimensionality.

Nevertheless, even on such cases, the use of these algorithms can be useful if the reduction of the distance computations is important comparing with the brute force algorithm.

In this work the LAESA is applied to a real life task, the recognition of handwritten characters. We have chosen a syntactical approximation, the patterns are represented by the contour chains of the characters. The similarity measure is the edit distance between the strings that represents the patterns, i.e. the minimum number of deletions, insertions and substitutions to transform a string into the other. This election is because of the small error rate obtained. As this distance is very expensive the use of the LAESA is justified. However, the dimensionality of the space is very high and then the asimthotic average constant time behaviour can not be observed.

On the experiments, different ways of building the pattern string are chosen and in all of them the LAESA proves to reduce significantly the number of distance computations.

2 The LAESA Algorithm

The algorithm requires a preprocessing procedure where, in linear time, a subset of patterns of the training set is selected. This prototypes are called the *base prototypes*. Also, the distances between base prototypes and the remaining patterns of the trainig set are computed and stored in a table. The performance of the algorithm depends on the number of selected base prototypes. If this number is too small the algorithm will make a lot of distance computations, if it is very high the overhead becomes unacceptable.

The search algorithm repeatedly performs five basic steps until all prototypes have been eliminated: distance computing, updating the nearest neighbour, updating the lower bound, approximating and eliminating. In fact the LAESA is no an algorithm but a family of them. In the approximating and eliminating steps, it make use of two generic functions that can be selected depending on the task. These functions are called *choice* and *condition*, and they allow different strategies of base prototypes management. In this work, the *choice* function forces the selection first of base prototypes, and the *condition* function rejects the elimination of base brototypes while they are contributing to the elimination[MOV, 92].

3 Feature Extraction

In this section is described the way that is used to obtain the pattern representation from the images of the characters.

The procedure uses eight direction vectors and describes the external contour of the character. It begins searching the first black pixel scanning the image form the top. Next it follows the border of the character until it returns to the

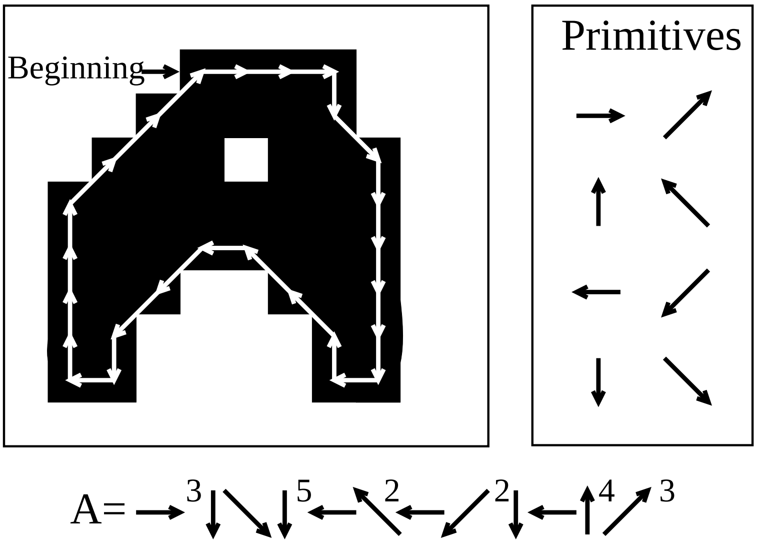

Figure 1: Example of chain–coding extraction

beginning pixel. During this traversal the algorithm builds a string depending on the direction that it follows to find the next pixel of the border (fig 1).

In order to reduce the variability of the direction vectors a smoothing algorithm is used. In this algorithm the smoothed strings are builded by substituting each element of the string by the average direction vector in a window around the element.

3.1 The Edit–distance

The edit–distance is defined as the minimum cost transformations that can be done to transform a string into an other. These transformations can be deletions, insertions or substitutions of symbols of the string. Recursively this distance can be defined as follow:

$$d(\lambda, \lambda) = 0$$

$$d(xa, yb) = min \begin{cases} d(x, yb) + W_d & \\ d(xa, y) + W_i & \\ d(x, y) + W_s & \text{if} \quad a \neq b \\ d(x, y) & \text{if} \quad a = b \end{cases}$$

where W_d, W_i and W_s are respectively, the cost of making a deletion, insertion or substitution transformation and λ is the empty string.

Of course, this distance can be computed in a time of $O(|x|, |y|)$ using a programming dynamic technique. Moreover, in real life experiments, this complexity can be reduced to $O(min(|x|, |y|))$ using a window around the best partial solution.

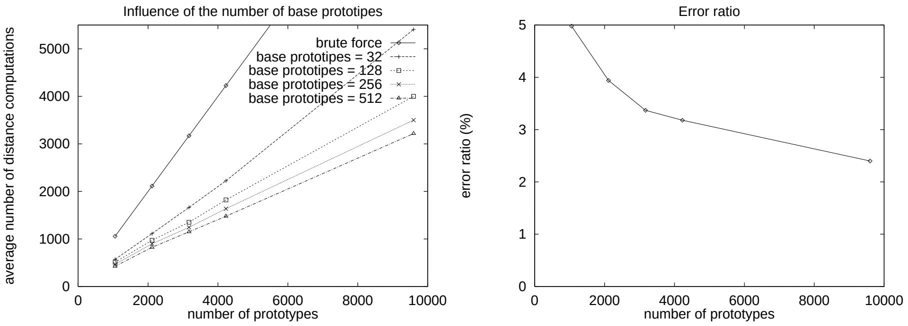

Figure 2: average number of distance computations and error rate for different number of base prototypes using an increasing training sample

4 Experiments

In order to demonstrate the performance of the LAESA in handwritten character recognition, a database of 100 writers was used. Each character is represented by an image of 128×128 pixels. On the experiments only the subset of the digits was used. For each writer there are around 100 images, around 10 of each digit.

The increasing size training samples for the experiments were build taking the writers in groups of 10 and using as training sample the needed number of groups. Each experiment was repeated as many times as necessary in order to have as training set all the possible combinations of groups. In the figures appear the average of all the experiments. The weights for the edit distance was $W_d = 1$, $W_i = 1$ and $W_s = 1$.

The first experiment was using the chain–code without smoothing. The set of experiments was repeated four times in order to show the behaviour of the LAESA using an increasing number of base prototypes. The results are represented in fig. 2 together with the brute force results. It can be seen that the number of distances computed is around one half using 32 base prototypes and this ratio decreases as the number of base prototypes increase. —— is represented the behaviour of the error rate. Obviously, the error rate falls as the train sample increases.

In order to show the behaviour of the LAESA varying the dimensionality of the search space a different set of experiments where carried out reducing the size of the images before extracting the chain code. The three experiments was for images of 128×128 (the original), 64×64 and 32×32. The average length of the strings was reduced on the same rate (160, 77 and 36 respectively). The experiments were made using 128 base prototypes. As in the previous case the result for the brute force is also presented. As it was supposed the LAESA reduces significantly the number of distance computations comparing with the brute force. This reduction increases as the dimensionality decreases. Obviously, as it can be see in fig. 3 more the images are reduced bigger is the error rate.

It can be thought that the only way to reduce the dimensionality of the space is reducing the length of the string. But the dimensionality can decrease also reducing the variability of symbols inside the string. This can be achieved eliminating the (quasi)random fluctuations of the directions due to the noise.

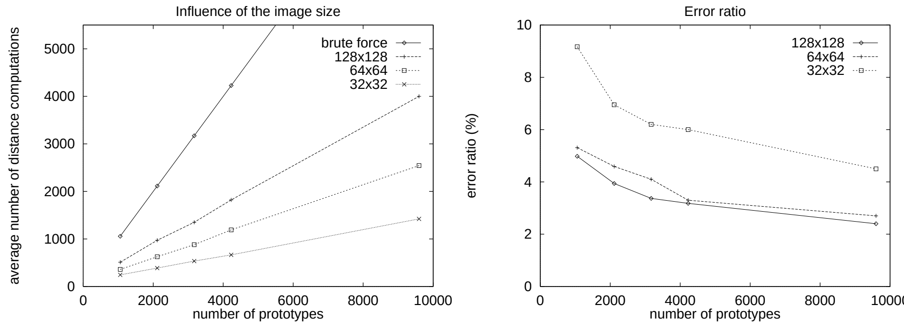

Figure 3: average number of distance computations and error rate for different image size using an increasing training sample and 128 base prototypes

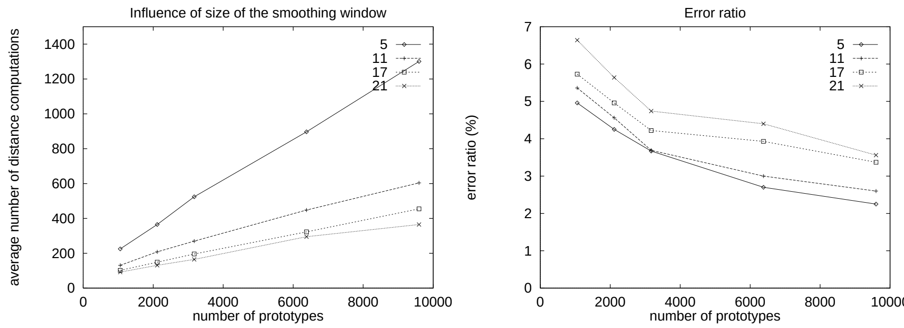

Figure 4: average number of distance computations and error rate for different smoothing windows using an increasing training sample and 64 base prototypes

This can be made averaging in a window around each component. As was to be expected, using the smoothing algorithm (fig. 4) very significant reductions of distance computations are obtained increasing the window size. In this figure the distance computatios of the brute force algorithm. However, error rate increases as the window size decreases.

5 Conclusions

From the previous experiments it is clear the LAESA presents an interesting behaviour even when we face high dimensionality problems. Moreover,

References

[Das, 91] *Dasarathy, B.Nearest Neighbour(NN) norms: NN Pattern Classification Techniques*

IEEE Computer Society Press, 1991.

[FBF, 77] *Friedman, J., Bentley, J., and Finkel, R.A. An Algorithm for Finding Best Matches in Logarithmic Expected Time*

ACM Transactions on Mathematical Software, Vol. 3, N 3, pp. 209-226, September 1977.

[FLL, 93] *Faragó, A., Linder, T., and Lugosi, G.Fast Nearest-Neighbor Search in Dissimilarity Spaces*

IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 15, N 9, pp. 957-962, 1993.

[FN, 75] *Fukunaga, K., and Narendra, M.A Branch and Bound Algorithm for Computing k-Nearest Neighbors*

IEEE Transactions on Computers, pp. 750-753, July 1975.

[KM, 83] *Kalantari, I., and McDonald, G.A Data Structure and an Algorithm for the Nearest Point Problem*

IEEE Transactions on Software Engineering, Vol. SE-9, N 5, pp. 631-634, September 1983.

[MO, 92] *Micó, L. y Oncina, J.Nuevos criterios para la selección en una reciente versión con preprocesamiento lineal del algoritmo AESA*

Actas del V Simposium Nacional de Reconocimiento de Formas y Análisis de Imágenes, pp. 261-268, Septiembre 1992.

[MOV, 92] *Micó, L., Oncina, J., and Vidal, E.An algorithm for finding nearest neighbours in constant average time with a linear space complexity*

Int. Conference on Pattern Recognition (ICPR), Le Hague, September 1992.

[MOV, 94] *Micó, L., Oncina, J., and Vidal, E.A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing-time and memory requirements*

Pattern Recognition Letters, Vol. 15, N 1, pp. 9-17, January 1994.

[RP, 91] *Ramasubramanian, V., and Paliwal, K.Fast Algorithms for Nearest-Neighbour Search and Application to Vector Quantization*

PhD dissertation, University of Bombay, 1991.

[Sha, 77] *Shapiro, M.The Choice of Reference Points in Best-Match File Searching*

Artificial Intelligence/Language Processing., Vol. 20, pp. 339-343, 1977.

[Vid, 86] *Vidal, E.An algorithm for finding nearest neighbours in (approximately) constant average time complexity*

Pattern Recognition Letters, Vol. 4, pp. 145-157, 1986.

[Vid, 94] *Vidal, E.New formulation and improvements of the Nearest-Neigbour Approximating and Eliminating Search Algorithm (AESA)*

Pattern Recognition Letters, Vol. 15, N 1, pp. 1-7, January 1994.

[VL, 88] *Vidal, E., and Lloret, M.J.Fast Speaker Independent DTW Recognition of Isolated Words Using a Metric-Space Search Algorithm (AESA)*

Speech Communication, Vol. 7, N 4, pp. 417-422, 1988.

[VRCB, 88] *Vidal, E., Rulot, H., Casacuberta, F.,* and *Benedí, J.On the use of a metric-space search algorithm (AESA) for fast DTW-bassed recognition of isolated words*

IEEE Transactions on Acoustics, Speech, and Signal Processing., Vol. 36, pp. 651-660, 1988.