Parallelizing a new environment based clustering method

Laura Lanzarini[1]
e-mail: laural@info.unlp.edu.ar

Armando De Giusti[2]
e-mail: degiusti@info.unlp.edu.ar

LIDI (Laboratory of Research and Development in Computer Science)[3]

ABSTRACT

There exists a wide range of problems which requires the automatic classification of a data set. In this sense, clustering techniques have been applied, since they are characterized by forming classes or groups using a predefined similarity measure.

The present article presents algorithm architecture and structure for paralleling clustering algorithm EBC (environment based clustering) which, deferring from usual solutions, processes input patterns in order to establish the similarity measure to be used.

Results obtained are analyzed over images of liver tissues with a maximum range of 256 colors, studying algorithm dependence on image resolutions and the number of different patterns in them. Then, critical points of the sequential algorithm are optimized over a PC net architecture.

Finally, the extension of the results obtained are discussed, as well as the solution presented for the case of high resolution images, in which the number of different patterns is of higher order (between 3000 and 5000).

Keywords: Parallel Algorithms, Clustering Techniques, Image Segmentation, Classification

1. INTRODUCTION

Clustering techniques, as its name suggests, are characterized by grouping input objects using same similarity measures. The result of this process is class or group formation [9].

The elements to be grouped are represented through their respective features vectors, and it is assumed that the ones which belong to a same class present close values for a given similarity measure.[7],[8].

When a classification of the patterns is meant to be performed without supervision, clustering techniques become the appropriate solution. This type of techniques is applicable in various areas such as medical diagnosis ([5], [6]), radar, video processing and weather prediction.

In general, all the applications present a high level of complexity as regards processing time. Also, in some of them, real time responses or short termed responses are required, all of which shows that parallel algorithm is highly justified [3].

In the present paper, a clustering method used in a monoprocessor scheme [1] [4] is presented and its time components are analyzed in detail [3]. From this analysis, a possible multiprocessor architecture (based on two unidimensional arrangements of homogenous processors, intercommunicated with respective resident master processes in

[1] Full-Time Co-Chair Professor, Fac. of Computer Sciences, UNLP.
[2] Director of the LIDI. Principal Researcher CONICET. Full-Time Chair Prof., Fac. of Computer Sciences, UNLP.
[3] LIDI (Laboratory of Research and Development in Computer Science). Faculty of Computer Science, University of La Plata (1900), Buenos Aires, Argentina. Tel / Fax: 54 - 221 - 422 7707. E-mail: lidi @ info.unlp.edu.ar

other two processors) is discussed as well as the attainable speed-up in function of the number of processors and image complexity in the treatment (resolution, color palette).[10], [11].

Finally, some of the difficulties in the implementation of parallel architecture are analyzed and solutions based on the multiprocessor architecture with distributed share memory (type SGI 2000) are discussed.

2. EBC (Environment based Clustering)

2.1. New technique suggested

In the particular case of clustering techniques, the algorithms can be separated into two classes: those that use an only representative or descriptor for each class, and those that use several descriptors.

The former renders good results when applied to problems in which the classes present very little dispersion since what belongs to the surrounding hypersphere can only be recognized with just one representative.

Variations of these methods use hypercubes and produce a similar effect [15]. In particular, the methods proposed by Simpson [13], [14] present alternatives to classify data in a few runs, but their result depends on the order of the input patterns.

On the other hand, those using several representatives, such as [8], solve the class dispersion problem, though they require the setting of initial similarity parameters which depend on the problem, allowing to establish a relationship between those representatives.

EBC is a new clustering method belonging to the second group with the objective of improving the previous suggestions in order to achieve an automatic classification that does not require initial parameters nor is dependent on the order of the analysis of the data.

Step 1: Analysis of the environment of each pattern to classify.

The process starts with an analysis of the input data or patterns.

Since the purpose is to relate them, their corresponding environments will be analyzed (see section 3). This analysis will allow to obtain two values P_i for each pattern:

1. *DistMAX:* every pattern P_j , with $j \neq i$, that is within a distance shorter than that value, will be considered to be similar to P_i and will therefore have a tendency to belong to the same class.
2. *DistMIN*: if the distance between P_i and P_j is shorter than this value, P_i and P_j will be considered to be very similar, and therefore it will be enough to use only one descriptor to represent both.

Step 2: Initial classes.

Initially, there will not be any class assigned.

Class formation:

From this point on, the next iterative process will allow to relate the patterns by creating the corresponding classes:

Step 3: Distribution of the patterns among the existing classes.

Let $C = \{C_1, ..., C_k\}$ be the set of classes created so far.

Let $P = \{ P_1, ..., P_n \}$ be the set of patterns to classify.

Each class C_l will be represented by a set of prototypes:

$Prot_l = \{Prot_{l1}, ..., Prot_{ls}\}$; with l=1..k

Note that the amount of prototypes varies with the class.

Each pattern not yet classified will analyze its distance with the prototypes of each class in the following way:

- If dist($Prot_{ji}$, P_t) < $DistMAX_{Class\ j}$, the pattern P_t will belong to class j, where

  $DistMAX_{Class\ j}$,= average($DistMAX_{Protji}$)

with i=1..s, s = number of prototypes in class j

- If P_t turns out to belong to several classes, these will be all grouped into only one class.
- If P_t turns out to belong to an only class, it will be necessary to analyze if there is some new information to contribute to the class; that is, if it can be a new prototype.

  To do so it must be true that dist($Prot_{ji}$, P_t) > $DistMIN_{Class\ j}$
- If, on the contrary, P_t does not belong to any of the existing classes, a new class with this P_t as the only prototype will be created.

The values $DistMAX_{Class\ j}$ and $DistMIN_{Class\ j}$ will be obtained from the average of the values of the prototypes of $Class_j$.

Step 4: Deletion of small classes.

All classes with less patterns than the 0.5% of the total of patterns to classify will be deleted.

Repeat steps 3 and 4 until 90% of the input patterns are classified, or until the number of patterns per class is constant.

Step 5: Joining of near neighbors.

Each pattern and its closest neighbor will be analyzed. If they belong to different classes but the distance between them is shorter than the DistMAX of any of the two classes, the classes will be grouped.

2.2. Initial analysis of input patterns

a) Representation of input patterns

It is important to bear in mind that, depending on the characterization used and the problem involved, patterns can be repeated; therefore, not only pattern characteristics but also their cardinality will be taken into account for each pattern.

b) Distance estimation

This is one of the most important steps in order to achieve a correct result.

For each pattern, two distance values are required:

Distance to its nearest neighbor:

For pattern P_i, it will be denoted as $DistMIN_{Pi}$.

$$DistMIN_{Pi} = \min(dist(P_i, P_j)) \quad \text{with } j \neq i$$

Distance between patterns of a same class

P_i will accept as members of its class those patterns that fulfill the following condition

$$dist(P_i, P_j) \leq DistMAX_{Pi} \qquad \text{with } j \neq i$$

In order to determine this threshold value, the three shortest distances will be considered, and for each of them the number of patterns (multiplied by their cardinality) will be registered.

Be *TotPatrones* the sum of the patterns found at these three distances (see Fig. 1).

$DistMAX_{Pi}$ will be the distance that allows to include 50% of *TotPatrones*.

Thus, $DistMAX_{Pi}$ will be, for P_i, a measure of proximity.

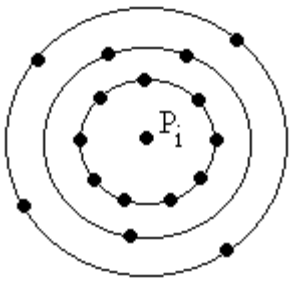

Fig.1 : P_i will have 9 neighbors at a distance D1, 3 at a distance D2 and 4 at a distance D3, with D1<D2<D3.
TotPatrones = 16

In the example of Fig. 1, $DistMAX_{Pi}$ = D1 and every pattern at a distance which is shorter than or equal to D1, will belong, for P_i, to its class.
By taking three distances greater than zero, the central pattern is forced to have many representatives in order to be isolated, otherwise it will have at least one neighbor to which it will have to be grouped.

2.3. Implementation aspects

As it can be seen, classes are formed around prototypes. Even though it is not necessary to have the initial classes, the classification process can be sped up by chunking the characteristics space into equal sectors and by selecting a pattern from each of them.

Then each pattern will be considered the first prototype of a new class.

On the other hand, each prototype added to a class not only contributes to it with its characteristics but also with its similarity values DistMAX and DistMIN.

The admission of a pattern as a member or new prototype of class j will be given by $DistMAX_{Class\ j}$ and $DistMIN_{Class\ j}$ respectively.

Each of them is obtained as an average of the distance values of the prototypes forming it so far. In order to make the method independent from pattern insertion order, step 5 is applied.

3. PARALLEL EBC

3.1. General Description

The obtaining of favorable results on the part of algorithm EBC is based on performing a correct processing of input patterns. As a result of this stage, necessary similarity criteria are obtained in order to establish the classes and their descriptors.

Input patterns correspond to the different colors which appear in the image. This means that the size of the pattern space will be determined by the quality of the elements containing image color palettes.

EBC method is the conclusion of a project which aims at the recognition and classification of the elements present in a liver tissue sample [2]. In that problem, images were captured with a resolution of 8 bits per pixel in order to assure that the size of the patterns space had less than 256 elements.

The paralleling of the algorithm allows its application into images of 24 bits per pixel, solving response time restrictions.

Next, an analysis of a sequential algorithm is described:

The first part of the preprocessing of the input image can be divided in two stages:

1) Transformation of the input image, with a size of N x N in a set of P sized patterns. For each pattern its color will be obtained as well as its cardinality in the image and the significant neighbors.
2) Obtaining of the necessary simulator criteria in order to compose classes.

Once the patterns which compose the image are analyzed, the classes begin to compose themselves, using, firstly, the most representative element. This leads to the necessity of ordering them according to its cardinality and quantity of neighbors.

This is an iterative process which is carried out until a 90% of the patterns are grouped.

3.2. Sequential algorithm scheme

Initialization
Image preprocessing

- *Go from N xN pixel image to a set of P patterns { Time = **Tconv** }*
- *Determine for each pattern the judgment of acceptance { Time = **Tsim** }*
- *Order the patterns to be inserted in the RN { Time = **Tord** }*

*Group the patterns be means of an iterative process that for each pixel of the image: { Time = **Tsec** }*

- *Search the class they belong to.*
- *If it belongs to only one class, insert it; if it does not belong to anyone, then create a new class with this pattern and if it belongs to several classes, then unite them.*
- *Analyze the size of the formed classes and delete those which do not correspond to the expected ranges.*

Transmit/ present the results obtained.

The total processing time is:

TTotal = Tconv + Tsim + Tord + Tsec

3.3. Effectiveness of the clustering process with sequential algorithm

To test the effectiveness of EBC method, 194 images corresponding to different liver tissue samples have been grouped.

It is worth mentioning the fact that, due to the scale used, from one single liver tissue sample it is possible to obtain 200 images approximately. To assure the sample representativity, it is advisable to capture the images in the manner of a greek guard.

Each image has a resolution of 640 x 480 pixels to 256 colors. Also, it is expected that, if the samples have been taken under the same conditions (light, colors, etc.), they will have palettes of similar colors.

In all the cases it was possible to prove not only that EBC method converged but also that the algorithm was able to carry out the clustering using a unique iteration.

If image resolution increases (that is, if images with larger quantity of colors are used), it is evident that the number of iterations varies from between 2 and 3, depending on the spreading of the colors presented. In all the cases, the method converges.

4. TIME COMPONENTS OF SEQUENTIAL ALGORITHM

The cost analysis, in the sequential algorithm code cycles, was divided as follows:

1. Conversion time of NxN image in a set of P pattern.
   Tconv = $k_1 + 10 * N + N * N * (k_2 * P + k_3)$
2. Time to calculate similarity criteria.
   Tsim = $k4 * P^2 + k5*P$
3. Necessary time to order the pattern according to its cardinality and quantity of neighbors.
   Tord = k6 * P * Lg(P)
4. Time of each iteration of sequential algorithm.
   Tsec = $k7 * P^2 + k8 * P + k9$

Where k_i stands for a constant value with i= l..9.

In the following section, the previous equations are represented graphically. The values of the constants have been fixed taking into account the processing of liver tissues samples. This implies that Tsec is evaluated only once.

Figure 2 shows the graphic of the correspondent equation to Tconv and figure 3 shows remaining equations. All of them are expressed in millions of cycles.

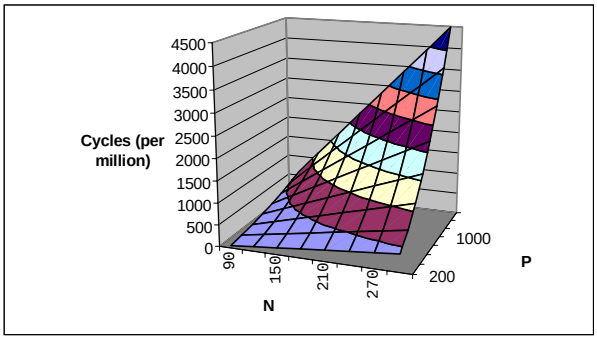

Fig. 2

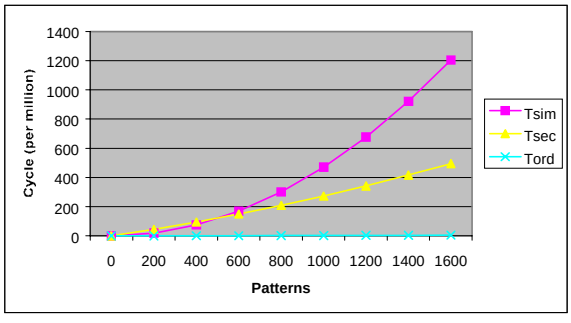

Fig. 3

It is evident that the principal time component is found in Tconv, NxNxP order. Tsim and Tsec evaluations are of P^2 order. For the examined images of liver tissues, in which the number of patterns (colors) is lower than 256, Tsec prevails as a second time factor.

5. SUGGESTED PARALLEL ARCHITECTURE

5.1. Architecture Scheme

The suggested parallel architecture is a pipeline multiprocessor of 4 levels, such as Figure 4 shows:

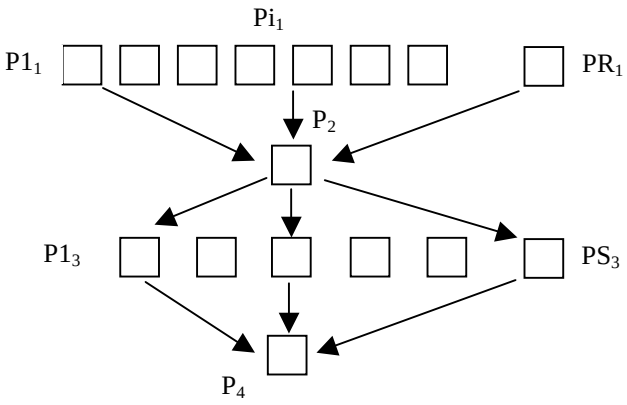

Fig. 4

Processors Pi_1 correspond to the first pipe level. They receive a sector of the image to perform the processing which corresponds specifically to Tconv time. The division of the image in R sectors allows distributing the initial processing of the patterns in R processors, with a minimum interaction among them.

P_2 processor is in charge of ordering data of R sectors related to initial S patterns, selecting more frequent S1 patterns and defining the distances which will determine the judgement of acceptance for each pattern.

In the third level, P2 distributes data of each pattern (color characteristics, center and influence zone) and, by processing all the image, S1 processors (Pi_3) refine the initial classification, selecting pixels for each pattern and carrying out the necessary class fusions and divisions which are necessary.

This processing principally solves the Tsec component of the algorithm. Finally, P4 receives level 3 data and carries out the ordering of definitive patterns and data output.

In the experimental analysis, the possibility (NxN pixels images) of having N, N/2, N/8 and $Lg_2(N)$ processors in the first level, P/4, P/8 and P/16 processors in the third level (in which P is the number of image different colors) has been considered. Also, execution time has been studied; in particular Tconv and Tsec components of the sequential algorithm.

5.2. Attainable Speed-Up.

Attainable Speed-up is shown in Figure 4 as function of number of the processors in the first level, and having the number of processors of the third level as parameter, for 256 x 256 pixel images with 256 colors.

In figure 5, the experience is repeated with 640 x 480 pixel images with 256 colors, considering N as the row number of the image.

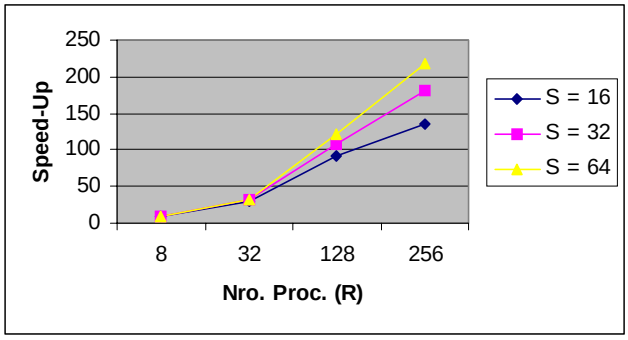

Fig. 4

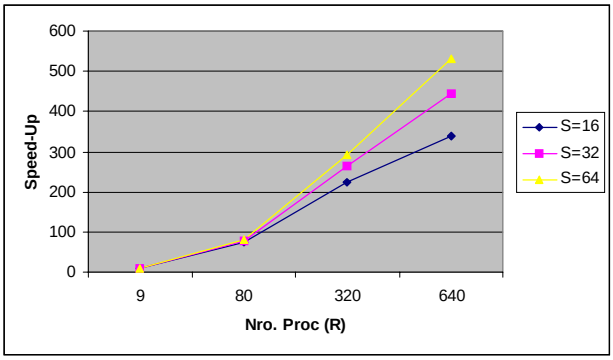

Fig. 5

5.3. Performance consideration in the treatment of image sequences

The performance of the suggested parallel architecture is a really important aspect to take into consideration. In the treatment of a single image, performance is low since the levels of the processors 1 and 3 cannot work simultaneously, all of which yields a significant number of idle processors. However, the real problem demands a treatment of the image sequences of the same liver tissue sample, so the performance increases notoriously since all of the pipe processors are active during almost all the process.

Another important aspect to consider is the model of communication. Although an algorithm around a shared memory PC clusters (which avoids repeated transmission of the image) is suggested, communications with P2 processor and with P4 are an important restraint for the attainable speed-up. Nowadays, the implementation of an algorithm on a computer with SGI Origin 2000 type with shared distributed memory is being studied, all of which will allow to optimize the cost of communication times.

6. CONCLUSIONS

Experimental results on different kind of images confirm the effectiveness of using EBC method for class based clustering, but the processing time increases with NxNxP where NxN is image resolution and P is the number of different patterns.

By decomposing the sequential algorithm, we verified that the transformation of the pixel space in pattern space Tpp is the main component of processing time (O (N2 P)). The second factor (O p2) is similarity analysis Tsa between pixels in different patterns (O p2). A theoretic relationship between Tpp, Tsa and image resolution was set.

By using these results we studied a parallel architecture, based on homogeneous processors and distributed memory, in order to solve the algorithm. Speed-Up was studied and some drawbacks for real implementation were discussed.

At the present time, we are studying the scalability of the parallel solution and the possible migration of the algorithm to a multiprocessor architecture with shared distributed memory, thus reducing inter-processor communication times.

REFERENCES

[1] Lanzarini, De Giusti, “Environment based Clustering: A new approach”. First International Workshop on Image and Signal Processing and Analysis. IEEE CAS & ASSP Croatia, 2000, pp.75-80.

[2] Lanzarini, “Reconocimiento de los elementos de muestras histológicas”. Phd Thesis School of Computer Science UNLP, Argentina, 2001.

[3] Lanzarini L., “Eficiencia del método EBC”. LIDI Technical Report, May 2000.

[4] Lanzarini L., De Giusti A. ,"Una nueva Red Neuronal para Clustering y Segmentación basada en el Entorno", Proceedings of VI Argentine Congress of Computer Science.Ushuaia, Argentina, Oct. 2000. pp. 234-239.

[5] Lanzarini, "Pattern Recognition in Medical Images using Neural Networks”. Journal of Computer Science & Technology, Vol. 1, No. 4, 2001, pp. 50-53.

[6] Lanzarini, Badrán De Giusti , "An Application of Neural Networks for Elements Classification in a Blood Sample". III International Congress on Information Engineering, 1997, pp. 301-308.

[7] Torbjorn Eltoft, “A New Neural Network for cluster-detection-and-labeling”, IEEE

Transactions on Neural Networks, Vol.9, No. 5, 1998, pp.1021 – 1035.

[8] Newton, Pemmaraju and Mitra, "Adaptive Fuzzy Leader Clustering of Complex Data Sets in Pattern Recognition", IEEE Transactions on Neural Networks, Vol.3, No. 5, 1998, pp.794 – 800.

[9] Maravall Gomez - Allende, "Reconocimiento de Formas y Visión Artificial".Addisson-Wesley Iberoamericana, 1994.

[10] Akl S, "Parallel Computation. Models and Methods", Prentice-Hall, Inc., 1997.

[11] Brinch Hansen, P., "Studies in computational science: Parallel Programming Paradigms", Prentice-Hall, Inc., 995.

[12] www.setcip.gov.ar/config_clementina2.htm.

[13] Simpson Patrick. "Fuzzy Min-Max Neural Networks – Part 1: Clustering". *IEEE Transactions on Neural Networks*, Vol. 3, nr. 5 , 1991, pp 776-786.

[14] Simpson Patrick. "Fuzzy Min-Max Neural Networks – Part 2: Clustering". *IEEE Transactions on Fuzzy Systems*, Vol. 1, nr 1 , 1993, pp 32-45.

[15] Meneganti Massimo, Saviello F., Tagliaferri R. "Fuzzy Neural Networks for classification and detection of anomalies". *IEEE transactions on Neural Networks*. , Vol. 9, nr. 5, 1998, pp 843-860.