ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2010. No. 9(105)
ELEKTRONIKA IR ELEKTROTECHNIKA

SYSTEM ENGINEERING, COMPUTER TECHNOLOGY
T 120
SISTEMŲ INŽINERIJA, KOMPIUTERINĖS TECHNOLOGIJOS

Evaluation of the Performances of a Parallel Algorithm to Recognize the Patterns in Relation with the Sequential Variant

C. Pughineanu
Faculty of Electrical Engineering and Computer Science, "Ştefan cel Mare" University of Suceava,
Str. Universitatii nr. 13, 720229 Suceava, România, phone/fax: +40 230 524801, e-mail: secretariateed.usv.ro

Introduction

The rhythm of human biological degradation, due to increased obesity, high blood pressure and the main cardiovascular affections, to cancer extension, to the breathing diseases etc., is more and more alarming. A first aim of this paper is the use of some clinical tests [1], based on medical investigations, on patients to prevent the main diseases that threaten the human species.

The pattern recognition algorithms are very useful in analyzing the information contained by the tests. The great volume of the entering data (hundreds of recordings) collected from patients and used by these algorithms have an important part in grouping them according to their genetic heritage. A correct clustering of these inheritances cannot be achieved by the sequential algorithms, having a single processor with limited memory, but by the parallel algorithms using a great number of processors, each having its own memory.

The problem that appears in the clustering process is constituted by the grouping of a set of data non-labeled in a number of classes. In solving the clustering problems, an important element is represented by the similarity measure, the grouping of the patients is achieved due to the similarity among them [2–8].

The k-means algorithm, model proposed by MacQueen (1967), is considered the simplest algorithm of data clustering and may be used in grouping the patients due to the predispositions they are exposed to.

This paper aims the implementation of a k-means algorithm on a cluster formed by 28 identical nods, each nod having 2 Xeon quad core processors of 2.33 GHz. These nods are connected to one another by a gigabyte web.

Description of the entering data set

The entering data set has its own structure depending on clinic test to be achieved on a number of patients. The data offered by a patient (a line of the matrix) may be quantitative, qualitative or mixed. To eliminate the effect of the different measurement units of the considered characteristics, their standardizing is necessary, namely changing them into numerical data. This change of the clinical test data into a numerical form plays an important part in the way of obtaining the clustering process and leads to the possibility of using the algorithm of data clustering namely of the k-means algorithm.

The clinical test is represented by a matrix in each line corresponding to a patient and each column corresponds to the observations of the respective patient. To achieve the patient grouping process we need a similitude function [9–13].

Measures of the similarity between two patients

Unlike the classical classifications that have a well established aim and clear grouping criteria, the automatic classifications have as a unique criterion the similarity [2] between the patients that are to be grouped. There is a variety of methods that may be used to measure the similitude.

In data clustering the mostly used similarity is represented by the Euclidean distance given by the following formula [1]

$$d(x,y) = \|x - y\| = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\ , \qquad (1)$$

where $x = (x_1,...,x_n)$ and $y = (y_1,...,y_n)$ are two points given in the cartesian coordinates.

Another method to measure a similarity is represented by the cosine of the direction of two vectors $x = (x_1,...,x_n)$ and $y = (y_1,...,y_n)$ and is given by the following formula [2]

$$\cos\theta = \frac{(x,y)}{\|x\|\|y\|}, \qquad (2)$$

where (x,y) is the scalar product; $\|\cdot\|$ is the Euclidian norm, $x, y \in R^n$.

Practically, an automatic classification achieves an optimum patient clustering from the point of view of the similarity ratio between them.

Description of the clustering algorithm

As mentioned when testing this algorithm, we shall work with great sets of data. If the data set that contains n patients $(x_1,...,x_n)$, in which every patient is considered as a vector of observing the dimension d, then the k-means clustering presupposes the dividing of this into k non-overlapping and non-empty partitions $(k < n)$ $S = \{S_1, S_2, ..., S_k\}$ so that the similarity function minimizes [3]

$$E = \arg\min_{s} \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - c_i \right\|^2, \quad (3)$$

where c_i is the average of S_i.

If in this case we use the cosine similarity, the more the function approaches 1, the closer the two patients are.

The sequential algorithm of *n* patients in *k* classes is:

```
1. |S|/k members of the set S are randomly selected to form k-subsets
2. While E is not stable:
3. For each of the k subsets it is computed a means ci, 1≤i≤k.
4. Compute distance d(i,j), 1≤i≤k, 1≤j≤n of each vector so that d(i,j)=||xj-ci||
5. Each vector is assigned to its corresponding subset according to the closest distance.
```

The k-means serial algorithm has the execution time equal to $O(i_s kn)$ where *k* is the number of clusters and i_s is the number of iterations [4].

The classification procedure in the parallel case is practically progressed as follows: one starts from an arbitrary arrangement of the patients in a number of pre-established groups, after which the patients are transferred from one group to another to minimize the variation inside the groups and thus to maximize the variation among the groups. The number of transfers may be, likewise, specified by the researcher.

The procedure follows a simple and easy way to classify the entrance data set by a number of clusters (supposed *k* clusters) fixed apriorically. The basic idea is to define *k* weight centers, one for each cluster. These weight centers must intelligently be fixed as different locations lead to different results. Thus, the best choice is to fix them, as much as possible, farther one from another. The next step is to take each point along the entrance data set and to associate them to the closest weight centre. When there are no undecided points, the first step is complete, and an initial clustering is achieved. In this point one must recalculate *k*, new weight centers of the clusters resulted from the previous step. After we have these *k* new centroids, a new link was achieved between the same data set and the closest new weight centre. A nod was generated. As a result of this nod, we notice that the *k* weight centers change their location step by step when no more modifications are made. In other words, the centroids do no longer move.

Being given an initial set of *k* means, $c_1, c_2, ..., c_k$ that may be given at random or by an heuristic algorithm, the k-means algorithms has two steps:

The assignation step: assigns each observation to the cluster with the closest average

$$S_i = \left\{ x_j \mid \left\| x_j - c_i \right\| \le \left\| x_j - c_m \right\| \right\}, \quad (4)$$

for $m = 1,...,k$.

The adjustment step: the computing of new averages as being centroids of the observation cluster.

$$c_i = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j \,. \quad (5)$$

The parallel algorithm proposed to testing is:

```
Master Process
1. k equal subsets of the set S are formed randomly
2. Each subset is sent to each of the k slaves
3. The k resulting subsets from k slaves are received
```

```
Slave Process
1. A subset Si is received from master process
2. While E is not stable:
3. For each subset Si a means ci is computed
4. The mean ci is broadcasted to every other slave
5. Compute distance d(i,j), 1≤i≤k, 1≤j≤n/k of each vector so that d(i,j)=||xj-ci||
6. Each vector is assigned to its corresponding subset according to the closest distance
7. The k subsets computed in step 6 are broadcasted to every other slave
8. The new subset Si is formed by collecting vectors that belong to ci that were sent from other slaves in step 7
9. End_while
10. Send the subset Si to master process
```

Time complexity analysis

The parallel algorithm proposed has 4 communication phases and 3 calculation phases. We note with T_c the time complexity of communication and T_p be the time complexity of computation.

The 7 steps are described as follows:

Phase 1: the master process sends the k subsets equally to k slave. Thus

$$T_{c1} = k\left(T_s + \frac{n}{k}T_d\right). \quad (6)$$

Phase 2: after every slave process receives a subset from the master process, this calculates the c_i average of the subset S_i (one of the k). This phase lasts

$$T_{p1} = |S_i| = \frac{n}{k}. \quad (7)$$

Phase 3: every slave communicates its own average to all the other slaves. Thus

$$T_{c2} = T_s + T_d. \quad (8)$$

Phase 4: every slave calculates the Euclidian distance of each vector of the subset S_i and determines the new members of each subset depending on the distance. This phase lasts

$$T_{p2} = 2k\frac{n}{k} = 2n. \quad (9)$$

Phase 5: in this phase each form is transmitted to the corresponding subset

$$T_{c3} = T_s + \frac{n}{k}T_d. \quad (10)$$

Phase 6: every slave forms its new subset S_i. Thus

$$T_{p3} = \frac{n}{k}. \quad (11)$$

Phase 7: each slave forms the new subset S_i. Hence

$$T_{c4} = T_s + \frac{n}{k}T_d, \quad (12)$$

where T_s is the constant time necessary to send an blank message and T_d is the constant necessary for sending a datum (i.e. 4 bytes of integer).

If i_p is the number of iterations of the loop of step 2 then the total time complexity may be written as

$$T_t = T_{tc} + T_{tp}, \quad (13)$$

where T_{tc} is the total time complexity of communication and T_{tp} is the total time for calculation:

$$T_{tc} = \sum_{i=1}^{4} T_{ci} = O\left(n + i_p\frac{n}{k}\right), \quad (14)$$

$$T_{tp} = \sum_{i=1}^{3} T_{pi} = O\left(2i_p k\frac{n}{k}\right) = O\left(2i_p n\right), \quad (15)$$

$$T_t = T_{tc} + T_{tp} = O\left(2i_p n\right). \quad (16)$$

Taking as uniform the distribution of the n forms, every process slave needs $O(n/k)$. Therefore, the parallel k-means algorithm has total space complexity $O(n)$.

Experimental results

To test the proposed algorithm we used a cluster formed of 28 identical nods, each having 2 Xeon quad core processors of 2.33 Ghz. These nods are connected among them by a gigabyte web, the theoretical calculation power being of 2 Tflops. The k-means program is achieved using OpenMPI implementation of the Message Passing Interface standard (MPI).

As entrance data we used data sets from UCI Machine Learning Repository[15]. The algorithm was rolled on four entrance sets: Hepatitis containing 155 patients with 20 observations, Mammographi Mass containing 961 patients with 6 observations, Parkinsons containing 197 patients with 23 observations and Parkinsons Telemonitoring containing 5875 patients with 26 observations. The data were selected to differ both as number of patients and as number of observations for each patient separately. For every entrance set we grouped the patients into 10, 25, 50, 100 classes.

The reducing of the execution time is due to the way the application was achieved. The losses that are due to communication are minimum as this operation implies a small volume of information transferred among the nods.

If k is the number of processes then we may measure the performance of k-means algorithm parallel to the serial one using the speedup as follows

$$S_p = \frac{t_s}{t_p} = O\left(\frac{k}{2}\right). \quad (17)$$

Conclusions

In this article we presented a parallel algorithm for patient clustering. This algorithm was implemented and tested on the cluster. From the results of the tests one may observe that by doubling the number of nods on which the application is computed, the execution time decreases by a ratio very close to 50%.

The final result is influenced by the initial positions of the k centers of weight; the elimination of this drawback was achieved by considering more variants of the initial positions of the weight centers.

Even if the entrance data set is of hundreds of thousand recordings, the speedup increase obtained by the parallel computing is outstandingly superior to the sequential one.

Acknowledgements

I would like to thank Faculty of Electrical Engineering and Computer Science, "Ştefan cel Mare" University of Suceava, Romania for allowing me to use the computer facilities in my experiment.

References

1. **Pentiuc Şt.-Gh.** Aplicații ale recunoaşterii formelor în diagnosticul automat. – Bucureşti: Editura Tehnică, 1997.
2. **Kohonen T.** Self-organization and associative memory. - Springer-Verlag Berlin Heidelberg, New York, London, Paris, Tokio, 1988

3. **Cover T. M., Hart P. E.** Nearest Neighbor Pattern Classification // IEEE Trans. on Information Theory, 1967. – No. 1(13). – P. 21-26,
4. **Zamir O., Etzioni O.** Web Document Clustering: A Feasibility Demonstration // Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998. P. 46-54
5. **Asuncion A., Newman D. J.** UCI Machine Learning Repository Irvine. - CA: University of California, School of Information and Computer Science, 2007. [Online: http://www.ics.uci.edu/~mlearn/MLRepository.html].
6. **Ungurean I.** Job Scheduling Algorithm based on Dynamic Management of Resources Provided by Grid Computing Systems // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 7(103). – P. 57–60.
7. **Vahdat–Nejad H., Zamanifar K.,** A New Randomized Algorithm for Handling Scheduling Conflicts in Grids // Advances in Electrical and Computer Engineering, 2009. – Vol. 9. – No. 3. – P. 22–26.
8. **Alexa D., Goras T. C.; Sârbu A., Pletea I. V., Filote C., Ionescu F.** An Analysis of the Two–Quadrant Convertor with RNSIC // IET Power Electronics, 2008. – Vol. 1, No. 2. – P. 224–234.
9. **Jeļinskis J., Lauks G.** Detection of Trends of Internet Traffic using Sequential Patterns // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 5(93). – P. 3-6.
10. **Pentiuc S.G., Schipor O.A., Danubianu M., Schipor M.D., Tobolcea I.** Speech Therapy Programs for a Computer Aided Therapy System // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 7(103). – P. 87–90.
11. **Rață G., Rață M., Filote C., Strugaru C.** Theoretical and Experimental Aspects Concerning Fourier and Wavelet Analysis for Deforming Consumers in Power Network // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 1(97). – P. 62-66.
12. **Dambrauskas A., Rinkevičius V.** Algorithmic Methods of Variational Calculus // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 5(85). – P. 25–28.
13. **Ciufudean C., Larionescu A., Filote C.** Equivalent Structure for Nonlinear–Signal Cable Networks // Electronics and Electrical Engineering. – Kaunas: Technologija, 2009. – No. 5(93). – P. 65–68.

Received 2010 02 14

C. Pughineanu. Evaluation of the Performances of a Parallel Algorithm to Recognize the Patterns in Relation with the Sequential Variant // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 9(105). – P. 65–68.

To achieve an algorithm to efficiently recognize the patterns to return the best solution, we need an intensive processing of the entrance set data. In most cases there is a compromise between the taking over of the entrance set data and the algorithm execution time. The intensive processing of the entrance set data needs rather high calculation resources which cannot always be obtained from the ordinary calculation systems. In this paper we suggest the parallelization of an algorithm to recognize the patterns in scientific literature, its parallelization and the evaluation of this variant in relation to the sequential algorithm. This algorithm will be tested using a cluster formed of 28 nods, each nod having 2 quad core 2.33GHz processors. Bibl. 13 (in English; abstracts in English and Lithuanian).

C. Pughineanu. Lygiagrečiojo ir nuosekliojo algoritmų našumo įvertinimas vaizdų atpažinimo sistemose // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2010. – Nr. 9(105). – P. 65–68.

Nustatyti siekiamo atpažinti objekto pradiniai parametrai padeda padidinti algoritmo efektyvumą. Daugeliu atveju galima suderinti algoritmo vykdymo ir pradinių parametrų įvedimo laiką. Parametrams apskaičiuoti paprastai reikia didelių skaičiavimo sistemos išteklių, kuriuos ne visada galima užtikrinti. Apžvelgta mokslinė literatūra ir pasiūlytas algoritmas jungiantis lygiagretųjį ir nuoseklųjį algoritmus. Jis ir buvo patikrintas. Bibl. 13 (anglų kalba; santraukos anglų ir lietuvių k.).

DOI: 10.5755/j02.eie.9179