Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 42(2), 2015, Pages 365–373
ISSN: 1223-6934

Statistical considerations on the *k*-means algorithm

Dana Mihai and Mihai Mocanu

Abstract. Cluster analysis from a data mining point of view is an important method for knowledge discovery in large databases. Clustering has a wide range of applications in life sciences and over the years it has been used in many areas. The k-means is one of the most popular and simple clustering algorithm that keeps data in main memory. It is a well known algorithm for its efficiency in clustering large data sets and converges to acceptable results in different areas. The k-means algorithm with a large number of variables may be computationally faster than other algorithms. The current work presents the description of two algorithmic procedures involved in the implementation of k-means algorithm and also describes a statistical study on a known data set. The statistics obtained from the experiment, by detailed analysis, can be used to improve the future implementation techniques at the k-means algorithm to optimize data mining algorithms which are based on a model.

2010 Mathematics Subject Classification. Primary 60J05; Secondary 60J20.
Key words and phrases. k-means algorithm's, cluster analysis, data mining.

1. Introduction

Knowledge Discovery and Data Mining are rapidly growing fields in the area of Computer Science. They group together methods and techniques, which allow to analyse very large data sets to extract and discover previously unknown structures and relations out of huge heaps of unnecessary detailed information. Topics include: classification and clustering, cluster analysis, deviations analysis, trend associations, dependency modeling, sequential patterns, analysis of time series, classification by decision trees or neural networks.

Over the last two decades we have witnessed an explosive growth in the generation and collection of data. Information of all kinds needs to be filtered, prepared and classified so that it will be a valuable aid for decisions and strategies. We developed needs for new techniques and tools that can intelligently assist us in transforming row data into useful knowledge [1].

The ever-growing repository of data in almost all fields can contribute significantly towards future decision making provided appropriate knowledge-discovery mechanisms are applied for extracting hidden, but potentially useful information embedded in the data. A knowledge discovery system employs a wide class of machine-learning algorithms to explore the relationships among tuples, and characterize the nature of relationships that exist between them. Classification and clustering are two most commonly examples for knowledge-discovery techniques that are applied to extract knowledge. Classificatory analysis refers to a set of supervised learning algorithms which study pre-classified data sets in order to extract rules for classification. Cluster analysis and clustering techniques refer to unsupervised learning algorithms for data

Received April 28, 2015. Accepted June 2015.

analysis in which the aim is to partition a given set of data elements into homogenous groups called clusters [4].

In this paper we demonstrate how the popular k-means cluster algorithm can be combined with statistical-based assumptions in order to obtain improvements in efficiency and also optimization of data mining algorithms based on k-means. Cluster analysis is one of the major data mining methods for knowledge discovery in large databases and also a statistical method used for partitioning a sample into homogeneous classes to create an operational classifications. It is the process of grouping large data sets according to their similarity. Cluster analysis is a major tool in many areas of engineering, medical and scientific applications including data segmentation, discretization of continuous attributes, data reduction, outlier detection, noise filtering, pattern recognition, image processing [2], data compression, vector quantization and optimization. Also its applications appear in the commercial field: nowadays organizations have large volumes of data, related to their business processes, and resources, and this data can provide statistical information, so it will be useful to get some knowledge about this data to improve performance and profit. In the same time data clustering is useful in many non-commercial applications such as health care systems, web, etc [3].

Many clustering algorithms have been proposed in the literature and there are many clustering algorithms in use today. In this paper we focus on the k-means algorithm. One of the simple clustering algorithms, k-means, was first published over 50 years ago. In spite of the fact that thousands of clustering algorithms have been published since then, k-means is still widely used.

The k-means algorithm is well known for its efficiency in clustering large data set.

With a large number of variables k-means may be computationally faster than other algorithms (if k is small) and also may produce tighter clusters compared to other algorithms.

Starting from the idea that data mining is on the interface of Computer Science and Statistics, utilizing advances in both disciplines to make progress in extracting information from large databases, in this article we apply the k-means algorithm on a known set of data in order to obtain statistics with very low error limits, statistics that are necessary for the implementation of programs allowing for future improvement techniques.

On the other hand the results obtained in this work will be the starting point for new research to optimize data mining algorithms which are based on model.

The rest of this paper is organized as follows: in Section 2 we will view some related work, Section 3 presents the classical k-means technique, in Section 4 we will review the experimental results and in Section 5 we conclude and also we present the future work.

2. Related work

The progress of clustering methodology has been a truly interdisciplinary endeavor. The fields who collect and process real data have all contributed to clustering methodology. The notion of the "data clustering" appeared for the first time in the title of a 1954 article that presented anthropological data. Data clustering is also known as Q-analysis, typology, clumping and taxonomy (Jain and Dubes, 1988), depending on the domain where it is applied. In the literature some of the most popular books that published on data clustering are: classic, Sokal and Sneath (1963), Anderberg (1973), Hartigan (1975), Jain and Dubes (1988), extensively studied in data mining, Han and Kamber (2000) and machine learning, Bishop (2006).

Clustering algorithms can be divided into two groups: hierarchical and partitional. Hierarchical clustering algorithms recursively find nested clusters in two ways: in agglomerative mode and in divisive mode. The principles on which the agglomerative mode and the divisive mode are based are summarized below: for the agglomerative mode - starting with each data point in its own cluster and merging the most similar pair of clusters successively to form a cluster hierarchy, for the divisive mode - starting with all the data points in one cluster and recursively dividing each cluster into smaller clusters. Partitional clustering algorithms in comparison to hierarchical clustering algorithms, are sometimes more efficient in that they find all the clusters simultaneously as the partition of the data and do not impose a hierarchical structure.

The k-means algorithm is one of the most popular hierarchical algorithms. Ease of implementation, efficiency, simplicity and empirical succes are the main reasons for its popularity. Keeping data in main memory it works fast and may form the basis for other clustering algorithms. It is the best known algorithm because it is fast and converges to acceptable results in different areas. It is a simple iterative method to partition a given dataset into a user specified number of clusters, k. This algorithm has been discovered by several researchers across different disciplines, most notably Steinhaus (1956), Lloyd (1957, 1982), Ball and Hall (1965), Forgey (1965), Friedman and Rubin (1967) and McQueen (1967). Gray and Neuhoff provide a nice historical background for k-means placed in the larger context of hill-climbing algorithms [5].

3. Materials and methods

In this section we will extensively describe the k-means algorithm, description that includes the presentation of mathematical concepts and also enumeration of solutions consisting in proposing limitations and generalizations to solve various problems of the algorithm.

3.1. The k-means algorithm.

The algorithm works as follows: the user first must define the number of clusters to be founded by k-means; k-means starts by initializing a number of prototypes equal to number of desired clusters, then makes two steps; the first is assigning each point to its closest center, then moving each prototype to the mean of its assigned points. These two steps will be repeated until it converges to a solution.The algorithm depends on minimizing the square sum of error to assign each point to its cluster. Its simplicity and acceptable results mean it has a wide usage. However k-means has some drawbacks: first the user may not know in advance the number of clusters; also k-means is sensitive to the random initializing of its prototypes which may give poor clusters, since a different initialization may give different results, and this will cause k-means to converge to a suboptimal solution rather than the global optimum. Another disadvantage of k-means is its sensitivity to outliers [3].

The algorithm operates on a set of d-dimensional vectors, $D = \{x_i \mid i = 1, ..., N\}$, where $x_i \in \Re^d$ denotes the i-th data point. The algorithm is initialized by picking k points in $\Re^d$ as the initial k cluster representatives or "centroids". Techniques for selecting these initial seeds include sampling at random from the dataset, setting them as the solution of clustering a small subset of the data or perturbing the global mean of the data k times. Then the algorithm iterates between two steps till convergence:

Case 1: Data Assignment. Each data point is assigned to its *closest* centroid, with ties broken arbitrarily. This results in a partitioning of the data.

Case 2: Relocation of "means". Each cluster representative is relocated to the center (mean) of all data points assigned to it. If the data points come with a probability measure (weights), then the relocation is to the expectations (weighted mean) of the data partitions.

The algorithm converges when the assignment (and hence the c_j values) no longer changes. Each iteration needs $N \times k$ comparisons, which determines the time complexity of one iteration. The number of iterations required for convergence varies and may depend on N, but the algorithm can be considered linear in the dataset size.

One issue to resolve is how to quantify "closest" in the assignment step. The default measure of closeness is the Euclidean distance, in which case one can readily show that the non-negative cost function,

$$\sum_{i=1}^{N} \left(argmin_j \|x_i - c_j\|_2^2 \right)$$

will decrease whenever there is a change in the assignment or the relocation steps, and hence convergence is guaranteed in a finite number of iterations [5].

3.2. Limitations. Besides the great sensitivity to the initialization, the k-means algorithm presents other several problems. First, we can notice that k-means is a limiting case of fitting data by a mixture of k Gaussians with identical, isotropic covariance matrices ($\sum = \sigma^2 I$), when the soft assignments of data points to mixture components are hardened to allocate each data point solely to the most likely component. So, it will falter whenever the data is not well described by reasonably separated spherical balls, for example, if there are non-covex shaped clusters in the data. This problem may be alleviated by rescaling the data to "whiten" it before clustering, or by using a different distance measure that is more appropriate for the dataset. For example, information-theoretic clustering uses the kl-divergence to measure the distance between two data points representing two discrete probability distributions. It has been recently shown that if one measures distance by selecting any member of a very large class of divergences called Bregman divergences during the assignment step and makes no other changes, the essential properties of k-means, including guaranteed convergence, linear separation boundaries and scalability, are retained. As long as using a corresponding divergence this result makes k-means effective for a much larger class of datasets.

For describing non-convex clusters k-means can be associated with another algorithm. One first clusters the data into a large number of groups using k-means. These groups are then agglomerated into larger clusters using single link hierarchical clustering, which can detect complex shapes. This approach also makes the solution less sensitive to initialization, and since the hierarchical method provides results at multiple resolutions, one does not need to pre-specify k either. The cost of the optimal solution decreases with increasing k till it hits zero when the number of clusters equals the number of distinct data-points. This situation makes difficulties in two cases: the first case is directly compare solutions with different numbers of clusters and the second case is to find the optimum value of k. If the desired k is not known in advance, one will typically run k-means with different values of k, and then use a suitable criterion to select one of the results. Alternatively, one can progressively increase the number of clusters, in conjunction with a suitable stopping criterion. Bisecting k-means achieves this by first putting all the data into a single cluster, and then recursively splitting the least compact cluster into two using 2-means. Both these approaches thus alleviate the need to know k beforehand.

The algorithm is also sensitive to the presence of outliers, since "mean" is not a robust statistic. A preprocessing step to remove outliers can be helpful. Post-processing the results, for example to eliminate small clusters, or to merge close clusters into a large cluster, is also desirable [5].

3.3. Generalizations and connections. As mentioned earlier, k-means is closely related to fitting a mixture of k isotropic Gaussians to the data. Moreover, the generalization of the distance measure to all Bregman divergences is related to fitting the data with a mixture of k components from the exponential family of distributions. Another broad generalization is to view the "means" as probabilistic models instead of points in R^d. Here, in the assignment step, each data point is assigned to the most likely model to have generated it. In the "relocation" step, the model parameters are updated to best fit the assigned datasets. Such model-based k-means allow one to cater to more complex data, for example the sequences described by Hidden Markov models.

Also exists the posibility to "kernelize" k-means. Though boundaries between clusters are still linear in the implicit high-dimensional space, they can become non-linear when projected back to the original space, thus allowing kernel k-means to deal with more complex clusters. The k-medoid algorithm is similar to k-means except that the centroids have to belong to the data set being clustered. Fuzzy c-means is also similar, except that it computes fuzzy membership functions for each clusters rather than a hard one [5].

4. Experiments and results

The current section presents the description of two procedures of implementing k-means algorithm, database description with which we have worked and finally description of statistical results by comparing the two cases on the data set, results represented by diagrams. The implementation of k-means algorithm was performed in the C programming language after an approach by Ethan Brodsky. We used the DEV-C++ software Integrated Development Environment as presented in [12].

We described two algorithmic procedures involved in the implementation of the k-means algorithm: the Procedure *CalcClusterCentroids* which sets a new cluster center (see Algorithm 1) and the Procedure *ChooseAllClustersFromDistances* which determines the objects grouped around the center (see Algorithm 2).

The Procedure *CalcClusterCentroids* sets a new cluster center. First step is to initialize cluster centroid coordinate sums to zero. The next step is to calculate sum all points. For every point which cluster is it in, we update count of members in that cluster. Then the next step is to calculate the sum point coordinates for finding centroid. To find the centroid we divide each coordinates sum by number of members. For each cluster, if cluster centroid coordinate sums are zero then the cluster is empty. The last step is for each dimensions we divide by zero for any empty clusters.

Algorithm 1 Procedure CalcClusterCentroids

```
▷ Initialize cluster centroid coordinate sums to zero
for $i = 0 \to length(k)$ do
    $vectorcluster\_member\_count[i] \leftarrow 0$
    for $j = 0 \to length(dim)$ do
        $vectornew\_cluster\_centroid[i * dim + j] \leftarrow 0$
    end for
end for
▷ Sum all points
▷ For every point
for $i = 0 \to length(n)$ do
    ▷ Which cluster is it in
    $active\_cluster \leftarrow cluster\_assignment\_index[i]$
    ▷ Update count of members in that cluster
    $vectorcluster\_member\_count[active\_cluster] \leftarrow$
    $vectorcluster\_member\_count[active\_cluster] + 1$
    ▷ Sum point coordinates for finding centroid
    for $j = 0 \to length(dim)$ do
        $vectornew\_cluster\_centroid[active\_cluster * dim + j] =$
        $vectornew\_cluster\_centroid[active\_cluster * dim + j] + vectorX[i * dim + j]$
    end for
end for
▷ Divide each coordinate sum by number of members to find mean/centroid
▷ For each cluster
for $i = 0 \to length(k)$ do
    if $vectorcluster\_member\_count[i] = 0$ then
        $write$ "$Empty\ cluster$"
    end if
    ▷ For each dimension
    for $j = 0 \to length(dim)$ do
        ▷ Will divide by zero here for any empty clusters
        $vectornew\_cluster\_centroid[i * dim + j] =$
        $vectornew\_cluster\_centroid[i * dim + j]\ div\ cluster\_member\_count[i]$
    end for
end for
```

Algorithm 2 Procedure ChooseAllClustersFromDistances

```
▷ For each point
for $i = 0 \to length(n)$ do
    $best\_index \leftarrow -1$
    $closest\_distance \leftarrow BIG\_double$
    ▷ For each cluster
    for $j = 0 \to length(k)$ do
        ▷ Distance between point and cluster centroid
        $current\_distance \leftarrow vectordistance\_array[i * k + j]$
        if $current\_distance < closest\_distance$ then
            $best\_index \leftarrow j$
            $closest\_distance \leftarrow current\_distance$
        end if
    end for
    ▷ Record in array
    $vectorcluster\_assignment\_index[i] \leftarrow best\_index$
end for
```

The Procedure *ChooseAllClustersFromDistances* determines the objects grouped around the center. The first step through the whole sequence of points. The next step through the whole sequence of clusters. For each cluster we define the distance between point and cluster centroid. The last step, using a vector, determines the objects grouped around the center.

Database that we used in this paper to obtain statistical results is called *Perfume.data* which contains 20 classes of 28 instances each (560 instances), where each class refers to a type of perfume [14].

In the experiment we applied k-means algorithm on the database *Perfume.data* for two cases and after clustering we obtained statistics on the following notions: Euclidean Distances and Plot of Means for Each Cluster [13].

Euclidean Distances is the straight line distance between two points in Euclidean space. In a plane with p_1 at (x_1, y_1) and p_2 at (x_2, y_2) it is $\sqrt{((x_1 - x_2)^2 + (y_1 - y_2)^2)}$. Plot of Means for Each Cluster provides a succinct graphical representation of how well each object is defined by the each cluster.

For the first case we worked with three clusters and we presented the Euclidean Distances statistics (see Table 1) and also the Plot of Means for Each Cluster (see Figure 1). The results obtained in *Table* 1 represent the Euclidean Distances between Clusters, Distances below diagonal and Squared distances above diagonal.

Cluster Number	**No.1**	**No.2**	**No.3**
No.1	0.0000	292214.4	22236.2
No.2	540.5686	0.0	221376.0
No.3	471.5254	470.5	0.0

TABLE 1. Euclidean Distances

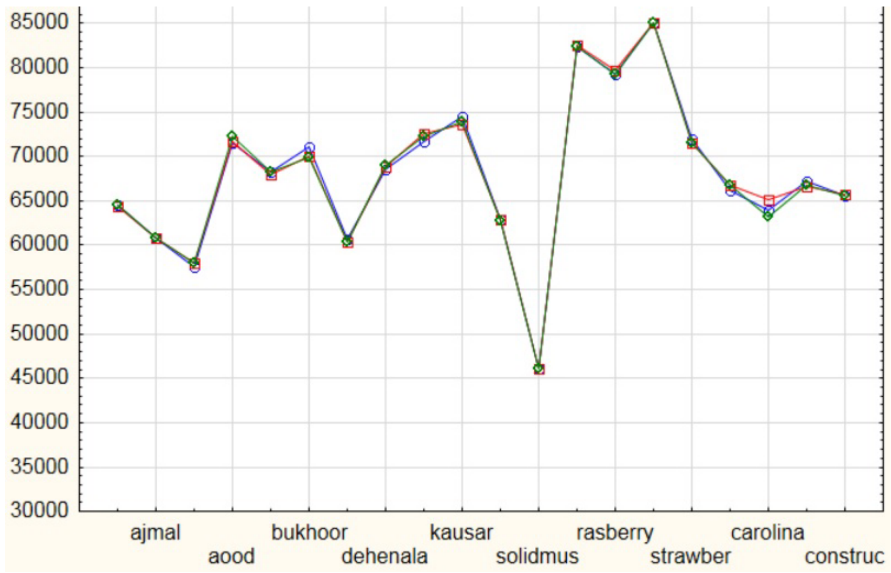

FIGURE 1. Plot of Means for 3 Clusters - Cluster 1=blue, Cluster 2=red, Cluster 3=green (Perfume.sta)

Applying k-means with a value of k=3 onto the *Perfume.data* set we obtained results as graphical representations in *Figure* 1 - Plot of Means for Each Cluster. Each cluster determines for each type of perfume statistics as Mean, Standard Deviation and Variance.

For the second case we worked with four clusters and we presented the Euclidean Distances statistics (see Table 2) and also the Plot of Means for Each Cluster (see Figure 2). The results obtained in *Table* 2 represent the Euclidean Distances between Clusters, Distances below diagonal and Squared distances above diagonal.

Cluster Number	**No.1**	**No.2**	**No.3**	**No.4**
No.1	0.0000	181538.4	2223571.9	279555.7
No.2	426.0723	0.0	380679.8	243448.4
No.3	472.8339	617.0	0.0	217050.4
No.4	528.7303	493.4	465.9	0.0

Table 2. Euclidean Distances

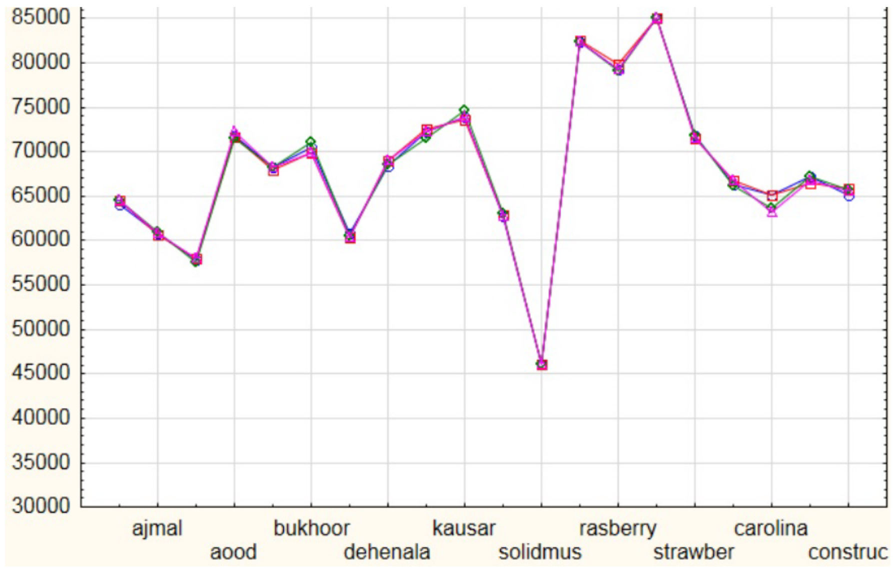

Figure 2. Plot of Means for 4 Clusters - Cluster 1=blue, Cluster 2=red, Cluster 3=green, Cluster 4=pink (Perfume.sta)

Applying k-means with a value of k=4 onto the *Perfume.data* set we obtained results as graphical representations in *Figure* 2 - Plot of Means for Each Cluster. Each cluster determines for each type of perfume statistics as Mean, Standard Deviation and Variance.

These statistics obtained from the experiment, by detailed analysis, can be used to improve the future implementation techniques at the k-means algorithm to optimize data mining algorithms which are based on model.

5. Conclusions and future work

The current paper highlights statistical improvements on the k-means algorithm and proves their efficiency through a statistical study on a known database. In this paper we discussed the concepts of data mining, cluster analysis and primarily we focused on presenting thoroughly the k-means algorithm, both from the mathematical point of view and from the point of view of its implementation.

Despite the problems presented in Section 3, k-means remains the most widely used partitional clustering algorithm in practice. The algorithm is simple, easily understandable and reasonably scalable, and can be easily modified to deal with streaming data. To deal with very large datasets, substantial effort has also gone into further speeding up k-means, most notably by using *kd*-trees or exploiting the triangular inequality to avoid comparing each data point with all the centroids during the assignment step. Continual improvements and generalizations of the basic algorithm have ensured its continued relevance and gradually increased its effectiveness as well.

In the future we aim to focus on the comparison of statistically viewpoint of the clustering method which includes k-means algorithm with the classification method which includes k-nearest neighbors algorithm. We also started investigating an approach to applying the k-means algorithm in the domain of the topographic engineering, especially for developing topographical maps.

References

[1] J. Grabmeier and A. Rudolph, Techniques of Cluster Algorithms in Data Mining, *Data Mining and Knowledge Discovery* **6**(4) (2002), 303–306.

[2] D. Birant and A. Kut, ST-DBSCAN: An algorithm for clustering spatialtemporal data , *Data & Knowledge Engineering* **60**(1) (2007), 208-221.

[3] M. F. Eltibi and W. M. Ashour, Initializing K-Means Clustering Algorithm using Statistical Information, *International Journal of Computer Applications* **29** (7), (2011).

[4] A. Ahmad and L. Dey, A k-mean clustering algorithm for mixed numeric and categorical data, *Data & Knowledge Engineering*, **63**(2) (2007), 503-527.

[5] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, Top 10 algorithms in data mining, *Knowledge and Information Systems* **14**(1) (2008), 1–37.

[6] Z. Güngör and A. Ünler, K-Harmonic means data clustering with tabu-search method, *Applied Mathematical Modelling* **32**(6) (2008), 1115-1125.

[7] Y. Li and H. Wu, A Clustering Method Based on K-Means Algorithm, *Physics Procedia* **25** (2012), 1104-1109.

[8] T. Tarpey, A parametric k-means algorithm, *Computational Statistics* **22**(1) (2007), 71–89.

[9] B. D. Gerardo, J.-W. Lee, Y.-S. Choi, and M. Lee, The K-Means Clustering Architecture in the Multi-stage Data Mining Process, In: Computational Science and Its Applications ICCSA 2005, LNCS **3481**, Springer-Verlag Berlin Heidelberg (2005), 71–81.

[10] Y. Zhao and G. Karypis, Data Clustering in Life Sciences, *Molecular Biotechnology* **31** (1) (2005), 55–80.

[11] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, An Efficient k-Means Clustering Algorithm: Analysis and Implementation, *IEEE Transaction on Pattern Analysis and Machine Intelligence* **24**(7) (2002), 881–892.

[12] `http://www.bloodshed.net/devcpp.html`, web reference last accessed in April 2015.

[13] `http://www.statsoft.com`, web reference last accessed in May 2015.

[14] `http://archive.ics.uci.edu/ml/`, web reference last accessed in May 2015.

(Dana Mihai, Mihai Mocanu) Department of Computers and Information Technologies, Faculty of Automation, Computers and Electronics, University of Craiova, 107 Bvd. Decebal, Craiova, 200440, Romania
E-mail address: `mihai_danam@yahoo.com, mmocanu@software.ucv.ro`