Beyond K-means: Clusters Identification for GIS

Chapter · January 2011

DOI: 10.1007/978-3-642-19766-6_8

4 authors:

Andreas Hamfelt
Uppsala University
43 PUBLICATIONS **290** CITATIONS
SEE PROFILE

Mikael Karlsson
Uppsala University
1 PUBLICATION **3** CITATIONS
SEE PROFILE

Tomas Karl Ernst Thierfelder
Swedish University of Agricultural Sciences
64 PUBLICATIONS **580** CITATIONS
SEE PROFILE

Vladislav B. Valkovsky
Uppsala University
9 PUBLICATIONS **62** CITATIONS
SEE PROFILE

Beyond K-means: Clusters Identification for GIS

Andreas Hamfelt[1], Mikael Karlsson[2], Tomas Thierfelder[3], and Vladislav Valkovsky[1],

[1] Uppsala University, Informatics and Media, Box 513,
SE-751 20 Uppsala, Sweden
{Andreas.Hamfelt, Vladislav.Valkovsky}@im.uu.se

[2] Eins SAP Consulting, Bellmansgatan 2,
SE-118 20 Stockholm, Sweden
mikael@mkarlsson.se

[3] Swedish University of Agricultural Sciences, Department of Energy and Technology, P.O. Box 7032,
SE-750 07 Uppsala, Sweden
Thomas.Thierfelder@et.slu.se

Abstract. Clustering is an important concept for analysis of data in GIS. Due to the potentially large amount of data in such systems, the time complexity for clustering algorithms is critical. K-means is a popular clustering algorithm for large scale systems because of its linear complexity. However, this requires a priori knowledge of the number of clusters and the subsequent selection of their centroids. We propose a method for K-means to find automatically the number of clusters and their associated centroids. Moreover, we consider recursive extension of the algorithm to improve visibility of the results at different levels of abstraction, in order to support the decision making process.

Keywords: Clustering; K-means algorithm; Initial centroids determination

1 Introduction

Clustering is an important concept to unite objects in groups (clusters) according to their similarity. Cluster analysis has been applied in a wide variety of fields and in particular in GIS [6, 7, 8, 9]. The large amount of data in GIS requires minimization of the time complexity for data analysis algorithms in general and for clustering algorithms in particular. The K-means algorithm is one of the most widely used clustering techniques in GIS applications [1, 2] because of its linear time complexity. However, this requires a priori knowledge of the number of clusters and the

subsequent selection of their centroids. In practice, unfortunately, most often neither the number of clusters nor the reasonable initial partition for centroids are known for users. Therefore, identifying the number of clusters and initial centroids selection in advance is a very important topic in cluster analysis [18].

The objective of this paper is to automate the process for finding the number of clusters and their associated centroids, which in turn gives an opportunity to improve visibility and tractability of the results by a recursive extension of the modified K-means algorithm.

The paper is structured as follows. Sect. 2 briefly describes the basic K-means algorithm and its underlying problems. Sect. 3 presents some previous attempts to solve the abovementioned problems. Then Sect. 4 describes K-means modification with determination of initial centroids. Experiments with the modified K-means clustering and proper results are presented in Sect. 5. Sect. 6 demonstrates the potential opportunity to realize a recursive extension for the modified K-means clustering procedure. In Sect. 7 we summarize the results, discuss further work and conclude.

2 The Basic K-means Algorithm and its Problems

The K-means algorithm [11, 12] is one of the best-known and most popular clustering algorithms [13, 14]. K-means seeks an optimal partition of the data by minimizing the sum of the squared error (SSE) criterion [16], (the sum of the squared Euclidean distance between each data point and its nearest cluster center) with an iterative optimization procedure, which belongs to the category of hill-climbing algorithms [10].

In the basic algorithm [15] we first choose K initial centroids, where K is a user-specified parameter, namely the number of clusters desired. Each point is then assigned to the closest centroid, and each collection of points assigned to a centroid is a cluster. The centroid of each cluster is then updated based on the points assigned to the cluster. We repeat the assignment and update steps until no point changes the clusters, or equivalently, until centroids remain the same. K-means is formally described on Fig. 1

1. Select K points as initial centroids.
2 **repeat**
3 Form K clusters by assigning each point to its closest centroid.
4 Recompute the centroid of each cluster.
5 **until** Centroids do not change.

Fig. 1. Basic K-means algorithm

K-means is regarded as one of the most valuable clustering methods [13] due to its ease of implementation. It works well for many practical problems, particularly when the resulting clusters are compact and hyperspherical in shape [10]. The time complexity of K-means is approximately linear [15], therefore K-means is a good choice for clustering large-scale data sets.

While K-means has some desirable properties, it also suffers from several major drawbacks.

The first problem of K-means is that the iterative procedure cannot guarantee the convergence to a global optimum, although the convergence of K-means was proved [17]. Since K-means can converge to a local optimum, different initial points generally lead to different convergence centroids, which makes it important to start with a reasonable initial partition in order to achieve high quality clustering solutions. However, in theory, there are no efficient and universal methods for determining such initial partitions [10].

The second problem of K-means is that K-means assumes the number of clusters K to be already known by the users, which, unfortunately, usually is not true in practice. Like the situation for cluster initialization, there are also no efficient and universal methods for the selection of K [10]. Therefore, identifying K in advance becomes a very important topic in cluster analysis [18].

To illustrate the disadvantage of requiring a predefined number of clusters, two diagrams are shown below. Both diagrams have three visual clusters with 50 points in each.

Fig. 2 shows a successful calculation with three predefined clusters (the clusters are recognized by the shape of the points: squares, circles and triangles).

In the diagram in Fig. 3 the predefined number of clusters is two. The real number of clusters is three and here the problem with determining a correct number of clusters has occurred (the clusters are recognized by the shape of the points: squares and circles)

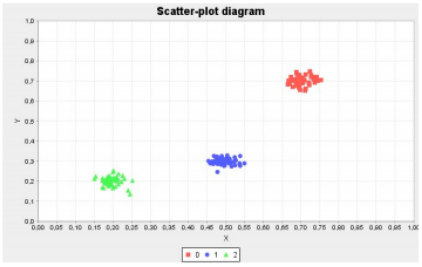

Fig. 2. K-means clustering with three predefined clusters

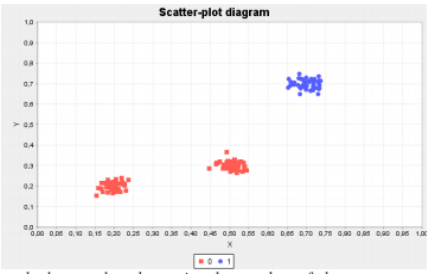

Fig. 3. K-means clustering with two predefined clusters

3 Previous Work

To overcome the abovementioned problems different approaches and techniques have been proposed.

Krishna and Murty [19] proposed the genetic K-means (GKA) algorithm, integrating a genetic algorithm with K-means, in order to achieve a global search and fast convergence.

Jain and Dubes [20] recommend running the algorithm several times with random initial partitions. The clustering results on these different runs provide some insights into the quality of the ultimate clusters.

Forgy's method [11] generates the initial partition by first randomly selecting K points as prototypes and then separating the remaining points based on their distance from these seeds.

Likas et al. [21] proposed a global K-means algorithm consisting of series of K-means clustering procedures with the number of clusters varying from 1 to K. One disadvantage of the algorithm lies in the requirement for executing K-means N times for each value of K, which causes high computational burden for large data sets.

Bradley and Fayyad [16] presented a refined algorithm that utilizes K-Means M times to M random subsets sampled from the original data.

The most common initialization (used for example in the software packages SAS Enterprise Miner and Matlab) was proposed by Pena, Lozano et al. [22]. This method is selecting randomly K points as centroids from the data set. The main advantage of the method is simplicity and an opportunity to cover rather well the solution space by multiple initialization of the algorithm.

Ball and Hall proposed the ISODATA (Iterative Self-Organizing Data Analysis Technique) algorithm [23], which is estimating K dynamically. For selection of a proper K, a sequence of clustering structures can be obtained by running K-means several times from the possible minimum Kmin to the maximum Kmax [10]. These structures are then evaluated based on constructed indices and the expected clustering solution is determined by choosing the one with the best index [24].

The popular approach for evaluating the number of clusters in K-means is the Cubic Clustering Criterion [25] used in SAS Enterprise Miner.

Using hierarchical clustering algorithms to get approximation for initial centroids has also been proposed [27, 28, 29]

4 K-Means Modification

We will try to overcome the two abovementioned K-means problems by dividing the clustering process into two stages. The first stage is the determination of the number of clusters and their centroids in the data set. The second stage is K-means clustering of a given data set with centroids determined during the first stage.

4.1 Initial Centroids Determination

To find the initial centroids we will use a so called training set - a randomly and uniformly chosen subset of the given data set [30]. To determine the initial centroids (and the following number of clusters) we have to find the clusters for the generated training set without a priori knowledge about them.

Because we are not to presuppose some predefined types of distributions for the points in the clusters neither for the data set nor for the training set, we will need a clustering procedure which is not sensitive to types of data distribution.

Such a procedure was proposed in [3, 4] to investigate matrix topology and we will use it for our purpose. Let us describe this procedure briefly.

Originally the procedure was formulated for general (asymmetric) matrixes. We reformulate it for our Euclidean case.

So, let us have an n×n symmetrical matrix D of Euclidean distances between all n points of the training set. Let the m points be selected from the training set according to some procedure. Such a group of points is called an embryo.

Let us enumerate indexes for the points of the training set in such a way that the points from the embryo will have consecutive indexes 1, 2, . . , m . After such an enumeration the leftmost m×m submatrix of matrix D will represent distances between m points of the embryo.

In our case we are treating the embryo as the initial cluster and different algorithms could be used to create it. It was shown in [5] that nearest neighbor clustering [26] is acceptable for our purpose and we will follow this approach.

The next step is to check if some other points from the training set are similar enough to be combined with the embryo. In our case a point outside the embryo will be combined with the embryo if distribution of distances from that given point to points in the embryo is statistically similar to the distribution of distances of points inside the embryo. The criterion of similarity is The Two-sample Kolmogorov-Smirnov test of fit [31, 32].

If some point outside the embryo satisfies The Two-sample Kolmogorov-Smirnov test it will be included in the embryo with index m+1 and the rest of the points of the training set outside this new (m+1)×(m+1) cluster will be checked in a similar way. This process will continue until we cannot find any more points belonging to our growing cluster.

As a result the first cluster will be created and if k1 points were combined in the cluster then the leftmost k1×k1 submatrix of matrix D will represent the distances between the points in the first cluster. After that the same procedure is repeated for the points of the training set not included in the first cluster and as a result the second cluster will be created, etc.

In the end of the process the created clusters will be represented as submatrixes on the main diagonal of matrix D. Because matrix D is symmetrical it is sufficient to use triangle matrixes representation. On the basis of the identified clusters, the coordinates of their centroids are determined. To improve quality for identification of centroids we are repeating the initial centroids determination procedure a few times and selecting as final centroids determination the variant with minimum SSE.

4.2 K-means Clustering with Known Centroids

During this stage we are using only parts 1 and 3 from the basic K-means algorithm (Fig. 1). Part 1- "Select K points as initial centroids" is realized by the initialization stage of our algorithm. Part 3 –" Form K clusters by assigning each point to its closest centroid". Because we identified centroids during the first stage we do not need any iterations to recompute centroids of clusters and it is sufficient only to form clusters by assigning each point of data set to its closest centroid. Such an opportunity decreases drastically the time complexity of the K-means algorithm because we are not repeating the K-means procedure many times to reach the convergence criterion.

5 Experiments and Results

To check the approach a series of experiments was carried out. For visibility of the results clustering was done for the two dimensional case. Formally the results could be extended for higher dimensionalities.

Clusters were modeled by given the coordinates of their centroids and randomly generated points around the centroids. Coordinates of the points for each axis were generated according to normal distribution with given mathematical expectation (coordinates of centroid) and standard deviation.

So, for each experiment the following parameters were defined: N- number of points in the data set; K- number of generated clusters; n- number of points in training set; m- number of points in embryo; σxi , σyi- standard deviations for the x and y coordinates of cluster i. α- significance level for The Two-sample Kolmogorov-Smirnov test.

Fig. 4 demonstrates our clustering procedure for the case of well-separated clusters. Here $N = 10000$, $K = 10$, $n = 400$, $m = 8$, $\sigma x = \sigma y = 0.01$ for all clusters, $\alpha = 0.05$. As we can see the modified K-means clustering algorithm has recognized all 10 generated clusters correctly.

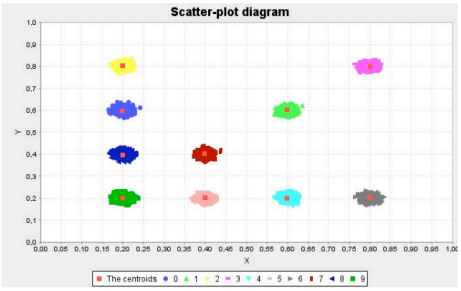

Fig. 4. Modified K-means clustering; correct clusters identification.

Fig. 5 demonstrates the clustering procedure for the case of well-separated clusters for N = 10000, K = 10, n = 100, m = 8, σx = σy = 0.01 for all clusters, α = 0.05. As we see here the modified K-means clustering algorithm has recognized correctly only 6 clusters but incorrectly combined the remaining 4 clusters into 2 clusters. This is a negative effect of a too small sized training set: the training set here is not representing properly the points of the data set.

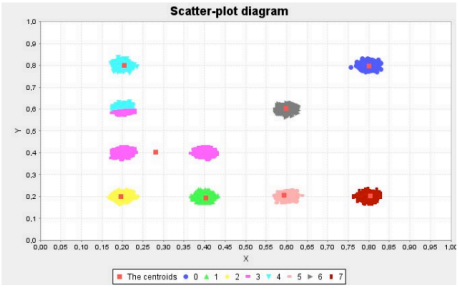

Fig. 5. Modified K-means clustering; incorrect clusters identification.

The experiment represented in Fig. 6 is similar to the experiment in Fig. 4 with N = 100000 points . We can see that clusters are identified correctly for a large number of points in the data set, because of a proper size of the training set, n = 400

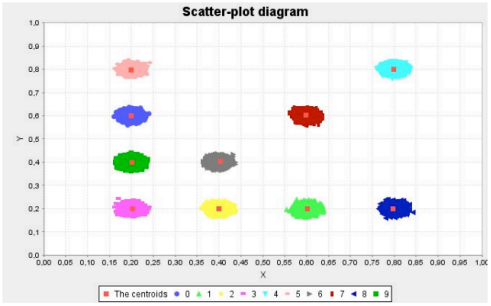

Fig. 6. Modified K-means clustering; correct clusters identification.

An important property of the clustering process is the opportunity to separate closely arranged clusters. Fig. 7 demonstrates the result of modified K-means clustering for closely arranged clusters. Here N = 10000, K = 3, n = 400, m = 8, σx = σy = 0.02 for all clusters, α = 0.05. As we see all three clusters are identified correctly.

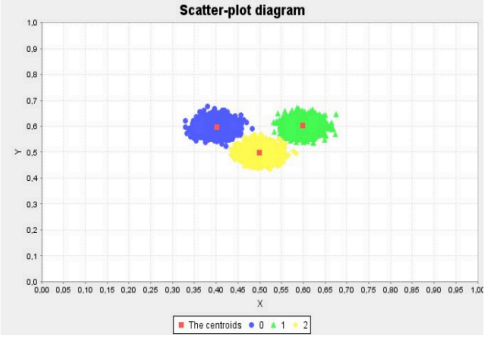

Fig. 7. Modified K-means clustering; closely arranged clusters; correct clusters identification.

6 Recursive Extension

The large amount of data in GIS and the potentially complex structure of the result of the cluster analysis give rise to a need to represent the results at different levels of abstraction. For instance, some clusters could consist of smaller clusters, in turn consisting of other clusters, etc. An opportunity to represent clustering results by proper levels of abstraction could improve visibility of the results and simplify the decision making process during interpretation of GIS data.

The automation of initialization of centroids in modified K-means clustering provides a *hypothetical* opportunity to realize a recursive extension of the algorithm. As discussed above, in the process of creating clusters we are using the principle of statistical similarity for the distribution of distances of points inside clusters and the distribution of distances of points inside the embryos. So, if the embryos will be created on the base of a "rough picture" of our data set then the "rough clustering" procedure will be performed and proper clusters will be recognized. Each of the recognized clusters in turn could be treated as a new data set and modified K-means clustering could be used recursively for each of these data sets. This recursive procedure could be repeated until a proper level of abstraction is obtained. In this hypothetical approach we need some instrument to obtain embryos, which are representing the current data sets with different precision. The size n of the training set can serve as such an instrument. The larger the size of the training set, the more precisely it represents the given data set.

The experiments presented below illustrate the potential opportunity of the considered approach. In these experiments 10 clusters were modeled in such a way that they could be treated as two "big" clusters each one including five smaller closely arranged clusters. In all these experiments we are using the same data set with N = 100000, K = 10, $\sigma x = \sigma y = 0.01$ for all clusters, $\alpha = 0.05$.

Fig. 8 demonstrates the first recursive step of the clustering procedure for the following parameters: n = 50, m = 25. As we can see the modified K-means clustering algorithm has recognized only one cluster, or simply speaking, combined all data set in the cluster.

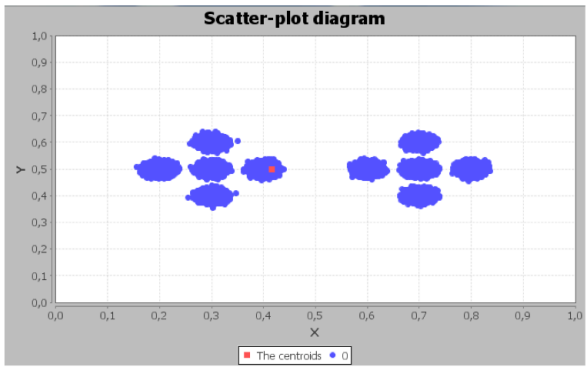

Fig. 8. First recursive step; combining the whole data set into one cluster.

Fig. 9 demonstrates the second recursive step with n = 100, m = 25. We can see that, with these parameters the modified K-means clustering algorithm has recognized two "big" clusters.

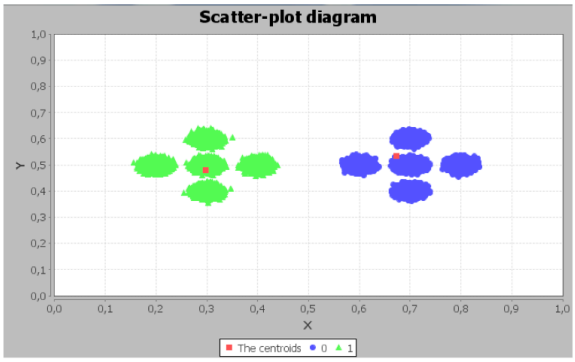

Fig. 9. Second recursive step; two "big" clusters are recognized.

Fig. 10 demonstrates the third recursive step with n = 400, m = 10. We can see that with these parameters the modified K-means clustering algorithm has recognized five clusters inside each "big" cluster.

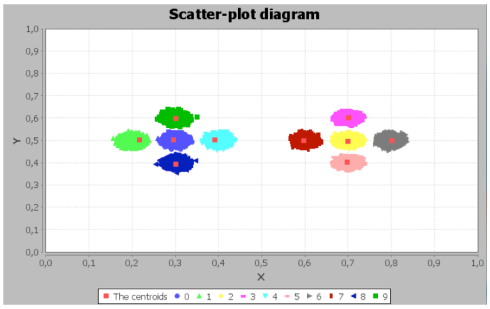

Fig. 10. Third recursive step; five clusters inside each "big" cluster are recognized

These experiments only illustrate the potential opportunity to realize the considered recursive extension of the modified K-means clustering algorithm. A lot of research has to be done to investigate applicability of such an approach to clusters identification for GIS.

7 Conclusions and Future Work

This paper presents a two-staged modified K-means clustering algorithm. In contrast to basic K-means clustering, this algorithm does not require a priori knowledge neither about the number of clusters nor the initial partition of centroids.

To support the robustness of the algorithm concerning data distribution, nonparametric statistics is used for clusters identification.

The considered possibility of a recursive extension of the modified K-means clustering algorithm is shown and illustrated with experiments.

The proposed modified K-means clustering algorithm needs further investigation. Our experiments rely on modeled data sets. How the algorithm will work with real data (in particular with outliers and noise) is an open question. Further investigation should include an analysis of the efficiency of the proposed clustering procedure for different parameters of the algorithm and different characteristics of the analyzed data. How efficient is the algorithm for different shapes of clusters and for high-dimensional data? Will the considered recursive extension of the algorithm work in reality? To validate the suggested modified K-means clustering algorithm these and other topics will need to be considered.

References

1. Bacao F, Lobo V, Painho M (2005) Self-organizing maps as substitutes for K-means clustering. In: Sunderam VS et al. (eds): ICCS 2005, LNCS 3516, pp 476-483
2. Galjano P, Popovich V (2007) Intelligent images analysis in GIS. In: Popovich VV et al. (eds) Information fusion and geographic information systems. Proceedings of the third international workshop, LNG&C, pp 45-68
3. Valkovsky VB, Gerasimov MB (1995) Approximate recursive solution for large scale traveling salesman problem (in Russian). Proceedings of St. Petersburg Electrotechnical University, No 489, St Petersburg, pp 27-37
4. Valkovsky VB, Gerasimov MB, Savvin KO (1999) Phase transitions inTSP and matrix topology. In: Proceedings of the joint workshop on integration of AI and OR techniques in constraint programming for combinatorial optimization problems. Universita degli studi di Ferrara- Facolta di Ingegneria, Italy
5. Karlsson M (2009) Modifying K-means clustering for Data Mining. Master thesis, Uppsala University
6. Murray AT, Estivil-Castro V (1998) Cluster discovery techniques for exploratory spatial data analysis. In: International journal of geographical information science, 12, Issue 5, July, pp 431-443
7. Pick J (2004) Geographic information systems. Proceedings of American conference on information systems, AMCIS 2004
8. Jain A, Murty M, Flynn P (1999) Data clustering: a review. ACM computing surveys 31(3): 264-323
9. Kolatch E (2001) Clustering algorithms for spatial databases: a survey, http://citeseer.ij.nec.com/436843.html
10. Rui X, Wunsch DC II (2009) Clustering. IEEE Press series on computational intelligence, John Wiley & Sons
11. Forgy E (1965) Cluster analysis of multivariate data; efficiency vs. interpretability of classifications. Biometrics, 21: pp 768-780
12. MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium, 1, pp 281-297
13. Duda R, Hart P (2001) Pattern classification, 2nd edn. New York, NY: John Wiley & Sons
14. Theodoridis S, Koutroumbas K (2006) Pattern recognition, 3rd edn. San Diego, CA: Academic Press
15. Tan PN, Steinbach M, Kumar V (2006) Introduction to Data Mining. Addison Wesley
16. Bradley P, Fayyad U (1998) Refining initial points for K-means clustering. International conference on machine learning (ICML-98), pp 91-99
17. Selim S, Ismail M (1984) K-means-type algorithms: a generalization convergence theorem and characterization of local optimality. IEEE Transactions on pattern analysis and machine intelligence, 6(1): pp 81-77
18. Dubes R (1993) Cluster analysis and related issue. In: Chen C, Pau L, Wang P (eds) Handbook of pattern recognition and computer vision, River Edge, NY: World Science Publishing Company, pp 3-32

19. Krishna K, Murty M (1999) Generic K-Means algorithm. IEEE Transactions on systems, man, and cybernetics- part B: Cybernetics, 29(3): pp 433-439
20. Jai A, Dubes R (1988) Algorithms for clustering data. Englewood Cliffs, NJ: Prentice Hall
21. Likas A, Vlassis N, Verbeek J (2003) The global K-means clustering algorithm. Pattern recognition, 36(2), pp 451-461
22. Pena JM, Lozano JA, Larranaga P (1999) An empirical comparison of four initialization methods for K-means algorithm. Pattern recognition letters 20: pp 1027-1040
23. Ball G, Hall D (1967) A clustering technique for summarizing multivariate data. Behavioral science, 12: pp 153-155
24. Milligan G, Cooper M (1985) An examination of procedures for determining the number of clusters in a data set. Psychometrika, 50: pp 150-179
25. SAS Institute Inc., SAS technical report A-108 (1983) Cubic clustering criterion. Cary, NC: SAS Institute Inc., 56 pp
26. Cover TM, Hart PE (1967) Nearest neighbor pattern classification. IEEE Trans. inform theory 13(1): 21-27
27. Fisher DH (1987) Knowledge acquisition via incremental conceptual clustering. Machine learning 2: 139-172
28. Higgs RE, Bemis KG,Watson I, Wikel J (1997) Experimental designs for selecting molecules from large chemical databases. Journal of chemical information and computer sciences (37) 5: 861-870
29. Meila M, Heckerman D (2001) An experimental comparison of several clustering and initialization methods. Machine learning 42: 9-29
30. Han J, Kamber M (2006) Data Mining. Concepts and techniques. Elsevier Inc.
31. Wasserman L (2007) All of nonparametric statistics. Springer-Verlag
32. Kolmogorov A (1941) Confidence limits for an unknown distribution function. Annals of mathematical statistics 12, 461-483