Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 177 (2007) 1400–1408
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

The *k*-means range algorithm for personalized data clustering in e-commerce

Georgios P. Papamichail [a,*], Dimitrios P. Papamichail [b]

[a] *Department of Management Science and Technology, Athens University of Economics and Business, 47A Evelpidon & 33 Lefkados Str., Athens 11 362, Greece*
[b] *Computer Science Department, State University of New York at Stony Brook, Stony Brook, NY 11794-4400, USA*

Available online 16 June 2005

Abstract

This paper describes the *k*-means range algorithm, a combination of the partitional *k*-means clustering algorithm with a well known spatial data structure, namely the range tree, which allows fast range searches. It offers a real-time solution for the development of distributed interactive decision aids in e-commerce since it allows the consumer to model his preferences along multiple dimensions, search for product information, and then produce the data clusters of the products retrieved to enhance his purchase decisions. This paper also discusses the implications and advantages of this approach in the development of on-line shopping environments and consumer decision aids in traditional and mobile e-commerce applications.

Keywords: Heuristics; Distributed consumer decision-making; Range search; Data clustering; Personalized systems

1. Introduction

During the last few years we have witnessed an enormous growth in the number of organizations that develop sophisticated interactive web environments to accommodate the on-line shopping experiences of consumers. Shoppers desire to define their preferences and customize the purchase information within the electronic shopping environment according to their individual needs. In most situations, they are not able to evaluate all available alternatives and typically follow a two-step model to fulfil their purchasing processes. In the first step, they identify a subset of the available alternatives by choosing from a vast range

* Corresponding author. Tel.: +30 108203671; fax: +30 108828078.
E-mail addresses: pmichael@aueb.gr (G.P. Papamichail), dimitris@cs.sunysb.edu (D.P. Papamichail).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2005.04.011

of products, and, in a second step, they perform relative comparisons among these to arrive at their final decisions [12]. Interactive decision aids, tools that assist shoppers in their purchase activities, appear to have strong favourable effects on both the quality and the efficiency of the purchasing process [6].

At the same time, data clustering algorithms promise to deliver efficient solutions to many of the problems arising from the interactions of consumers with the increasing volume of information in on-line shopping environments. By the term clustering we mean the unsupervised process through which a large number of data items are classified into disjoint and homogenous groups (clusters) based on similarity. Although promising in many application areas such as pattern classification, data mining or decision-making, it poses several restrictions to the decision maker when little information is known a priori about the nature of the data [8]. Therefore the choice of an appropriate method, taking into account these restrictions, is crucial to the effective exploration of interrelationships among the data items, in order to make a meaningful assessment. A simple and commonly used algorithm for producing clusters by optimising a criterion function, defined either globally (over all patterns) or locally (on a subset of the patterns), is the k-means algorithm [10]. It starts with a random initial partition and keeps reassigning the patterns to clusters based on the similarity between the pattern and the clusters. Its main drawback remains the fact that it is very expensive for the very large data sets of patterns met in real life applications while it often converges to a local minimum.

In our research approach the consumer models each one of his preferences as a range of values within a user-defined interval. These value intervals form a multi-dimensional ''preference vector'', where the dimensions are defined by the number d of the decision parameters. This d-dimensional ''preference vector'' is then mapped to an iso-oriented rectangle, where each side of the rectangle corresponds to each one of the d decision preference dimensions [11]. At the same time, each one of the products listed in an online store possesses certain values for each one of the corresponding decision dimensions. These values form an d-dimensional data point for each product in the store. Thus, the subset of the products that meet the criteria set by the consumer is comprised of the data points that lie inside the iso-oriented rectangle. The k-means clustering algorithm is then employed to classify these data points into disjoint clusters allowing the consumer to easily distinguish between alternatives and eliminate at a first level dominated clusters and at a second level dominated data items.

This paper combines the multi-dimensional range tree and the k-means algorithm to produce an orthogonal range data clustering method, which facilitates the efficient personalized decision-making in e-commerce applications. The aforementioned approach allows the consumer to model his decision preferences along multiple dimensions, not just price, defining in this way multi-dimensional decision vectors. Then the range search reduces the initial data set of patterns that need to be examined for similarity along these multiple dimensions, thus narrowing significantly the decision space. Finally, the k-means algorithm produces disjoint data clusters and the consumer can focus more effectively on the search for optimal solutions that fit his needs. This approach allows the development of distributed interactive decision aids in e-commerce since it relies on a set of scalable, user-intuitive, real-time algorithms with affordable time and space complexities.

2. Research approach

First the consumer states his preferences by defining value intervals for each one of them. For the sake of simplicity and visualisation purposes we assume he states only two. In the first one his preference lies between the values x_1 and x_2 and in the second between the values y_1 and y_2. In this way, an iso-oriented rectangle is formed, named R, a rectangle with sides parallel to the axis. The products offered by an online store are depicted as two-dimensional points with values p_i and q_i, and

$$p_i, q_i \in A = \{(p_1, q_1), (p_2, q_2), \ldots, (p_n, q_n)| \quad \text{where } (p_i, q_i) \in \Re^2, i, n \in I\}.$$

This representation forms the decision scene as shown in Fig. 1.

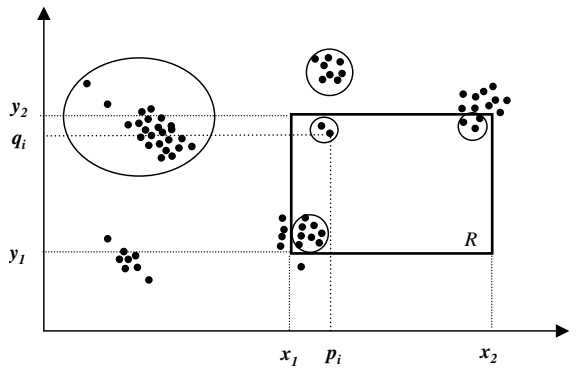

Fig. 1. The decision scene.

The consumer performs a two-stage purchasing process: in the first stage he identifies all the products that lie inside his preference rectangle, that is all data points (p_i, q_i) where $x_1 \leqslant p_i \leqslant x_2$ and $y_1 \leqslant q_i \leqslant y_2$, and in the second stage he focuses on the clusters formed by these data points. In Fig. 1, three clusters lie inside the R rectangle while three others lie outside. The algorithm proposed in this paper serves these two stages in a straightforward manner: First, it employs a range search to determine all the data points lying inside the consumer preference rectangle and, second, it uses the k-means algorithm family to calculate the corresponding clusters (Fig. 2).

It must be noted that for the clusters lying inside the rectangle only the points enclosed in it are reported, reducing significantly not only the total number of clusters reported but also the set of data points used, resulting in an enormous reduction in computational time. Furthermore, the consumer need not define his exact decision-making strategy a priori but can redefine his preferences along the value intervals after he retrieves and calculates the desired product items and corresponding clusters. In the following sections we present the k-means algorithm along with its variations for categorical and mixed values. Then we describe the proposed k-means range algorithm, comment on its computational complexity and discuss the implications of its deployment in consumer decision aids.

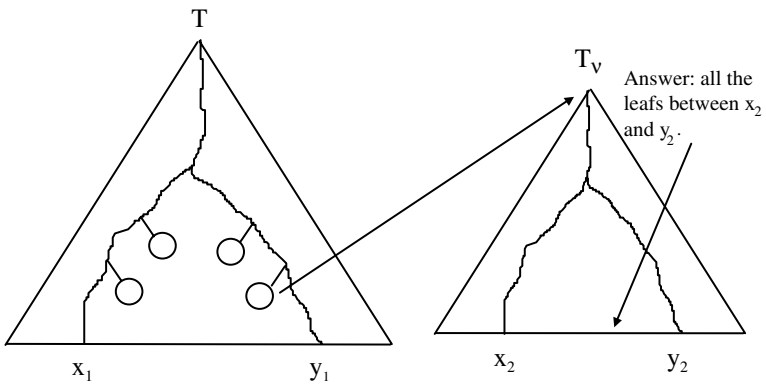

Fig. 2. The two-dimensional range tree.

2.1. The k-means algorithm

The k-means algorithm is a partitional, non-hierarchical data clustering method suitable for classifying large amounts of data into corresponding patterns. It is the simplest and most commonly used algorithm that employs a squared error criterion [2]. Provided with a set of n numeric objects and an integer number $k(k \leqslant n)$, it calculates a partition of patterns in k clusters. This process takes place in an iterative manner starting from a random initial partition and keeping on searching for a partition of n that minimizes the within groups sum of squared errors. The k-means algorithm is analysed in four steps [8]:

1. Choice of k cluster centers to coincide with k randomly chosen patterns or k randomly defined points inside the hyper-volume containing the pattern set.
2. Assignment of each pattern to the closest cluster center (cluster mean).
3. Recalculation of the cluster centers using the current cluster memberships.
4. Computation of the convergence (quality) function. If this is not satisfied then the process is repeated from step 2.

The k-means algorithm is suitable for large sets of numeric objects despite the fact that it is computationally expensive. It is sensitive to the selection of the initial partition and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen. On the other hand, most variants of the k-means algorithms have been proven convergent [14] while some variants like the ISODATA algorithm [3] include a procedure that searches for the best k cluster means at the cost of some performance. More formally, the k-means algorithm tries to minimize the squared error as it is described in function (1):

$$e^2 = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left\| x_i^{(j)} - y_j \right\|^2, \tag{1}$$

where $x_i^{(j)}$ is the ith pattern belonging to the jth cluster and y_j is the center of the jth cluster.

A number of variations to the k-means algorithm have been developed in an effort to improve its computational efficiency or extend its expressiveness in categorical or mixed data. First of all, the ISODATA algorithm [3] used the technique of merging and splitting clusters in order to obtain the optimal partition starting from any arbitrary initial partition, utilizing appropriate threshold values for performing this process. The dynamic clustering algorithm permitted other representations than the center of a cluster utilizing maximum-likelihood estimation, selecting a different criterion function [15]. Other research efforts improved computational complexity by reducing the number of (dis)similarity calculations [1,13]. But two very important steps in the evolution of the k-means algorithm family involve its extension to categorical and mixed numeric and categorical values [7] through the development of the k-modes and k-prototypes algorithms. K-modes uses a simple matching dissimilarity measure to deal with categorical objects while the k-prototypes defines a combined dissimilarity measure, integrating the k-modes and k-means algorithms to allow for clustering of mixed numeric and categorical attributes. This combined dissimilarity measure is described in function (2):

$$e^2 = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \sum_{p \in V_1} \left(x_{i,p}^{(j)} - y_{j,p} \right)^2 + \gamma \sum_{j=1}^{k} \sum_{i=1}^{n_j} \sum_{p \in V_2} \delta\left(x_{i,p}^{(j)}, y_{j,p} \right), \tag{2}$$

where V_1 is the set of the numeric attributes and V_2, is the set of the categorical attributes. Furthermore, the first term is the Euclidean distance measure on the numeric attributes and the second term is the simple

matching dissimilarity measure on the categorical attributes. The weight γ is used to avoid favoring either type of attribute [7]. More specifically the dissimilarity measure is depicted in function (3) and is referred to as simple matching [9]:

$$\delta\left(x_i^{(j)}, y_j\right) = \begin{cases} 0 & \left(x_i^{(j)} = y_j\right), \\ 1 & \left(x_i^{(j)} \neq y_j\right). \end{cases} \tag{3}$$

The computational cost of the k-means algorithm is $O(Tkn)$, where T is the number of iterations, k is the number of clusters and n is the number of data items in the input data set. Although the run-time of both the k-modes and the k-prototypes algorithms appears to increase linearly as both the number of clusters and the number of data items increase, the k-modes algorithm is much faster than both the k-means and the k-prototypes. Finally, all three algorithms are scalable and efficient when clustering very large data complex sets in terms of both the number of data items and the number of clusters [7].

2.2. Multi-dimensional range search

A formal description of the multi-dimensional range search is:
Input: A set S of n data points in the d-dimensional space, so that

$$S = \{(s_1, s_2, \ldots, s_n) | \quad \text{where } s_i \in \Re^d, \text{ and } i, d \in I\}.$$

A d-dimensional rectangle R, defined by a set of two-dimensional points, each one representing a rectangle dimension,

$$R = \{(x_1, y_1), (x_2, y_2), \ldots, (x_d, y_d) | \quad \text{where } x_i, y_i \in \Re, \text{ and } i, d \in I\}.$$

Output: All data points m lying inside the rectangle R.
The range tree [17] was introduced to solve the range-searching problem. This is a multi-level structure since its nodes have pointers to associated structures. The main tree T is then called the first-level tree, and the associated structures are second-level trees. A short explanation of how the range tree answers two-dimensional queries follows [5].

Let W be a set of n points in the plane and $[x_1, y_1]$, $[x_2, y_2]$ the query range in each of the two dimensions, respectively. At first we concentrate on finding the points whose first coordinate lies between x_1 and y_1. To achieve this we have to build a binary search tree on the first coordinate of the points and search with x_1 and $y_1$until we get a node v_1 where the search paths split. From the left child of v_1 we continue the search with x_1 and at every node that the search path goes left we report all points in the right subtree (nodes adjacent to the search path). Similarly, we continue the search with y_1 at the right child of v_1 and at every node that the search path goes right we report all points in the left subtree. Since T is balanced there are at most O(log n) such nodes, where n is the number of the data points. But we are not interested in all these points reported. We only want those whose second coordinate lies in the interval $[x_2, y_2]$. This is another similar query, provided we have a binary search tree on the second coordinate of the points reported from the first search. This leads to the construction of a data structure that has the following properties [5]:

- The main tree is a balanced binary search tree T built on the first coordinate of the points in W.
- Every node or leaf in T stores a pointer to an associated structure T_v which is a balanced binary search tree on the second coordinate of the points in W.

The algorithm description of the two-dimensional range search follows [5]:

Algorithm 2D *RangeQuery*$(T,[x_1,y_1],[x_2,y_2])$
Input: A 2-dimensional range tree T and a range $[x_1,y_1]$, $[x_2,y_2]$.
Output: All points in T that lie in the requested range.
$v_1 \leftarrow$ Find Split Node(T,x_1,y_1)
if v_1 is a leaf
then check if the point stored at v_1 must be reported
else $v \leftarrow$ Left child (v_1)
while v is not a leaf
do if $x_1 \leqslant x_{v1}$
then 1D *RangeQuery* (T_v, (left child(v)), $[x_2,y_2]$)
$v \leftarrow$ left child (v)
else $v \leftarrow$ right child (v)
Check if the point stored at v must be reported.
Similarly, follow the path from right child (v_1) to y_1, call 1D *RangeQuery* with the range $[x_2,y_2]$ on the associated structured of subtrees left of the path, and check if the point stored at the leaf where the path ends must be reported.
End2D *RangeQuery*

The multi-dimensional range tree uses $O(n \log^{d-1} n)$ space and answers the range-searching problem in $O(n \log^{d} n + m)$ time, where m is the number of data points reported. Furthermore, using a portion on the fractional cascading technique the query time is further reduced to $O(n \log^{d-1} n + m)$ [16]. Finally, it must be noted that a d-dimensional range tree is defined recursively from the corresponding tree for the $(d-1)$-dimensional case.

2.3. The k-means range algorithm

The proposed k-means range algorithm is a two-step process involving a multi-dimensional range search followed by a k-means clustering step in the case of numeric data points, or a k-prototypes clustering step in the case of numeric and categorical values.

Therefore a description of the proposed k-means range algorithm is as follows:

input n data points in the d-dimensional space, a d-dimensional rectangle R.
calculate all data points, be it m, lying inside the rectangle using a d-dimensional range tree search.
input k (number of the cluster means)
initialize k-means $y_1, y_2, \ldots, y_k$.
repeat
for each input data point x_i, $1 \leqslant i \leqslant m$
do
assign x_i to the jth cluster with nearest mean y_j, such that the quantity $\left(x_{i,p}^{(j)} - y_{j,p}\right)^2$ or $\gamma\delta\left(x_i^{(j)}, y_j\right)$, depending on the nature of the attribute, is minimum for all j, where $1 \leqslant j \leqslant k$
for each cluster C_j, where $1 \leqslant j \leqslant k$
do recalculate the clustering accuracy $r = \frac{1}{|C_j|}\sum_{y_j \in C_j} y_j$
compute the function $\sum_{j=1}^{k}\sum_{i=1}^{n_j}\sum_{p\in V_1}\left(x_{i,p}^{(j)} - y_{j,p}\right)^2 + \gamma\sum_{j=1}^{k}\sum_{i=1}^{n_j}\sum_{p\in V_2}\delta\left(x_{i,p}^{(j)}, y_{j,p}\right)$
until no data point has changed clusters (or the above quality functions becomes less than a given threshold).

In the above algorithm the clustering accuracy r is defined as the number of the instances y_j occurring in cluster C_j divided by the number of instances in the data set (which contains the m data points reported from the multi-dimensional range search). The time complexity of the k-means range algorithm can be easily calculated by adding the corresponding complexities of its two parts. First the multi-dimensional range search can be solved in $O(n \log^{d-1} n + m)$ time, where m is the size of the answer, as described in the previous section. The k-means part, or more correctly the modified k-prototypes clustering algorithm [7], has a time complexity of $O(Tkn)$, where T is the number of iterations, k is the number of clusters and n is the number of data items in the input data set. But as the range search has produced a significantly smaller set of data points, namely m, which dominates the other two factors T and k, the whole time complexity decreases correspondingly and becomes $O(Tkm)$, depending on a large extent to the nature of the data. Therefore the time complexity of the k-means range algorithm becomes $O(n \log^{d-1} n + Tkm)$ and since $Tk \ll m$, the overall time complexity is $O(n \log^{d-1} n + m)$.

3. Experimental evaluation

In order to evaluate the proposed k-means range algorithm we implemented a system in C++. Using this system, we applied both the k-means, in particular its k-prototypes variant, and the k-means range algorithms in two datasets, namely DS1 and DS2. All the results reported are on a Pentium machine running Linux. The clock speed of the processor is 90 MHz and the memory size is 128 Mbytes.

3.1. Dataset characteristics

The first dataset DS1 consisted of 10.000 movies retrieved from the Imdb movie database (www.imdb.com) where each movie is represented as a record with six attributes (media type, rating, year, category, price, length). The second dataset consisted of 50,000 books retrieved from the Amazon book catalog (www.amazon.com) where each book is represented as a record with six attributes (list price, final price, year, sales rank, customer rating, edition).

For each dataset and the number of clusters we compute the overall execution time of the k-means range algorithm over the k-prototypes, respectively. Since the first part of the k-means range algorithm employs a multi-dimensional range search preprocessing stage, its execution time is strongly dependent on the range of the values defined. Therefore we create two queries where the first covers 50% of the range of a numeric attribute and 50% of the range of a categorical attribute, and the second covers 40% of the ranges of the aforementioned attributes.

The performance of our algorithm is presented in Table 1. For each dataset we present the execution time of the k-prototypes algorithm. Additionally for each dataset and for each query, we present number of data points reported after the range search stage and the total execution time of the k-means range

Table 1
Overall results for 10 iterations

Dataset	k	k-means	k-means range			
			Query 1 (50% of the range)		Query 2 (40% of the range)	
		Processing time (in seconds)	# Points reported	Processing time (in seconds)	# Points reported	Processing time (in seconds)
DS1	64	7.605	2.165	2.256	1.856	9.852
DS2	64	37.904	12.232	10.425	9.783	8.613

algorithm. The results show that our algorithm improves the performance of the k-means algorithm, in the form of the k-prototypes, as it reduces significantly the number of data points reported.

4. Implications

The k-means range algorithm supports the purchase processes of the individual consumer in a straightforward manner. Firstly, it assists the shopper to form his preferences forming multi-attribute arguments, a step that requires special attention in negotiations and deal making [4]. Secondly, it computes the data items requested by the decision maker using a multi-dimensional range search. In the final step it produces the corresponding clusters assisting therefore the consumer to classify his options. The k-means range algorithm is suitable for the development of e-commerce applications since:

- It can be integrated in existing database systems. The range search used multi layered leaf-oriented balanced binary search trees, a heavily addressed and analysed data structure, very common to the data structures used for indexing in relational database management systems.
- The clustering algorithm used, works efficiently both on numeric and categorical data, and also in clustering large data sets.
- It can further be combined with other decision methods which can be built effectively on top of it, allowing the shopper to redefine his criteria and preferences based on the clusters computed. Furthermore the shopper need not reveal his purchasing strategy but use the classification produced to form it.
- The consumer can combine data items from various sources, filter them using the range search and classify them, thus being able to integrate product catalogs from different suppliers.
- It can be used even in distributed asynchronous situations, where agents execute indefinitely, searching for changes or opportunities in product patterns.
- The online store need not keep information regarding the shopper apart from his preference rectangle and final purchase decisions, enhancing in this way ethical factors such as anonymity in purchasing, capturing only changes in user preferences per session.
- In the case of mobile computing, the algorithm described in this paper can be serviced by the online shop server cluster, and the user can retrieve the filtered information in his mobile device through incremental steps, or apply decision-making software to shape his personalized purchasing criteria and preferences.

5. Conclusions

This paper presented a realistic algorithm for personalized clustering in e-commerce applications. This algorithm combines two widely used computational methods capable of manipulating efficiently very large data sets. It allows consumers to dynamically adjust their preferences and combine information from various sources to detect products that are overprized or otherwise dominated by competing alternatives, thus increasing market efficiency. Furthermore, it provides real-time interaction and scalability taking into account the information capacity restrictions of the consumer to assimilate information, refine it and transform it to knowledge. Our experimental results demonstrated that our algorithm can improve significantly the k-means algorithm in the overall time of computation, depending on the range of values requested by the consumer.

In its last part the paper also briefly discussed the implications of this approach both to the consumer and the online store. In the case of the former, it proposed a server sided approach where computational intensive activities are transferred to the intermediary or info-broker computing facilities therefore enforcing the autonomy of the final user. In the case of the latter, it puts forward the thought that since

it does not need to keep personal information on the customer but only his initially stated preference rectangle and his final purchasing decisions. Future research directions involve the development of a Web service which will implement these methods in a transparent way to the shopper and test it in complex decision-making situations existing in online product comparison across multiple vendors and multiple preference criteria.

References

[1] K. Alsabti, S. Ranka, V. Singh, An efficient *k*-means clustering algorithm, in: Proceedings of the First Workshop on High Performance Data Mining, Orlando, Florida, 1995.
[2] M.R. Anderberg, Cluster Analysis for Applications, Academic Press, 1973.
[3] G.H. Ball, D.J. Hall, A clustering technique for summarizing multivariate data, Behavioral Science 12 (1967) 153–155.
[4] C. Beam, A. Segev, M. Bichler, R. Krishnan, On negotiations and deal making in electronic markets, Information System Frontiers 1 (3) (1999) 241–258.
[5] M. De Berg, M. van Kreveld, M. Overmars, O. Swartzkopf, Computational Geometry—Algorithms and Applications, Second ed., Springer, 1999.
[6] G. Haübl, V. Trifts, Consumer decision making in online shopping environments: The effects of interactive shopping aids, Marketing Science 19 (1) (2000) 4–21.
[7] Z. Huang, Extensions to the *k*-means algorithm for clustering large data sets with categorical values, Data Mining and Knowledge Discovery 2 (1998) 283–304.
[8] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: A review, ACM Computing Surveys 31 (3) (1999) 264–323.
[9] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data—An Introduction to Cluster Analysis, Wiley, 1990.
[10] J. McQueen, Some methods for classification and analysis of multivariate observations, in: L.M. Le Cam, J. Newman (Eds.), Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, vol. 1, 1967, pp. 281–297.
[11] G.P. Papamichail, D.P. Papamichail, Towards using computational methods for real-time negotiations in electronic commerce, European Journal of Operational Research 145 (2) (2002) 3–9.
[12] J.W. Payne, J.R. Bettman, E.J. Johnson, The Adaptive Decision Maker, Cambridge University Press, Cambridge, UK, 1993.
[13] V. Ramasubramanian, K. Paliwa, Fast *k*-dimensional tree algorithms for nearest neighbor search with application to vector quantization encoding, IEEE Transactions on Signal Processing 40 (3) (1992) 528–531.
[14] S.Z. Selim, M.A. Ismail, *k*-means-type algorithms: A generalized convergence theorem and characterization of local optimality, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (1) (1984) 81–87.
[15] M.J. Symon, Clustering criterion and multi-variate normal mixture, Biometrics 77 (1997) 35–43.
[16] Y.K. Vaishnavi, Computing point enclosures, IEEE Transactions on Computers, Part C 31 (1) (1982) 22–29.
[17] D. Willard, New data structures for orthogonal range queries, SIAM Journal on Computing 14 (1985) 232–253.