Global k-Means with Similarity Functions[*]

Saúl López-Escobar, J.A. Carrasco-Ochoa, J.Fco. Martínez-Trinidad

National Institute for Astrophysics, Optics and Electronics,
Luis Enrique Erro No.1 Sta. Ma. Tonantzintla, Puebla, México C. P. 72840
{slopez,ariel,fmartine}@inaoep.mx

Abstract. The k-means algorithm is a frequently used algorithm for solving clustering problems. This algorithm has the disadvantage that it depends on the initial conditions, for that reason, the global k-means algorithm was proposed to solve this problem. On the other hand, the k-means algorithm only works with numerical features. This problem is solved by the k-means algorithm with similarity functions that allows working with qualitative and quantitative variables and missing data (mixed and incomplete data). However, this algorithm still depends on the initial conditions. Therefore, in this paper an algorithm to solve the dependency on initial conditions of the k-means algorithm with similarity functions is proposed, our algorithm is tested and compared against k-means algorithm with similarity functions.

1 Introduction

Clustering is a problem that frequently arises in several fields such as pattern recognition, image processing, machine learning, etc. As is well known, this problem consists in to classify a data set in two or more clusters.

The k-means algorithm is a frequently used clustering algorithm that minimizes an objective function, this algorithm assumes that the number of clusters in which the data set will be classified is known. The algorithm consists in the following steps:

1. Randomly select the initial centers.
2. Each object is assigned to the cluster which the distance of its center to the object is minimum.
3. Re-calculate the centers.
4. Repeat steps 2 and 3 until there is not change in the centers.

This algorithm has the disadvantage that it depends on the initial centers, for that reason; usually the algorithm is executed multiple times in order to find a better clustering.

In order to solve the dependency on the initial conditions of the k-means algorithm, the Global *k*-means algorithm was proposed [1], the basic idea underlying this algorithm is that an optimal solution for a clustering problem with *k* clusters can be obtained using a series of local searches using the *k*-means algorithm. At each local search the *k*-1 clusters centers are always initially placed at their optimal position

[*] This work was financially supported by CONACyT (Mexico) through project J38707-A.

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 392 – 399, 2005.

corresponding to the clustering problem with *k*-1 clusters and the remaining center is searched verifying each object in the data set.

On the other hand, the *k*-means algorithm only works with numerical variables due to the use of means for calculating the new centers in each iteration. For that reason, the *k*-means algorithm with similarity functions that allows working with qualitative and quantitative features and missing data was proposed [2], [3]. Problems with this kind of descriptions are very frequent in soft sciences as Medicine, Geology, Sociology, etc. In this kind of descriptions could be not possible to use a distance, only the degree of similarity between objects can be determined, through a similarity function. In this algorithm, the similarity among objects belonging to the same cluster is maximized and the similarity among different clusters is minimized.

As the *k*-means algorithm, the *k*-means with similarity functions depends on the initial conditions, therefore in this paper the global *k*-means with similarity functions is proposed.

This paper is organized as follows: in section 2 the global *k*-means algorithm is described. Section 3 describes the *k*-means with similarity functions algorithm. In section 4 we propose the global k-means with similarity functions algorithm. Experimental results are shown in section 5 and finally section 6 provides conclusions and future work.

2 Global *k*-Means Algorithm

The global *k*-means algorithm was proposed by Aristidis Likas, et al. [1], it constitutes a deterministic effective global clustering algorithm. It does not depend on the initial conditions or any other initial parameter and it uses the *k*-means algorithm as a local search procedure.

Suppose that a data set $X=\{x_1,\ldots,x_n\}$, $x_i\in R^d$ is given and it is required partitioning it in *k* clusters $M_1,\ldots,M_k$ such that the following objective function is optimized:

$$J(m_1,\ldots,m_k)=\sum_{i=1}^{n}\sum_{j=1}^{k}I_j(x_i)\partial(x_i,m_j)^2 \tag{1}$$

This function depends on the cluster centers $m_1,\ldots,m_k$., where

$$I_j(x_i)=\begin{cases}1 & if \quad x_i\in M_j\\ 0 & otherwise\end{cases} \tag{2}$$

and

$$\partial(x_i,m_j)=\left\|x_i-m_j\right\| \tag{3}$$

To solve a problem with *k*-clusters we start with one cluster (*k'*=1) and find its optimal position as the center of the data set. In order to solve the problem with two clusters (*k'*=2) the first center is placed at the optimal position for the problem with *k'*=1 and the *k*-means algorithm is executed *n* times placing the second center at each object x_i of the data set, x_i must be different to the solution for the problem with one cluster, i=1,…,*n*. After the *n* executions of the *k*-means algorithm we consider the solution for the clustering problem with k'=2 as the solution that minimizes the objective function (1). In general, let ($m_1^*(k\text{-}1),\ldots,m_{k-1}^*(k\text{-}1)$) denote the solution for the problem with (k-1)-clusters. Once the solution for the problem of finding (k-1)-clusters is obtained, this is used to solve the problem with k-clusters executing n times the k-means algorithm where each execution starts with the initial centers: ($m_1^*(k\text{-}1),\ldots,m_{k\text{-}1}^*(k\text{-}1),x_i$), $x_i \neq m_p^*(k\text{-}1)$, $p=1,\ldots,k\text{-}1$, $i=1,\ldots,n$. The best solution (which minimizes the objective function (1)) after the n executions is considered as the solution for the problem with k-clusters.

3 *k*-Means with Similarity Functions

The k-means with similarity functions algorithm was proposed by Martínez-Trinidad, et al in [2], [3]. It follows the same idea that the k-means algorithm but instead of using a distance for comparing objects, a similarity function is used.

Suppose that a data set $X=\{x_1,\ldots,x_n\}$ is given, where each object is described by a set $R=\{y_1,\ldots,y_s\}$ of features. Each feature takes values in a set of admissible values D_i, $y_i(x_j) \in D_i$ $i=1,\ldots,s$. We assume that in D_i there is a symbol "?" to denote missing data. Thus, the features can be of any nature (qualitative: boolean, multi-valued, etc. or quantitative: integer, real) and incomplete descriptions of objects are considered. A similarity function $\Gamma:(D_1 \times D_2 \times \ldots \times D_s)^2 \rightarrow [0,1]$, which allows comparing objects is defined. In this work, the similarity function used is:

$$\Gamma(x_i,x_j)=\frac{\left|\left\{y_k \mid C\left(y_k(x_i),y_k(x_j)\right)=1\right\}\right|}{s} \tag{4}$$

where C is a comparison function between features values.

We require partitioning the data set in k clusters $M_1,\ldots M_k$. In this kind of problems, it could be impossible to calculate means; so objects from the sample, called representative objects x_j^r, are used as centers of the clusters M_j, $j=1,\ldots,k$.

The data set must be classified according the representative objects of each cluster, i.e., given a set of representative objects, first we obtain the membership $I_j(x_i)$ of the object x_i to cluster M_j, after that, we calculate the representative objects for the new k-partition, this procedure is repeated until there is no change in the representative objects.

So, the objective function is:

$$J(x_1^r,\ldots,x_k^r)=\sum_{j=1}^{k}\sum_{i=1}^{n} I_j(x_i)\Gamma(x_j^r,x_i) \tag{5}$$

where

$$I_j(x_i)=\begin{cases}1 & if \quad \Gamma(x_j^r,x_i)=\max\limits_{1\le q\le k}\left\{\Gamma(x_q^r,x_i)\right\}\\ 0 & otherwise\end{cases} \tag{6}$$

That is, an object x_i will be assigned to the cluster such that x_i is the most similar with their representative objects.

In this case, the objective is to maximize this function.

To determine the representative objects the next expressions are used:

$$r_{M_j}(x_i) = \frac{\beta_{M_j}(x_i)}{(\alpha_{M_j}(x_i) + (1 - \beta_{M_j}(x_i)))} + \eta_{M_q}(x_i) \tag{7}$$

where $x_i \in M_j$ and $q=1,\ldots,k$ $q \neq j$

$$\beta_{M_j}(x_i) = \frac{1}{\left|M_j\right| - 1} \sum_{\substack{x_i, x_q \in M_j \\ x_i \neq x_q}} \Gamma(x_i, x_q) \tag{8}$$

$$\alpha_{M_j}(x_i) = \frac{1}{\left|M_j\right| - 1} \sum_{\substack{x_i, x_q \in M_j \\ x_i \neq x_q}} \left|\beta_{M_j}(x_i) - \Gamma(x_i, x_q)\right| \tag{9}$$

and

$$\eta_{M_k}(x_i) = \sum_{\substack{q=1 \\ i \neq q}}^{k} \left(1 - \Gamma(x_q^r, x_i)\right) \tag{10}$$

The representative object for the cluster M_j is defined as the object x_r which yields the maximum of $r_{M_j}(x_i)$

$$r_{M_j}(x_r) = \max_{x_p \in M_j} \left\{ r_{M_j}(x_p) \right\} \tag{11}$$

4 Global *k*-Means with Similarity Functions Algorithm

The global *k*-means algorithm solves the dependency on the initial conditions of the *k*-means algorithm, but only works with numerical features, therefore we propose an extension to the global *k*-means such that it allows working with mixed and incomplete data.

We consider a problem as the described in section 3. Our algorithm follows the same methodology that the global *k*-means algorithm with the difference that instead using *k*-means algorithm as local search procedure, the *k*-means with similarity functions is used, so it is guaranteed that the obtained centers belong to the data set.

In order to solve a problem with *k*-clusters we start with one cluster (*k'*=1) and we find its optimal position as the representative object of the data set, this is made by finding the object which is the most similar to all the objects of data set. In order to solve the problem with two clusters (*k'*=2) the first center is placed at the optimal position for the problem with *k'*=1 (let x_1^{r*} be the representative object for *k'*=1) and the *k*-means with similarity functions algorithm is executed *n*-1 times placing the second center at each object x_i of the data set, $x_i \neq x_1^{r*}$, $i=1,\ldots,n$. After the *n*-1 executions of the *k*-means with similarity functions algorithm, we consider the solution for the clustering problem with *k'* = 2 as the solution that maximizes the error function (5). In general, let ($x_1^{r*}(k-1), x_2^{r*}(k-1),\ldots, x_{k-1}^{r*}(k-1)$) denote the solution for the problem with (k-1) –clusters. Once the solution for the problem with (k-1)-clusters is obtained this is used to solve the problem with k-clusters executing n-(k-1) times the k-means with similarity functions algorithm where each execution starts with the initial centers: ($x_1^{r*}(k-1), x_2^{r*}(k-1),\ldots, x_{k-1}^{r*}(k-1)$,$x_i$), $x_i \neq x_p^{r*}$, p=1,…,k-1, i=1,…,n. The best solution after the n-(k-1) executions (which maximizes the error function (5)) is considered as the solution for the problem with k-clusters. The proposed algorithm is depicted in Table 1.

Table 1. Global k-means with similarity functions algorithm

```
Input: k = number of clusters
       n = number of objects of the data set
Output: RO [1,…,k] /* Representative Objects */
        OF /* Value of the objective function */
Count = 0
Seeds [1,…,k] = 0
Seeds[1] = most similar object to the data set
for k'=2 to k
   for i=1 to n
      if i ≠ Seeds[1,…,k'-1]
        [SRO,J] = KMeansWithSimilarityFunctions (Seeds[1,…,k'-1],i)
        /* SRO is the set of representative objects */
        /* J is the objective function */
        if J>count then
            count = J
            Seeds = SRO
RO = Seeds
OF = count
```

5 Experimental Results

We have tested the proposed algorithm on several data sets: Iris, Flags, Electro, Machine and Wine [4]. In all data sets we did experiments considering only information of the feature vector and ignoring class labels. The quality of the obtained solutions was evaluated in terms of the objective function (5). The description of each data set is given in Table 2.

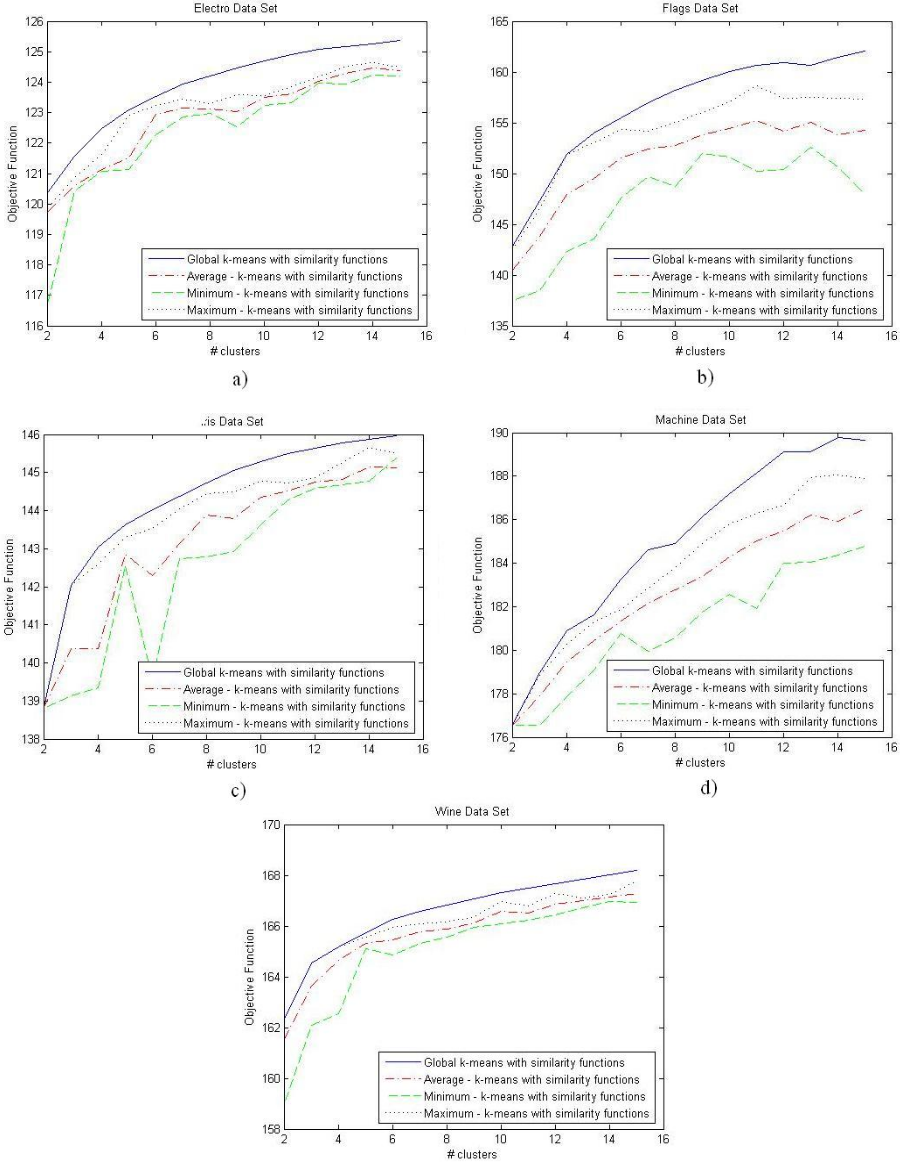
Table 2. Data set features

Data set	Objects	Qualitative features	Quantitative features
Iris	150	0	4
Flags	194	3	26
Electro	132	0	11
Machine	209	1	7
Wine	178	0	13

For each data set we did the following experiments:

- One run of the global k-means with similarity functions algorithm for the problem with k=2,…,15.

- *n* runs (where *n* is the number of objects of the data set) of the *k*-means with similarity functions algorithm for each problem with *k*=2,…,15 starting with random initial centers. For each data set, the average, the maximum and the minimum of the objective function were calculated.



a)

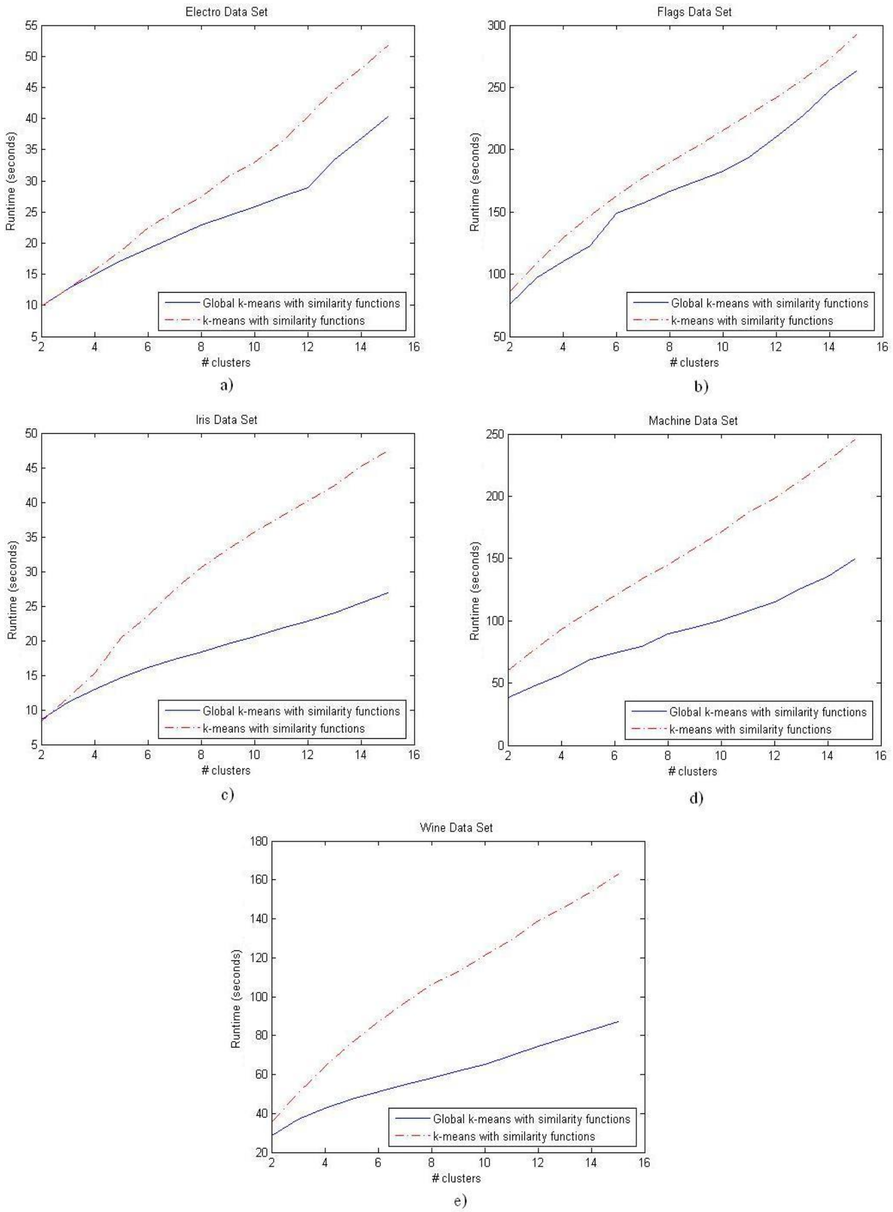
b)

c)

d)

e)

Fig. 1. Experimental results for data sets: a) Electro, b) Flags, c) Iris, d) Machine and e) Wine



a)

b)

c)

d)

e)

Fig. 2. Runtime for data sets: a) Electro, b) Flags, c) Iris, d) Machine and e) Wine

In figure 1 the value of the objective function obtained from the global *k*-means algorithm with similarity functions is compared against the average, the maximum and the minimum of the *n* values obtained from the runs of the *k*-means with similarity functions algorithm. In our experiments, the Global k-means with similarity functions algorithm obtained better results than the k-means with similarity functions algorithm and in few cases it obtains the same result that the maximum.

In figure 2 the runtime of each experiment is shown. The runtime of the Global *k*-means algorithm with similarity functions is less than the runtime of the *k*-means algorithm. This is due because for each value of *k* we carried out *n* runs of the *k*-means with similarity functions algorithm, and the global *k*-means with similarity functions execute only *n*-(*k*-*1*) runs of the *k*-means with similarity functions algorithm. Also, each time the global *k*-means with similarity functions algorithm uses the *k*-means with similarity functions, it starts with better seeds than the random selection, therefore, it converges faster.

6 Conclusions

In this paper the global *k*-means with similarity functions algorithm was introduced. Our method is independent of the initial conditions. It was compared against the *k*-means with similarity functions algorithm.

In our experiments, the global k-means with similarity functions algorithm obtained better clusters in terms of the objective function than the k-means with similarity functions, and only in a few cases, with small values for *k*, the results were the same that the maximum obtained with the *k*-means with similarity functions.

The runtimes of our algorithm were less than the time needed for the *n* executions of the *k*-means with similarity functions algorithm, and our algorithm's results were better.

As future work, we are going to find a fast global *k*-means with similarity functions algorithm in order to reduce the computational cost without significantly affecting the quality.

References

1. Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek, "The global *k*-means clustering algorithm", *Pattern Recognition 36*, 2003, pp. 451-461.
2. José Francisco Martínez Trinidad, Javier Raymundo García Serrano, and Irene Olaya Ayaquica Martínez, "C-Means Algorithm with Similarity Functions", Computación y Sistemas Vol. 5 No. 4, 2002, pp. 241-246
3. Javier R. García Serrano and J. F. Martínez-Trinidad, "Extension to c-means algorithm for the use of similarity functions", 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases Proceedings. Prague, Czech Rep. (1999). pp. 354-359.
4. C.L. Blake, C. J. Merz, UCI repository of machine learning databases, University of California, Irvine, Departament of Information and Computer Sciences, 1998.