A New Clustering Algorithm Based on K-Means Using a Line Segment as Prototype

Juan Carlos Rojas Thomas

Universidad de Atacama, Copiapó, Chile
juancarlos.rojas@uda.cl

Abstract. This project shows the development of a new clustering algorithm, based on *k-means*, which faces its problems with clusters of differences variances. This new algorithm uses a line segment as prototype which captures the axis that presents the biggest variance of the cluster. The line segment adjusts iteratively its long and direction as the data are classified. To perform the classification, a border region that determines approximately the limit on the cluster is built based on geometric model, which depends on the central line segment. The data are classified later according to their proximity to the different border regions. The process is repeated until the parameters of the all border regions associated with each cluster remain constant.

Keywords: Clustering, Kmeans, Variance, Central Line Segment, Border Region.

1 Introduction

The process of clustering consists on classifying in an unsupervised way a set of patterns (observations or data) into groups (clusters) [1]. There are many types of clustering algorithms. One of these is the center based algorithms. Compared with the others types of clustering algorithms, the center based algorithms are very efficient with big data bases and with high dimensional data. Usually, these algorithms try to minimize an objective function, which defines how good is the solution obtained [2].

1.1 K-Means

The *k-means* is a clustering algorithm which is considered a center based algorithm. This algorithm tries to find the k partitions that minimize the objective function. The objective function used by this algorithm is the mean square error [3]. This criterion, where m_i corresponds to the mean of the cluster C_i, n to the total number of objects, and k to the total number of clusters, is defined as [4]:

$$E = \frac{1}{n} \sum_{i=1}^{k} \sum_{x \in c_i} \left\| x - m_i \right\|^2 \tag{1}$$

In general the *k-means* algorithm performs the classification of the data according to a measure of distance to certain points considered the centers of the clusters in a

C. San Martin and S.-W. Kim (Eds.): CIARP 2011, LNCS 7042, pp. 638–645, 2011.

specific space of features. These points, called centroids, are initialized at the beginning of the algorithm, as well as the measure of distance, and the subsequent classification is performed according to proximity to those. Then, after the classification process is completed, the centroids are recalculated as the means of each cluster. Then, the data are reclassified, and the process is repeated until the centroids remain constant [5] [6].

1.2 Advantages and Disadvantages

The advantages of *k-means* are its velocity and its easy application in high dimensional spaces. However it has some disadvantages: the algorithm is applicable only if the mean is defined, the k number of clusters has to be estimated, often converges to a local optimum, and the final result depends on the initial values assigned to the centroids [3]. On the other hand, the criterion of the mean square error works well when the clusters are compact clouds well separated. However, when the differences in size of the geometry of the clusters are very big, the use of this criterion could divide the larger clusters [4].

2 Related Works

A lot of works have been made trying to overcome the disadvantages of this algorithm. However, the most of them are focused to resolve the estimation of the parameter k (the number of clusters) [7] [8], optimize the convergence speed to the solution [9] [10], the extension of the algorithm to ordinals sets [11], and to determine the initials coordinates of the centroids [12] [13].

Concerning the treatment of clusters of different sizes, the only work founded in the literature is [3]. This algorithm is a modification of *k-means*, whose objective is only to detect clusters with circular shapes.

3 Proposal

The algorithm proposed in this document confronts the limitation of k-means when it is used over clusters of very dissimilar variances, using a line segment as prototype. This new algorithm, as well as the original, can be used in space of high dimensions.

3.1 General Scheme

The inputs that the algorithm receives are the data set, the initial centroids and the number of clusters to detect. Then, the algorithm starts to adjust iteratively the parameters that determine the border regions associated with each cluster and used to capture their variances. This process consists on classifying the data according to their proximity to the different border regions, and then update their parameters. The process is repeated until the parameters of the all border regions associated with each cluster remain constants.

3.2 General Algorithm

```
Input: dataset, number of clusters, initial centroids
Begin
  Repeat
    Classify Data
    Calculate the Parameters of each Border Region
  Until the Parameters of the Regions Associated Remain
  Constant
End
```

3.3 Geometric Model

The geometric model which this algorithm uses is defined by geometrics shapes which border region is made by all points that are equidistant from the same central line segment. This distance is called "radius". Then, the parameters that determine the border region are the direction and length of the central line segment (specified by the coordinates of its extremes) and the radius of the figure. In two dimensions this model generates a rectangle with semicircles in its extremes, in three dimensions generates a cylinder with semispherical caps in its extremes. For simplicity the shape generated will be called "cylinder", independent of the dimensions considered. The figure 1 shows this concept.

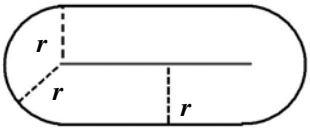

Fig. 1. The figure shows the geometric shape that is built in two dimensions according to the geometric model used by the algorithm. It is possible to note the central segment and the points that localized to a distance *r* (radius) made the border region.

3.4 Classification

This algorithm performs the classification founded the "cylinders" that best represent the data distribution of the data detected. The parameters that specify a cylinder are the direction and length of the central line segment and its radius. At the beginning of the process the radius are established to zero, and the central axis of each cylinders corresponds to the centroids of the clusters given as initial input to the algorithm, so in the first iteration the classification is performed according to the proximity of the data to the initials centroids. Then, the direction and length of each central line segment associated with the clusters are calculated, with the centroid as the midpoint, and finally the values of the radius of each figure are obtained. Once the values of parameters have been obtained, the data are reclassified. This classification depends on which of the following three situations is each datum, as the figure 2 illustrates:

a) The datum is not contained inside any cylinder: then the data is assigned to the cluster associated with the nearest cylinder.
b) The datum is contained inside only one cylinder: then the datum is assigned to the cluster associated with this cylinder.
c) The datum is contained inside more than one cylinder: then the datum is assigned to the cluster associated with the cylinder whose central segment is the nearest, among the cylinders that contain the datum.

Then, the process to calculate the central segments and radius is repeated, with the subsequence reclassification, until these remain constant.

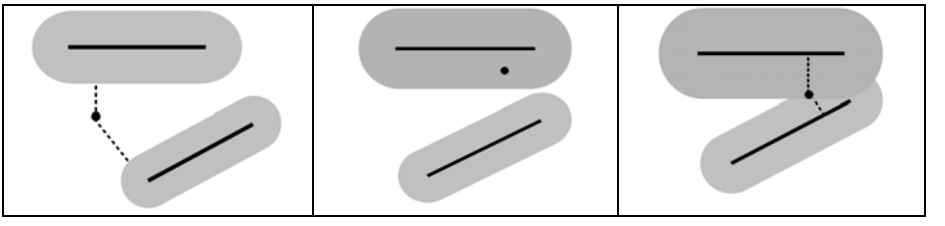

Fig. 2. The images show, from left to right, the situation of a datum extern from two cylinders, with the distances to both marked with dotted lines, the situation of a datum localized inside one cylinder, and the situation of a datum located inside two cylinders that intersects, with the distances to the central segments of both marked with dotted lines.

3.5 Obtaining the Central Line Segment

The generation of the cylinder is based on obtaining a line segment which corresponds to its central axis. It is obtained using the principal component analysis over the data set associated with the cluster, and extracting the vector that represents the component which captures the biggest variance of the data set. Then, a line segment is built, which corresponds to the central axis of the cylinder. This axis is aligned with the direction of the vector just calculated, centered in the centroid of the cluster. The length of this central axis is obtained calculating first the absolute magnitudes of the vectors projections associated to each datum, considering the centroid as the origin, over the line determined by the vector of biggest variance and the centroid. Then, the mean of these values is calculated, and the central axis length is specified, finally, as the double of the mean just calculated. This process is illustrated by the figure 3. Let d_{ij} be the vector associated with the *i-th* datum of the cluster *j*, v_j the principal component with biggest variance of cluster *j* (assumed with unitary magnitude), n_j the quantity of data of cluster *j*. Then, the length of central axis of cluster *j*, l_j, is given by the next formula (the black circle represents the inner product):

$$l_j = 2 * \frac{\sum_{i=1}^{n_j} \left| \vec{d}_{ij} \bullet \vec{v}_j \right|}{n_j} \tag{2}$$

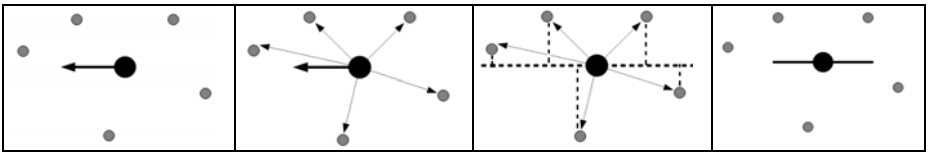

Fig. 3. The images show, from left to right, the generation of the central segment of one cluster (the gray circles correspond to data and the black circle to the centroid). First it is determined the principal component of the biggest variance (black arrow), and then the vectors associated with each datum, their projections over the line, and finally the building of the segment with the centroid as the midpoint.

3.6 The Calculus of the Radio

The radio of a cylinder is obtained calculating the mean of the distances of data to the central segment. Let x_{ij} be the *i-th* datum of the cluster *j*, s_j the central segment of the cluster *j*, $d(x_{ij}, s_j)$ the distance between the *i-th* datum and the segment of the cluster *j*, n_j the quantity of data of cluster *j*. Then, the radio associated with cluster *j*, r_j, is given by the next formula:

$$r_j = \frac{\sum_{i=1}^{n_j} d\left(x_{ij}, s_j\right)}{n_j} \tag{3}$$

3.7 Distance between a Datum and a Central Segment

This distance is defined as the length of the shortest line segment that connects a datum with some point of the cylinder central segment. To allow the algorithm be extensible to high dimensions the theorem of cosine is used to obtain these distances, to bring the calculation to a two dimensional plane. This procedure consists in generating a triangle whose vertices are the initial and final points of the cylinder central segment, and the datum. Then, the angles of the triangle are calculated using the cosine theorem. If all angles of the triangle are less than 90 degrees, then the distance between the datum and the central segment corresponds to the height of the triangle, which can be easily calculated. If one of the angles is greater than 90, then the distance between the datum and the two extremes points of the cylinder central axis. The lower value corresponds to the distance between the datum and the axis. Both situations are illustrated by the figure 4.

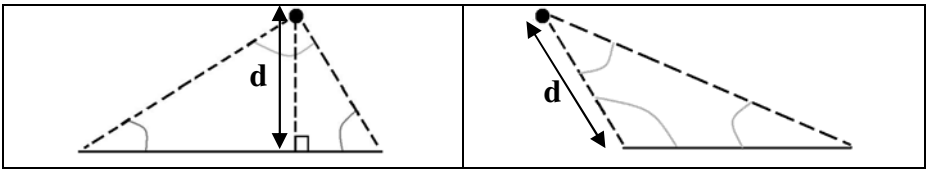

Fig. 4. The images show the two different types of triangulations that can be formed with a datum (black circle) and the line segment (continuous line), and how these are used to obtain the distance between them

3.8 Calculating the Distance between a Datum and a Cylinder

Previous to calculate the distance between a datum and a cylinder, distance between the datum and the cylinder's central segment is calculated. If this distance is less than the cylinder radius, then the datum is considered contained inside the inner space of the cylinder. If this distance is greater than the radius, the datum is considered extern to the cylinder, and then the distance is calculated as the difference between the distance to the central segment just calculated and the cylinder radius. The figure 5 shows this situation. Let x be a datum, v a cylinder, r the cylinder radius, s the cylinder central segment, and d the Euclidian distance, then the distance between a datum and a cylinder is specified as:

$$d(x,v) = d(x,s) - r \tag{4}$$

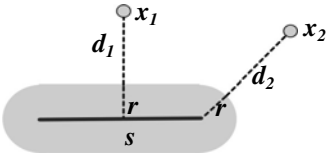

Fig. 5. The figure shows how the distances are obtained from two points (small circles) and a cylinder. The dotted lines which run from each point to the central segment s of the cylinder represent the distances from them to the segment. By subtracting the magnitude of the cylinder radius r the distances from the points to the cylinder border, d_1 and d_2, are obtained.

4 Experimental Results

The performance of the algorithm proposed was compared with the *k-means* in a series of tests. The data were generated artificially with Gaussian distribution. These tests were designed so that, from an initial configuration of clusters with similar variances, it was increasing gradually the ration between the clusters variances along one axis, as the figure 6 illustrates. To evaluate the performance of both algorithm the Rand index was used, which allows to measure the level of similitude between two partitions, with values ranging from zero (minimal similitude) to one (maximal similitude) [2].

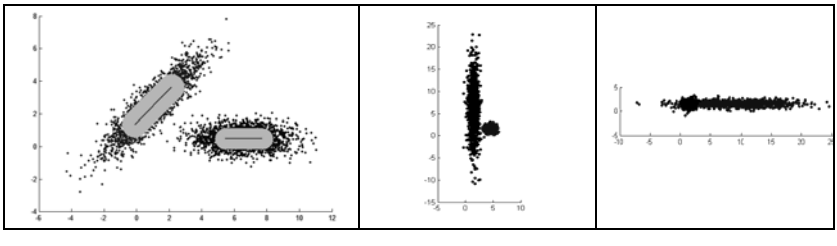

Fig. 6. The image of the left shows, superimposed on two clusters, the cylinders and their central axes after having been applied the algorithm. The next images show the most extremes configurations used in the tests, with a ratio of 1/10 between the variances along y axis (central image) and x axis (right image).

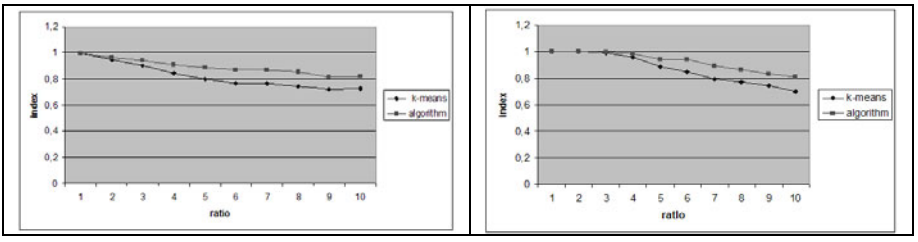

Fig. 7. The graphics illustrate the comparative performance when the ratio between the variances of the two clusters is gradually increased, indicating the values delivered by the Rand index v/s the ratio between variances. The left graphic shows the test series where the ratio is increased along the "y" axis, and the right shows the test series where the ratio is increased along the "x" axis.

The results show that, as the differences between the variances increase, the performance of the algorithm begins to overcome the *k-means*, as the graphics of the figure 7 illustrate.

5 Conclusions and Future Works

The algorithm demonstrated that it improved remarkably the performance of the *k-means* in the situations where the clusters have many different variances. In the situations where the variances are not so different, the performance is similar to *k-means*. However, it is still required a later research, so adding another criteria, or improving the ones that have been used, in the process of building the border region. It is also an option to define other border regions that allow a more accurate capture of the variances.

References

1. Jain, A.K., Murty, M.N., Flynn, O.J.: Data Clustering: a review. ACM Computing Surveys 31(3) (September 1999)
2. Gan, G., Ma, C., Wu, J.: Data Clustering Theory, algorithms and applications. SIAM, Society for Industrial and Applied Mathematics (May 30, 2007)
3. Fahim, M., Saake, G., Salem, A.M., Torkey, F.A., Ramadan, M.A.: K-Means for Spherical Clusters with Large Variance in Sizes. In: Proceedings of World Academy of Science, Engineering and Technology, Paris, vol. 35, pp. 177–182 (November 2008) ISSN 2070-3740
4. Guha, S., Rastogi, R., Shim, K.: CURE: An Efficient Clustering Algorithms for Large Databases. In: Proc. ACM SIGMOD Int. Conf. on Management of Data, Seattle, WA, pp. 73–84 (1998)
5. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
6. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 3rd edn. (2006)
7. Pham, D.T., Dimov, S.S., Nguyen, C.D.: Selection of k in K-means clustering. Mechanical Engineering Science 219, 103–119 (2004)

8. Pelleg, D., Moore, A.: x-means: Extending k-means with efficient estimation of the number of clusters. In: Proceedings of Seventeenth International Conference on Machine Learning, pp. 727–734. Morgan Kaufmann, San Francisco (2000)
9. Faber, V.: Clustering and the continuous k-means algorithm. Los Alamos Science 22, 138–144 (1994)
10. Phillips, S.: Acceleration of K-means and Related Clustering Algorithms. In: Mount, D.M., Stein, C. (eds.) ALENEX 2002. LNCS, vol. 2409, pp. 166–177. Springer, Heidelberg (2002)
11. Huang, Z.: Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. Data Minig and Knowledge Discovery 2(3), 283–304 (1998)
12. Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. In: Proceedings of the 15th International Conference on Machine Learning, pp. 91–99. Morgan Kaufmann, San Francisco (1998)
13. Deelers, S., Auwatanamongkol, S.: Enhancing K-Means Algorithm with Initial Cluster Centers Derived from Data Partitioning along the Data Axis with the Highest Variance. PWASET 26, 323–328 (2007)