Cluster medians

Luis Rojo-González[a,b]

[a] *Industrial Engineering Department, Universidad de Santiago de Chile, Chile*
[b] *Universitat Politècnica de Catalunya, Spain*

Abstract

In this work, we show a mathematical model for clustering context and the obtained results are compared performing two heuristic algorithms: i) $k-means$ and ii) Minimum spanning tree. The comparison is based on highly consumption of computational resources which means high running times using traditional ways (i.e. solvers) to find the optimal solution. On the other hand, are the heuristics which, despite they are not insure the optimality, work very fast and obtained results are close than the optimal solution. On this case, $k-means$ shows a 90.4% and Minimum spanning tree of 91.4% of efficacy within reduced running times. Whereas the mathematical model solves different instances on an average time of 42.98 (s) and a maximum of 231.55 (s) with an exponential behavior. [1]

1. Introduction

The *k-medians* clustering problem [3] is a cluster analysis algorithm, where within a dataset of n-observations we will consider k of these as centroids such that the 1-norm distances among whole observations to their respectively centroid k is minimum. So, it is clearly an optimization problem which is NP-complete [5, 10].

On this way, several studies had developed approximation algorithms to solve this in a polynomial time [8, 7]. The most used algorithm is based on Lloyd-style iteration which alternates between an expectation (E) and maximization (M) step, making this an Expectation–maximization algorithm. In the E step, all objects are assigned to their nearest median. In the M step, the medians are recomputed by using the median in each single dimension.

In this work, the formulation of the optimization problem is shown considering different dataset sizes and different number of clusters. Finally, obtained results are compared with the minimum spanning tree (MST) for clustering algorithm.

2. Statement of the problem

Let $\mathcal{I}$ be a set of observations and $\mathcal{J}$ be a set of attributes. Thus, $\mathcal{X}$ defines a matrix of $i \times j$-dimension $\forall i \in \mathcal{I}, j \in \mathcal{J}$. So, we can consider a 1-norm distance matrix among observations denoted as $d_{i,k} = \sum_j^{|\mathcal{J}|} |x_{i,j} - x_{k,j}|$, where $x_{i,j}$ is the attribute j of the observation i, $\forall j \in \mathcal{J}, i, k \in \mathcal{I}$.

Also, we consider $p \in \mathcal{P}$ centroids with a distance $d_{i,p}$ to each observation $i \in \mathcal{I}$. In particular for this problem, we consider a problem where these centroid p are one of the observations i such that the distance $d_{i,p}$ is equal to the distance $d_{i,k}$ defined above.

Email address: luis.rojo.g@usach.cl (Luis Rojo-González)

[1] You can find the dataset amongst other in https://drive.google.com/drive/folders/12rO1WJjAaaYk1g3t0302rNqu86kY9OKI?usp=sharing

3. Binary integer linear programming model (BILP)

Let $z_{i,p}$ be a binary variable that is 1 if the observation i belongs to cluster p, $\forall i \in \mathcal{I}, p \in \mathcal{P}$. Thus, we propose an optimization problem such that we find those clusters p which minimizes the total distance such as follows:

$$min \quad \sum_i \sum_p z_{i,p} d_{i,p} \tag{1}$$

$$s.t. \quad \sum_p z_{i,p} = 1 \qquad \forall i \in \mathcal{I} \tag{2}$$

$$\sum_i z_{i,i} = |\mathcal{P}| \tag{3}$$

$$z_{p,p} \geq z_{i,p} \qquad \forall i \in \mathcal{I}, p \in \mathcal{P} \tag{4}$$

4. Minimum spanning tree (MST)

A spanning tree is an acyclic subgraph of a graph G, which contains all the vertices from G. The minimum spanning tree (MST) of a weighted graph is the minimum weight spanning tree of that graph. With the classical MST algorithms [11, 9], the cost of constructing a minimum spanning tree is $\mathcal{O}(mlogn)$, where m is the number of edges in the graph, n is the number of vertices [6].

Once the MST is built for a given input, there are two different ways to produce a group of clusters. If the number of clusters k is given in advance, the simplest way to obtain k clusters is to sort the edges of the minimum spanning tree in descending order of their weights, and remove the edges with the first $k-1$ heaviest weights [1] . This can be obtained through R package *mstknnclust* [4].

5. Case of study

The case of study is about a case which requires to develop a customer segmentation to define marketing strategy, available in Kaggle website [2]. The sample Dataset summarizes the usage behavior of about 9000 active credit card holders during 6 months. The file is at a customer level with 18 behavioral variables of which we consider the next 4:

Balance: Balance amount left in their account to make purchases.

Credit limit: Limit of Credit Card for user.

Payments: Amount of Payment done by user.

Minimum payments: Minimum amount of payments made by user.

6. Computational experiments

The computational experiments consider instances related to different dataset size and number of clusters. The $k-means$ algorithm as well as MST algorithm was implemented in R using packages *cluster* [4]. For $k-means$ algorithm was used *Lloyd*-algorithm, 1000 iterations and 10 starting random subsets.

The implementation of the mathematical model was in AMPL version 20180110 and was solved using solver CPLEX 12.8.0.0 for integer programming. It was executed in a Macbook Air Intel Core i5, 1.6 Ghz, 8GB RAM.

The running time of the script was 1096.53 (s) to complete all iterations such as Table 1 shows considering combinations of data sizes and clusters.

Table 1: Resolution time (s) for BILP model (only those instances which considers 300 observations or more are shown).

	Data size (N)	K	time (s)
13	300	14	21.6531
14	300	15	20.6125
15	300	16	17.7855
16	350	14	29.7135
17	350	15	28.3866
18	350	16	28.9601
19	400	14	55.4809
20	400	15	54.0797
21	400	16	50.9207
22	450	14	104.1293
23	450	15	98.2445
24	450	16	92.2299
25	500	14	231.5455
26	500	15	115.2821
27	500	16	139.0939

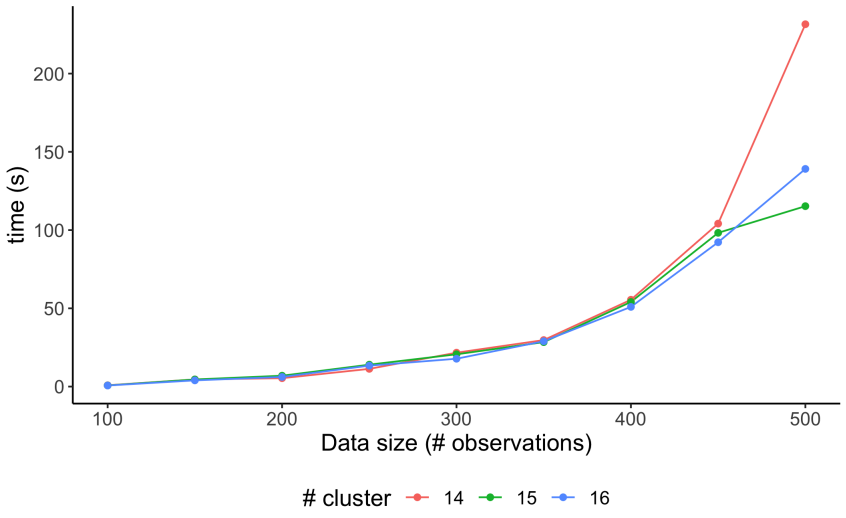

Figure 1: Resolution time (s) of BILP model.

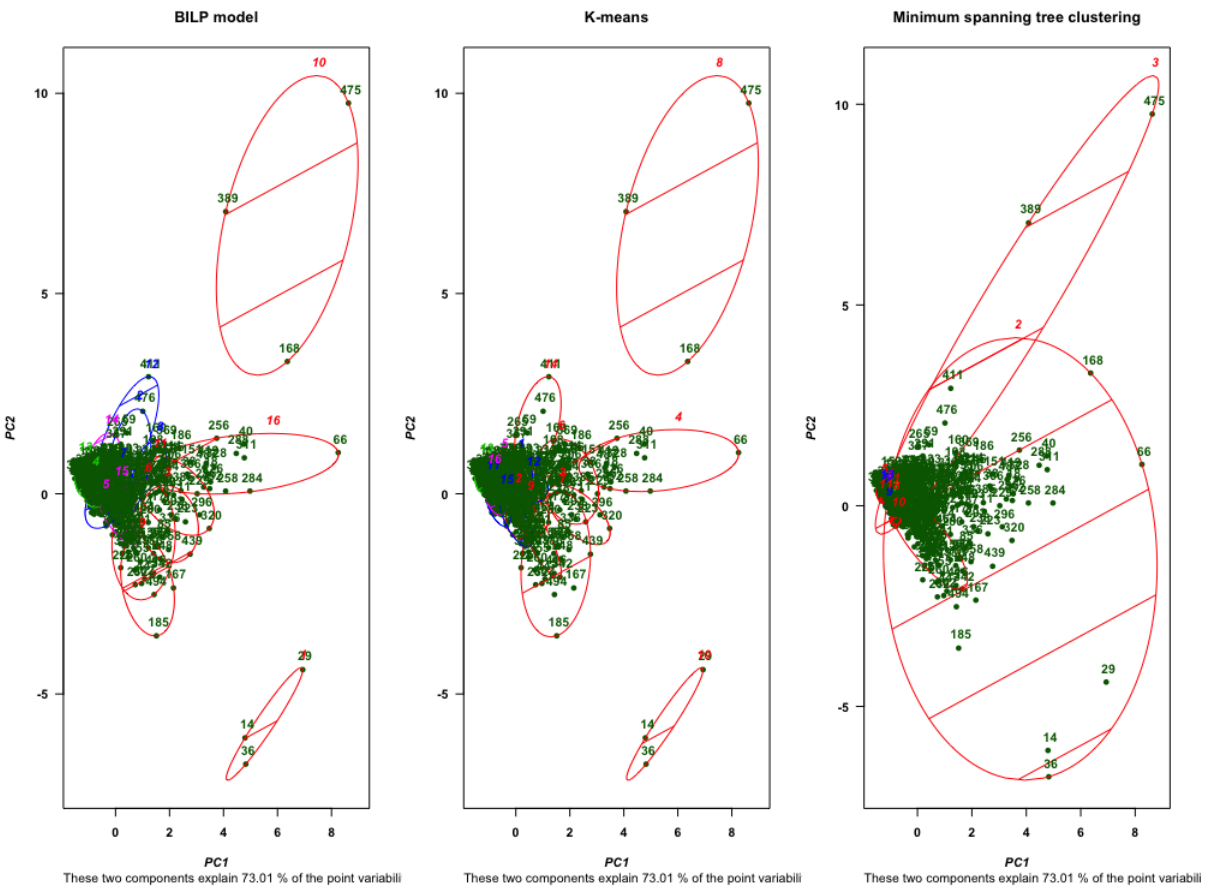

Figure 2: Obtained results by using two first principal components for $N = 500$ and $k = 16$.

6.1. Efficacy

Once mathematical problem is solved, a couple of ways to overcome the high consumption of computational resources are review according to obtained running times shown above, which has an exponential increasing as dataset size increase. Because of that, two approximation ways are considered: i) K-means and ii) Minimum spanning tree clustering such as we stated above. Figure 2 shows obtained clustering using two first principal components when $k = 16$ by each resolution method; but with one (big) difference: **running time**.

Although first two principal components can explain a big fraction of the variance, it is difficult to see important differences among each plot. Nevertheless, according to obtained results there is an efficacy of 90.4% and 91.6% for $k - means$ package and MST package, respectively. On the other hand, running times are quite low, so measuring them is not considered.

7. Conclusion

The comparison between BILP and $k - means$ package shows a 90.4% of efficacy which is an important ratio because of the difference on their running times, which is a clear advantage of the heuristic due to its performance is quite close to optimal solution with reduced running time and low computational cost. The same case occurs with MST clustering, where efficacy is even greater than $k - means$; even although they are very close from each other. Nevertheless any gain of performance is very important given that dataset sizes are very large the most part of the time.

References

[1] T. Asano, B. Bhattacharya, M. Keil, and F. Yao. Clustering algorithms based on minimum and maximum spanning trees. In *Proceedings of the fourth annual symposium on Computational geometry*, pages 252–257. ACM, 1988.

[2] A. Bhasin. Credit Card Dataset for Clustering. `https://www.kaggle.com/arjunbhasin2013/ccdata/downloads/ccdata.zip/1/`, 2018. [Online; accessed 28-September-2019].

[3] P. S. Bradley, O. L. Mangasarian, and W. N. Street. Clustering via concave minimization. In *Advances in neural information processing systems*, pages 368–374, 1997.

[4] J. P.-A. et al. Package 'mstknnclust'. `https://cran.r-project.org/web/packages/mstknnclust/mstknnclust.pdf`, 2019. [Online; accessed 8-October-2019].

[5] M. R. Garey and D. S. Johnson. A guide to the theory of np-completeness. *Computers and intractability*, pages 641–650, 1979.

[6] O. Grygorash, Y. Zhou, and Z. Jorgensen. Minimum spanning tree based clustering algorithms. In *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, pages 73–81. IEEE, 2006.

[7] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):881–892, 2002.

[8] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28(2-3):89–112, 2004.

[9] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[10] C. H. Papadimitriou. Worst-case and probabilistic analysis of a geometric location problem. *SIAM Journal on Computing*, 10(3):542–557, 1981.

[11] R. C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.