ESTIMATING THE NUMBER OF CLUSTERS FOR THE ALGORITHM OF K-MEANS IN A CONTEXT OF BIG DATA

Bruno Moura Valle Costa (Universidade de São Paulo)
brunomvcosta@usp.br

Renato de Oliveira de Moraes (Universidade de São Paulo)
remo@usp.br

Hugo Martinelli Watanuki (Universidade de São Paulo)
hwatanuki@usp.br

Cluster analysis is a technique of data science with applications in multiple fields of knowledge. This article presents the development and its application in the Big Data sphere. In this research, a platform from HPCC Systems named ECL was used for clustering in the database of crimes registered in the city of Chicago between 2001 and 2022. It was chosen as subject of analysis due to its massive volume of data that well exemplifies the context of Big Data and has free access. The HPCC Systems platform was chosen because is an open platform with parallel processing. However, the clustering algorithm available in the platform is K-means, that works only with numeric variables. As the chosen database has multiple qualitative fields relevant to analysis, like the type of crime that was committed, an alternative to the formation of clusters in a database with both numeric and categorical fields was developed. This article describes how the method k-prototypes, already available in Python, can be incorporated to ECL to parallel processing and distributed in order to form clusters from a very large database.

Keywords: Big Data, Cluster analysis, k-means, categorical data, non-hierarchical methods

1. Introduction

Cluster analysis has applications in multiple fields of knowledge. It consists of the subdivision of a database whose variables are classified neither as dependents nor independents and are referenced, in the context of Machine learning, as an unsupervised method. This method can be applied in diversified context, such as market segmentation and behavioral studies, in which it is possible to separate the data and create clusters inside of which there is high homogeneity and outside a high heterogeneity. This statistic tool has high impact and can generate high value in multiple scenarios, like public policies to specific groups, forecast of who is in which group, specific marketing for specific types of users, among other applications.

Cluster analysis can be divided in two subgroups of clustering methods: hierarchical and non-hierarchical. In simple terms, the main difference between them is that the second type requires as an input the number of *k* groups to be formed. Besides that, the non-hierarchical method requires less iterations and less computational power. Given that, when Big Data is the context of the study, the non-hierarchical method becomes preferred over the other.

Nowadays, multiple organizations have being collecting, storing and analyzing a massive and growing volume of data. They are usually referred to as "big data" due to their volume and speed at which they are collected, creating a new generation of tools to support decision-making (WATSON, 2014). Therefore, the study of these large databases progressively gains more attention at the same time that new technologies are collecting more data in real time, challenging traditional statistical approaches.

Given this context, cluster analysis appears to be a viable solution, specially the non-hierarchical methods, due to their lower requirements in terms of computational power and increased efficiency. The main focus of this study is the *K-means* algorithm, a non-hierarchical clustering method that fulfills the requirements described so far. This algorithm receives as an input a certain *k* number of centroids, which are then repositioned iteratively considering the Euclidian distance from the data point to the closest centroid, searching for the minimal distance.

This project utilized the big data processing platform HPCC Systems (*High Performance Computing Cluster*) created by *LexisNexis Risk Solutions and* focused on analysis of big data with high performance. Although the platform supports the k-means algorithm as a native library, it is only possible to work with categorical data if they are converted into dummy variables first.

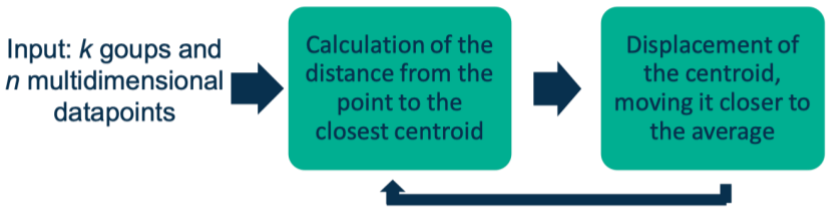
As a starting point to address this constrain, the *k-prototypes* library available in python was leveraged. Although this algorithm allows the processing of categorical data, it is not efficient for larger databases in a conventional computational platform. Hence, the need for importing this library into the *HPCC Systems* environment since it is better and more efficient when working with massive volumes of data.

2. Literature review

Separating a group of data objects from a database in homogenic groups, or clusters, is a fundamental operation of data analysis (Klosgen, Zytkow, 1996). Clustering represents an extremely common method in this type of data partition activity (Kaufman e Rousseeuw, 1990). Many algorithms already support a clustering solution for categorical variables, such as the hierarchical clustering methods (Anderberg, 1973), however, the computing costs of these algorithms is quadratic and for larger databases it becomes too costly and not efficient (Huang, 1998).

Kodinariya et al (2013) suggest that the *k-means* algorithm is the most common clustering method. For this reason, the study of its properties is relevant not only to *data mining* and *machine learning* areas, but also to a growing number of professionals in marketing, bioinformatics, customer management, engineering and other fields. Also, it is extremely more efficient for larger databases (Anderberg, 1973).According to the description of Arora et al (2015), the algorithm executes an iterative process that searches for convergence based on the Euclidian distance of a node (or data point) to the nearest cluster. The initial inputs are the group of *n* data points and *k* clusters. The average of the calculated Euclidian distances is taken and the cluster is repositioned to better fit that average and reduce the distance. An illustration of this process can be seen on the Figure 1.

Figure 1: Schematic of the *K-means* algorithm



Source: Author

However, as highlighted by Jian and Dubes (1988), this algorithm is capable of working only with numerical values as it reduces the cost of changing the value of the centroids. Therefore, as proposed by Huan (1998), a similar algorithm named *K-prototypes* can be applied, as it is capable of supporting categorical values.

The k-prototypes algorithm is already available in *python*, but it can be optimized to larger datasets through the *HPCC Systems* plataform, an opensource big data platform for intensive data processing, developed by the company *LexisNexis Risk Solutions* in the year 2000. The algorithm can be supported in the programming language utilized by the HPCC Systems, named ECL or Enterprise Control Language, via a programming language function named *Embedding*.

3. Methodology

This project has leveraged the *HPCC Systems* platform and its programming language *ECL* for coding an algorithm to support the clustering of *big data* containing numeric and categorical variables . In the simulation environment created, the data remains stored in nodes and is processed in parallel by the platform itself. Next, the data treatment and statistical methods adopted in the project are presented.

3.1. *K-Means*

As previously described, the *k-means* algorithm is an iterative non-hierarchical clustering method, which means that it requires as an *input* the number *k* of clusters to be generated throughout the iterations. This algorithm is already available in *HPCC Systems* exclusively for numerical data and as a complementary plug-in from the *Machine learning* library and, in that format, it was used as a starting point on the project.

3.2. *Elbow Method* and R^2

The Elbow Method is a graphical method widely used to determine the optimal number of clusters for a given dataset being clustered. According to Yuan and Yang (2019), a routine is followed with the basic idea of using the square of the distance of each point to its clusters. The sum of this values is called *SSE* and it is used as a performance indicator. Lower values of *SSE* indicate convergence of the model. When the number of clusters approaches the “real number” there is a steep fall on *SSE*. When the number of clusters surpasses the real value, the decrease of this value will be much slower. The method searches for the point of inflexion on the intensity of the decreased of the *SSE*, and graphically it resembles an elbow.

Complementary to the Elbow method, the R2 calculates the deviation of the predicted method (test base) versus the one of the original database, being tested for each value of the *k* clusters used on the Elbow algorithm in order to assess the quality of the model.

3.5. *K-prototypes*

The k-prototypes algorithm is a modification of the *k-means,* algorithm with the added support of categorical and/or non-numeric variables.

According to Huang (1998), the k-prototypes is as efficient as the *k-means algortihm*, but its usage depends on a few changes in the original algorithm . For numerical variables, a dissimilarity binary is adopted. In case the categories are the same, the coefficient is 0, but if they are different, it is 1. This binary value represents the "distance" between data points just as calculated iby the *k-means* method and, therefore, is able to be applied in the same way as previously presented.

Mathematically, the equation can be represented as follows:

$$d_1(X,Y) = \sum_{i=1}^{n} \delta\,(x_i, y_i)$$

Where d_1 is the distance between two data points, *n* the number of data points and x_i, y_i the coordinates of each data. This algorithm is the same as the *k-means* algorithm, with the following change in the calculation operation:

$$\delta\,(x_i, y_i) = \begin{cases} 0, & x_i = y_i \\ 1, & x_i \neq y_i \end{cases}$$

Defined the distance calculation, the initial centroids are chosen and the calculation of the distance between each point and its closest centroid is initiated, changing and improving the position of each centroid in each iteration. In *Python* this algorithm is available in the *k-prototypes* library.

3.6.Dataset

The chosen dataset for this project was the database of crimes from the Chicago Police Department between the years of 2001 and 2021. This dataset was chosen due to its ease of access, large size, good discretion of variables and popular usage for testing of algorithms in massive data analysis. The dataset has 1.77GB in size and 7.515.455 records with 22 variables each. Due to the volume of both datapoints and variables, whose nature is not ficticious but associated with real crimes, qualitative conclusions can be taken from clustering.

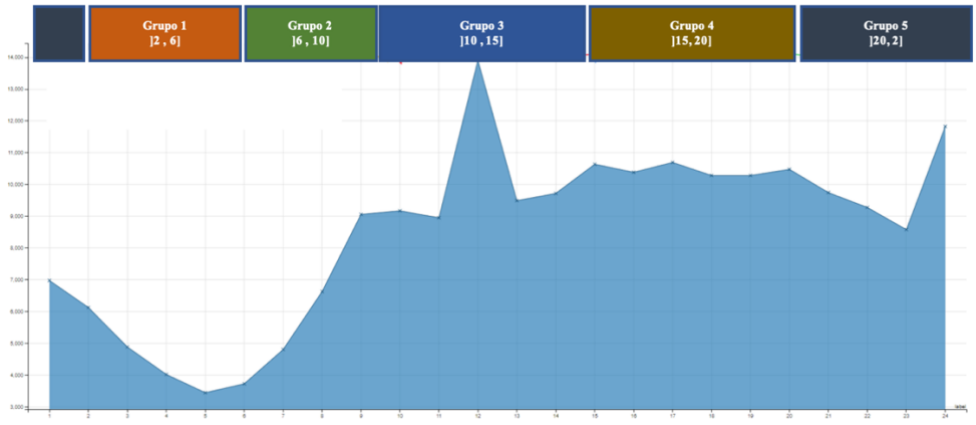
Many variables of the dataset can be associated t even though they are naturally different. An example consists of the 5 variables that represent micro divisions of the Chicago city: *block, Beat, District, Ward* and *Community Area*. Those variables present different internal subdivisions that result in different categories, relevant to different analysis. As the purpose of this study is the development of a statistical tool and not a profound analysis of the dataset itself, variables that can represent the same attribute as the location, for instance, can be can be removed from the analysis. Therefore, these 3 variables were chosen to analysis:

- Crime Type: 76 Categories listing the types of crimes registered in the city;
- Time: Transformed into a categorical variable with 5 groups, as presented in Figure 3.
- Location: Latitude and Longitude, transformed into kilometers trough a geometric conversion from an origin point – the *Willis Tower*. This origin point has the following coordinates:
    - Latitude: 41,8786367º N
    - Longitude: -87,6358583º O

  For each point, the following conversion to distance in km was adopted:
    - $Latitude\ (km) = (Latitude_{Point0} - Latitude) * Radius_{earth}$
    - $Longitutde\ (km) = \left(Longitude_{base} * \cos\left(Latitude_{point0}\right) - Longitude * \cos\left(Latitude\right)\right) * Radius_{earth}$

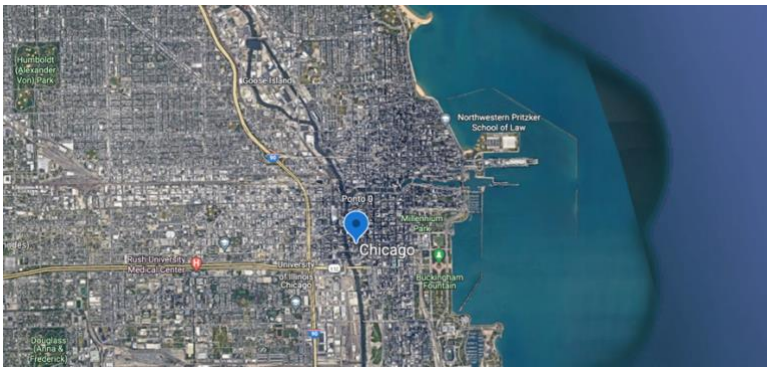
  Where $Radius_{earth}$ can be translated as the radius of the planet defined as $Radius_{earth} = 6371\ km$; $Latitude_{Point0}$ and $Latitude_{point0}$ as the previously presented coordinates of the *Willis Tower*.

Figure 2: Hourly distribution of crimes



Source: Author

Figure 3: Location of "Point 0" in a map



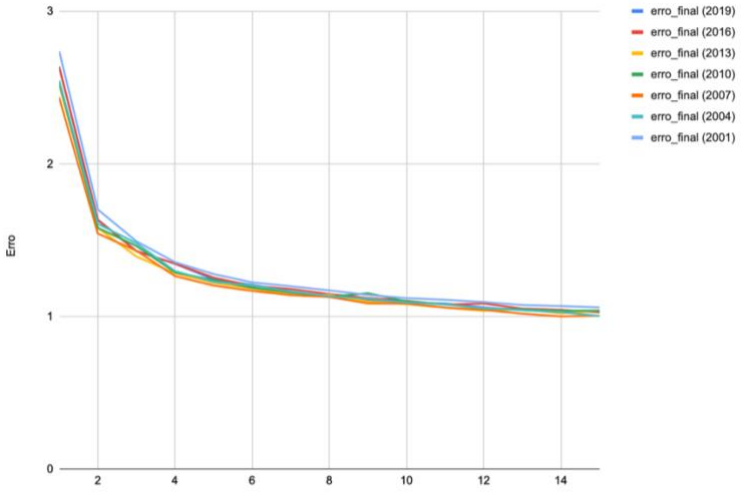
Source: Google Earth

4. Results

4.1. Algorithm for the calculation of the *Elbow Method*

An algorithm to calculate the error of each number of clusters *k was developed* in ECL. Thereafter, it was possible to plot these results and determine manually what would be the ideal number *k* of clusters of the chosen dataset. As a result, the chart presented in Figure 4 was plotted and a conclusion can be driven from the "elbow" formed at the n=4 number of clusters. Hence, it can be concluded that, for this specific dataset, the ideal number of clusters is 4.

Figure 4: Chart Number of Clusters X Error



Source: Authot

4.2.Adoption of the *k-prototypes* library in the *HPCC Systems*

In order to import the *k-prototypes* library to ECL, an available feature named *Embedding* was used, that consists in a function that supports the execution of a second programming language in ECL, in order to take advantage of the existing resources and benefits of that language. Thus, it works as a "function", accepting multiple initial values as *inputs*, processing them and

returning the desired output. The ECL code is available in the following github repository: https://github.com/Brunomvcosta/IC---Bruno.

During the utilization of this feature, special care was taken regarding the type of processing of the data. One of the main benefits brought by the *HPCC Systems* platform is the ability of processing the data in parallel from multiple computing nodes, allowing more efficiency. When the *embedding* function was ran, the researchers have confirmed that the parallel processing feature was still be working as intended.

4.3.Results in ECL with *embedding*

A code in ECL was developed, taking advantage of the *embedding* feature and the *k-prototypes* library available in *python*. As a result, for the number of n=4 clusters, previously determined as ideal according to the elbow method, the results presented in Table 1 were achieved.

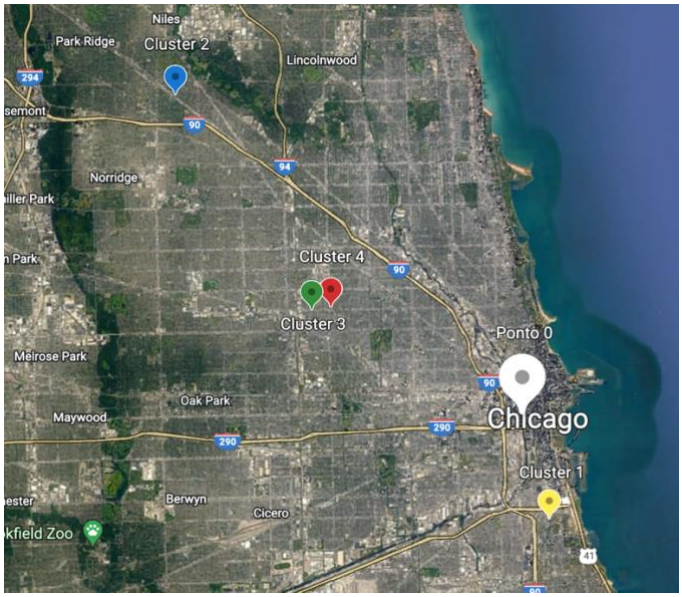
Tabele 1: Clusters formed in ECL

Cluster	*Hour*	*Crime*	*Distance Y (km)*	*Distance X (km)*
1	3.69	Theft	-0.42	2.64
2	4.53	Battery	4.20	7.14
3	1.56	Battery	4.41	7.30
4	3.00	Battery	4.33	7.23

Fonte: Autor

For easier visualization, these clusters were represented in a map with their respective distances in *Y* and *X* axis from Point 0 (Willis Tower) and plotted on *Google earth*. The map is presented in Figure 5.

Figure 5: Location of the centroids of each cluster on a map



Source: Google Earth

Thus, the following clusters were formed:

- Cluster 1 is next to the city center with the most common crime being theft happening more frequently in the timeframe from 10:00AM to 08:00PM. As it is a central commercial location, theft crimes are more common in Business hours, which is in line with the kind of economic activity of the location, based on *White Collar* jobs in larger office buildings, with usually highly-paid employees circulating in and out of the buildings and becoming attractive to criminals.
- Cluster 2 is next to the Chicago O'Hare Airport, the second largest airport in the country, and with predominance of battery crimes from 8PM to 2AM. This type of crime, with higher concentration after hours, can be associated with the lower airport passenger traffic and no natural light, thus favoring this kind of crime.
- Clusters 3 and 4 are concentrated in a lower income suburb, with the main crime being battery. As it is a residential suburb, the more common time of this cluster is supported by the fact that a larger number of people are either at home or returning home for rest. Both clusters differ in time. While the first has higher concentration from 8PM to 2AM, when people are still awake but going to bed, the second has a larger concentration from 4AM to 10AM, when people are awakening and getting ready for starting the day. Therefore, the formation of clusters in the same location regarding the same crime but in different times of the day can be observed.

These clusters represent the social interactions and economic activity of the city with multiple tendencies that directly relate to reality. Between 10AM and 8PM, there is a movement towards the commercial center of the city, where a considerably high number of people exercise their economic activity. At night and dawn, this concentration is transferred to the suburbs. The airport also works as a center of crimes due to the high volume of passengers, tourists and cargo traffic during a timeframe when the city is still "asleep".

The processing time of the algorithm was 10 minutes and 57 seconds when considered parallel processing.

4.4. Results in ECL with *embedding*

Complementary to the code in ECL, a code in *python* was also elaborated in order to compare the gains in efficiency and validation of the clusters formed in ECL. The results

are presented on Table 2.

Table 2: Clusters formed in Python

	Hour	*Crime*	*Latitude*	*Longitude*
1	3.69	THEFT	-0.42	2.64
2	4.53	BATTERY	4.20	7.14
3	1.56	BATTERY	4.41	7.30
4	3.00	BATTERY	4.33	7.23

Source: Author

The processing time of this algorithm was 44 minutes and 44 seconds when using python on Google Collab.. In Google's platform, the code can be processed in parallel trough blocks of code, but each block is processed sequentially, while in ECL the data is distributed and ran in parallel, allowing higher efficiency for larger datasets.

Comparatively, when the results from both platforms are presented side by side, the formation of identical clusters can be seen, with differences in value (when the attribute is numeric) only after the fourth decimal place. The categorical variables (Hour and Crime) formed identical clusters.

Tabele 3: Comparison of results

Hour		Crime		Latitude		Longitude	
Python	ECL	Python	ECL	Python	ECL	Python	ECL
3.69	3.69	Theft	Theft	-0.42	-0.42	2.64	2.64
4.53	4.53	Battery	Battery	4.20	4.20	7.14	7.14
3.00	3.00	Battery	Battery	4.33	4.33	7.23	7.23
1.56	1.56	Battery	Battery	4.41	4.41	7.30	7.30

Source: Author

Also, when considered the time of execution, a gain of 75.5% on time can be seen when comparing the code in ECL with parallel processing with the code in *python.*

Table 4: Comparison of Execution Time

Execution Time (*s*)	
ECL	*Python*
657	2684
Δ=	-75.5%

Source: Author

5. Conclusion

Considering all the procedures executed throughout the study, it can be verified the non-exact nature of *big data* analysis. In this sense, the application of the k-means algorithm for clustering massive volumes of data can be challenging due to the diversity of the data. In the specific case of the Chicago police crime database, the challenge was related to the fact that many of the fields were non-numerical and/or categorical. To deal with this challenge, it was decided to explore the application of the k-prototypes algorithm, for execution in parallel and distributed data processing platforms..

Throughout the project, complementary tools were used to give meaning to the clustering of the data and to support the analysis of the assertiveness of the code. In order to achieve that, the elbow method was applied to determine the ideal number of clusters to be formed.

Based on the optimal number of clusters, the barrier presented by the categorical nature of the data attributes was reinforced as each and any cluster generated by the unchanged *k-means* wouldn't have any real meaning, since the categorical variables treated as continuous integers would lose any information that they could possess. Thus, a study of the applicability of the *k-prototypes* algorithm in ECL took place, and HPCC Systems has allowed for an optimized big data analysis. Adapting this algorithm, it was possible to achieve the generation of 4 *clusters* with relevant information of the historic tendency of crimes in the city of Chicago. This study contributes with the big data processing area by providing tools for practitioners that are challenged to cluster large volumes of diversified data , which tends to be a common scenario in the *Big Data industry.*

References

Klosgen, W. and Zytkow, J.M. 1996. **Knowledge discovery in databases terminology**. Advances in Knowledge Discovery and Data Mining, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Eds.), AAAI Press/The MIT Press, pp. 573–592

Kaufman, L. and Rousseeuw, P.J. 1990. **Finding Groups in Data—An Introduction to Cluster Analysis**. Wiley.

Dubes, R. and Jian, A.K. 1979. Validity studies in clustering methodologies. Pattern Recognition, 11:235–254.

MIDDLETON, Anthony; CHALA, Arjuna. **Introduction to HPCC (High Performance Computing Cluster). 2011**. Disponivel em: http://cdn.hpccsystems.com/whitepapers/wp introduction HPCC.pdf. Acesso em: 28 maio 2021.

WATSON, H. J. **Tutorial: Big Data Analytics: Concepts, Technologies, and Applications**. Communications of the Association for Information Systems, Vol. 34, Article 65, 2014.

ARORA P., DEEPALI, D., VARSHNEYM S. Analysis of K-means and K-Medoids Algorithm for Big Data. Procedia Computer Science, Vol 78, Article 507-512, 2016.

Z. HUANG. **Extensions to the k-means Algorithm for Clustering Large Data Sets with Categorical Values**. Data Mining and Knowledge Discovery. Vol 2, Article 283-304, 1998.

YUANG, C., YANG, H. **Research on K-Value Selection Method of K-Means Clustering Algorithm**. Multidisciplinary Scientific Journal, Vol 16, Article 226-235, 2019.

ROUSSEEUW, P.J., KAUFMAN, I. **Finding Groups in Data: An Introduction to Cluster Analysis**. New York John Wiley&Sons, Hoboken, NY, USA, 1990.