Clustering Credit Card Data With K-Means++

João Brito, M9984
Department of Computer Science
University of Beira Interior
Covilhã, Portugal
joao.pedro.brito@ubi.pt

***Abstract*—In this project, the K-Means++ algorithm will be explored, by implementing it from scratch. Subsequently, normalisation and feature reduction techniques will be used interchangeably to determine their impact on the performance of K-means++.**

I. Introduction

A clustering algorithm attempts to group data instances, based on their attributes, so that instances in the same cluster are as similar as possible. On the other hand, if two instances belong to different clusters, it should be safe to assume that they are quite different (i.e. belong to different classes). With that being said, K-Means is an unsupervised algorithm by design, meaning that we don't possess any information regarding the cardinality or naming of those classes. In this document, two techniques will be applied to find the optimal number of clusters (referred to as ”K”).

II. Dataset Overview

The dataset chosen for the purposes of this project contains behavioural information about credit card usage, by a certain group of customers. There are close to 8600 instances with 17 independent variables [1]:

- **Balance**: current account balance at the time;
- **Balance Frequency**: how frequently the balance is updated (on a scale of 0 to 1);
- **Purchases**: quantitative amount of purchases made;
- **One Off Purchases**: maximum purchase amount done in a short period of time;
- **Installment Purchases**: amount of purchases done in installments;
- **Cash Advance**: cash in advance given;
- **Purchase Frequency**: how frequently purchases were made (on a scale of 0 to 1);
- **One Off Purchases Frequency**: how frequently purchases were made in a short period of time (on a scale of 0 to 1) ;
- **Installment Purchases Frequency**: how frequently installmen purchases were made (on a scale of 0 to 1);
- **Cash Advance Frequency**: how frequently cash was given in advance (on a scale of 0 to 1);
- **Cash Advance Trx.**: number of transactions made with cash in advance;
- **Purchase Trx.**: number of purchase transactions made;
- **Credit Limit**: limit of the credit card;
- **Payments**: total amount of payments made;
- **Minimum Payments**: smallest payment made;
- **Prc. Full Payment**: percentage of payments made in full;
- **Tenure**: tenure of the credit card;

The file containing the dataset went through a simple pre-processing stage where lines with missing values were removed, followed by a shuffling operation.

III. Overview Of K-Means++

Being one of the most popular clustering algorithms, K-Means relies on simple and intuitive concepts. To that end, it requires one really important parameter: ”K”. This integer can be derived from intuition (i.e. what is expected *a priori*) or through generic heuristics (section IV). Based on such value, the algorithm will try to create ”K” clusters centred on ”K” centroids. Given its iterative nature, this procedure involves moving the centroids closer and closer to the data points associated with them, until no meaningful changes occur.

K-Means++ uses much the same formula, with some improvements regarding centroid initialisation: they are chosen from the data points and, typically, start off further apart from each other. This approach ensures that the centroids can connect with more points and, ultimately, minimises the risk of centroids falling in the same region [2]:

Algorithm 1: K-Means++

```
input : $K$ (number of clusters), $X$ (data instances)
output: $K$ centroids
// centroid initialisation
$C_1 \leftarrow$ sample a point uniformly at random from $X$
for $i \leftarrow 1$ to $K$ do
    $C_i \leftarrow$ sample $x \in X \sim \frac{D(x)^2}{\sum_{x \in X} D(x)^2}$
end
// regular K-Means implementation
while !stopping_criterium do
    // cluster assignment
    for $i \leftarrow 0$ to $len(X) - 1$ do
        connect $x_i$ to the closest $c \in C$
    end
    // centroid update
    for $i \leftarrow 0$ to $K - 1$ do
        $C_i \leftarrow$ mean of every $x$ connected to $C_i$
    end
end
```

IV. Validation Metrics

There are two types of evaluation metrics for a clustering algorithm: internal and external. On one hand, an internal metric only takes in consideration the data points that have been clustered. Therefore, there is no ground truth to refer to and the measurable aspects rely on the uniformity and distance between clusters.

On the other hand, external metrics possess the actual class labels. While most of the available instances lack their labels (after all, we are talking about an unsupervised algorithm), during test time it may be possible to annotate a small quantity of new, never before seen, instances. This information would be really helpful in determining the success (or lack thereof) of K-Means. Obviously, it isn't always possible to get new instances and annotate them, so these techniques are more laborious and sometimes even impractical.

For the purposes of this work, two internal metrics were used, given the lack of labels in the original dataset and the impossibility of getting new data.

A. Elbow Method

The elbow method is one of the most widely used heuristics for determining the optimal number of clusters. It requires a loss function, of which we can highlight two [3]:

- **Distortion**: the average of the squared distances between every data point and its centroid;
- **Inertia**: the sum of the squared distances between every data point and its centroid;

For every value of "K" tested, the corresponding value of the chosen loss function is computed and plotted on a graph. Then, we can, visually, determine where the values of the loss function starts to decrease in smaller decrements (the point of diminishing returns). The point described is the elbow of the curve.

Despite being effective as is, this method relies on visual inspection and the interpretation of the observer. To solve this situation, the elbow can be analytically determined by calculating the distance between each *(K, loss)* point and an imaginary line connecting the first of those points and the last (figure 3) [4].

B. Silhouette Score

The following method can be described as being more informed and reliable [5]. The main characteristics of the silhouette score lie on the fact that it takes in consideration the intra-cluster ($(a(i))$) and inter-cluster ($(b(i))$) distances:

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} D(i, j),$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} D(i, j).$$

The $a(i)$ term computes the average distance between data point i and every data point j belonging to the same cluster (the division by $|C_i| - 1$ ignores the distance between i and i). The other term ($(b(i))$) returns the minimum distance between i and every other j belonging to the other clusters (i.e. the closest neighbour to i).

Finally, the $s(i)$ term joins the two terms mentioned before:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\})}, if|C_i| > 1.$$

This $s(i)$ term (not defined for "K"=1, due to the inter-cluster distances) gets closer to 1 when $b(i)$ is bigger than $a(i)$ (the clusters are far apart and the intra-cluster distances are small). That would be the ideal situation, whereas a value close the -1 means that the instances are likely in the wrong cluster. If, by chance, $s(i)$ evaluates to 0, the interpretation would be that the instance i is close to the boundary between clusters.

V. Experiments

The experimental stage of this project involved some tests with normalisation and PCA. These two processing steps are commonly mentioned within the clustering domain, so the following tables will try to establish some key notions about such techniques. As such, there were 4 configuration in play:

- **Baseline**: raw and normalised data (table I);
- **PCA 95%**: PCA with 95% variability applied on both the raw and normalised data (table II);
- **PCA 90%**: PCA with 90% variability applied on both the raw and normalised data (table III);
- **PCA 75%**: PCA with 75% variability applied on both the raw and normalised data (table IV);

A note worthy detail is that, in the following tables, the elbow score that is highlighted is not the lowest value, but rather the one that is furthest from the imaginary line between the first and last point (i.e. the elbow, as seen in section IV-A).

	Raw data		Normalised data	
K/Metric	**Distortion**	**Silhouette**	**Distortion**	**Silhouette**
1	4,47*e-7	-	0.65	-
2	3,18*e-7	**0.5**	0.38	**0.4**
3	2,88*e-7	**0.5**	**0.32**	0.38
4	**2,41*e-7**	0.47	0.28	0.35
5	2,06*e-7	0.35	0.25	0.34
6	1,85*e-7	0.35	0.24	0.33
7	1,75*e-7	0.35	0.21	0.29
8	1,55*e-7	0.33	0.20	0.3

TABLE I: Baseline performance (raw and normalised data)

The above table shows the conservative nature of the silhouette score (it hinted at a value of 2-3 for K with the raw data and 2 with the normalised data) and the more forgiving results of the elbow method (K should be 4 with the raw data and 3 with the normalised version). These results clearly show that the normalised configuration puts the data points in a tighter, less expansive, space and, therefore, the clusters are closer to each other (a more cluttered space).

	PCA 95% raw data		PCA 95% normalised data	
K/Metric	**Distortion**	**Silhouette**	**Distortion**	**Silhouette**
1	4,27*e-7	-	0.62	-
2	2,98*e-7	**0.51**	0.35	**0.37**
3	2,68*e-7	0.5	**0.28**	0.31
4	**2,21*e-7**	0.48	0.27	0.35
5	1,87*e-7	0.39	0.22	0.35
6	1,77*e-7	0.4	0.19	0.33
7	1,56*e-7	0.4	0.17	0.33
8	1,36*e-7	0.4	0.16	0.34

TABLE II: PCA at 95% variability

The results in table II do not differ much from the ones observed in table I, which is to be expected given that we only decreased the overall variability by 5%:

	PCA 90% raw data		PCA 90% normalised data	
K/Metric	**Distortion**	**Silhouette**	**Distortion**	**Silhouette**
1	4,06*e-7	-	0.59	-
2	2,77*e-7	**0.52**	**0.32**	0.41
3	2,49*e-7	**0.52**	0.30	**0.42**
4	**1,96*e-7**	0.48	0.22	0.41
5	1,68*e-7	0.42	0.20	0.39
6	1,59*e-7	0.43	0.18	0.38
7	1,38*e-7	0.43	0.16	0.39
8	1,19*e-7	0.41	0.14	0.39

TABLE III: PCA at 90% variability

Table III shows some more interesting results, where there is visible confusion between the appropriate number of clusters. The elbow score either points out a value of 2 or 4 and the silhouette score is torn between 2, 3 and even 4.

	PCA 75% raw data		PCA 75% normalised data	
K/Metric	**Distortion**	**Silhouette**	**Distortion**	**Silhouette**
1	3,55*e-7	-	0.51	-
2	2,26*e-7	**0.54**	0.24	0.49
3	**1,77*e-7**	**0.52**	**0.17**	**0.49**
4	1,44*e-7	0.45	0.14	0.49
5	1,28*e-7	0.46	0.12	0.47
6	1,1*e-7	0.45	0.11	0.48
7	1,05*e-7	0.44	0.08	0.45
8	0,87*e-7	0.44	0.07	0.5

TABLE IV: PCA at 75% variability

This last table (IV) and the application of PCA at 75% variability show and affinity towards 3 clusters and also serve the purpose of allowing us to plot a 3D point cloud of our data instances (with 75% variability we are left with 3 features, the maximum number of dimensions we can see):

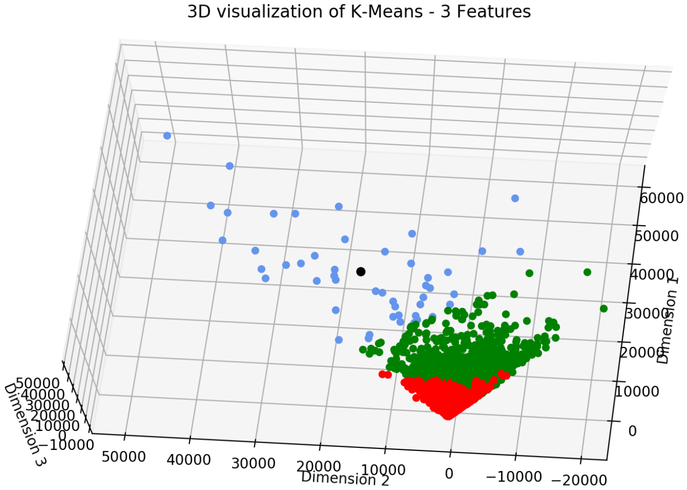

Fig. 1: 3D plot of the raw data with PCA at 75% and 3 clusters

In the above figure we can see that the 3 clusters are very different in shape: one cluster is really sparse, another is dense and the remaining one sits in between the other two. The following figure shows the normalised version of the data:

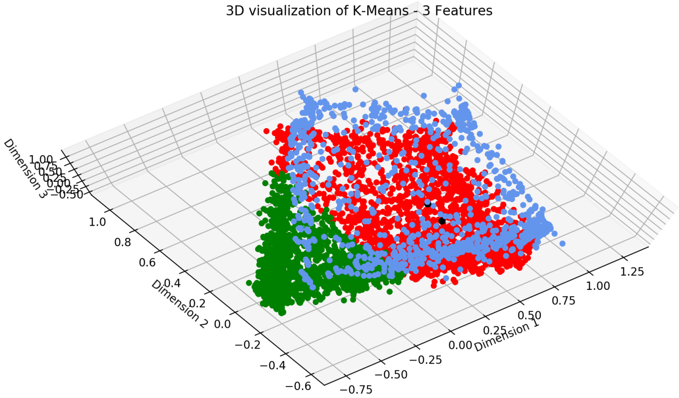

Fig. 2: 3D plot of the normalised data with PCA at 75% and 3 clusters

Figure 2 shows more or less a similar story (the green and red colours are swapped from figure 1): the top cluster is sparse, the one in green (formerly in red) is dense and the last one is average. It is really interesting to see that, for the blue cluster, the centroid is not the geometric centre but is rather close the region of higher density.

Finally, as for the aforementioned plot of the elbow method, the following figure shows the imaginary line and the distances calculated between it and the *(K,loss)* points. The auxiliary lines in blue are actually orthogonal to the imaginary line, despite not looking like it (it has to do with the proportions of the axes). In this case, the elbow is not exactly noticeable, and this a very good reason for the addition of an analytical method of determining it (the values next to the dots are the distance between the corresponding dot and its projection on the imaginary line):

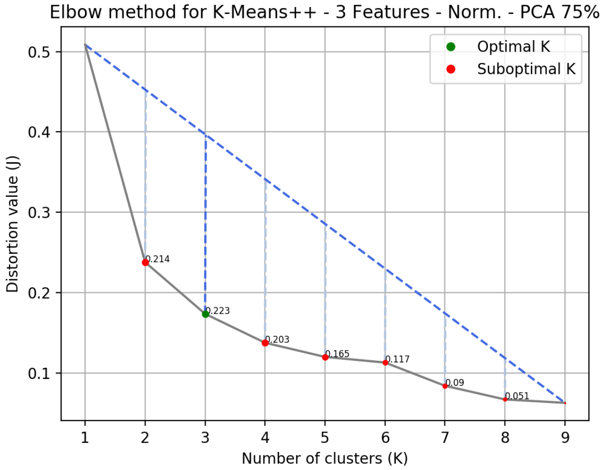

Fig. 3: Elbow method plot for PCA at 75%

VI. Conclusion

This project involved the clustering of data points in an unsupervised manner with K-Means++. Several experiments were conducted with the goal of determining the impact of normalisation and PCA for the aforementioned purposes. Those techniques help to reduce the effect of outliers and balance the features, which leads to better cluster formation and centroid positioning.

References

[1] Credit Card Dataset For Clustering. [Online] https://www.kaggle.com/arjunbhasin2013/ccdata

[2] K-Means++: The Advantages of Careful Seeding. [Online] http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf

[3] Elbow Method for optimal value of k in KMeans. [Online] https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/

[4] Finding the optimal number of clusters for K-Means. [Online] https://www.linkedin.com/pulse/finding-optimal-number-clusters-k-means-through-elbow-asanka-perera/

[5] Silhouette (clustering). [Online] https://en.wikipedia.org/wiki/Silhouette_(clustering)