VarKMeans

Giuseppe Di Fatta and Romain Choulet

July 2019

I. Introduction

—KMeans is an algorithm used in Data Mining to find k clusters (group of data) for the data set used for the algorithm. It use the number of cluster that the user want, k, to initialize k centroids (actually the means of the clusters), and move them to divide the data points into k groups and partition the data set. Our idea was to improve this algorithm and allow the user to enter a range $[k_{min}, k_{max}]$ for the number of clusters (k) of the KMeans algorithm. Our method will then apply the algorithm of KMeans for each $k \in [k_{min}, k_{max}]$ and finds which one is the more likely to be optimal. But how can we determine this "more likely to be optimal" k ?

The idea is to use the Elbow Method, which is a method used to find the more likely optimal number of clusters k for a data set. We determine the percentage of variance explained (which correspond to the variance between the clusters against the variance of each data point between the Grand Mean of the data set) for each application of KMeans in $[k_{min}, k_{max}]$, and we try to find the k value for which the percentage of variance explained is good enough, and for which the increment of number of cluster doesn't really have a better percentage of variance. Basically, we try to find the k where the next gain of percentage value explained will drop and form an "elbow" in a graphic representation.

To calculate this percentage of variance explained, we use the F-Test ratio :

$$F(x) = \frac{BSS(x)}{TSS}$$

The BSS is the Beetween-Group Variability, defined by :

$$BSS(x) = \sum_{i}^{x} |C_i|(m - m_i)^2$$

with x the number of clusters, $|C_i|$ the number of elements in the cluster C_i, m_i the centroid of the cluster C_i and m the Grand Mean (the mean of all the data points, defined by :

$$m = \sum_{y \in \cup C_i} \frac{y}{N}$$

with N the number of data points in the data set, and y a data points). The TSS, the Total Variation, is defined by :

$$TSS = \sum_{y \in \cup C_i} (m - y)^2$$

with y a data point, C_i a cluster and m the Grand Mean. For a same data set, it's a constant, regardless of the number of clusters. Actually, the TSS actually correspond to the Beetween-Group Variability plus the Within-Group Variability (WSS, defined by

$$WSS(x) = \sum_{i}^{x} \sum_{y \cup C_i} (m_i - y)$$

with x the number of clusters, C_i a cluster, y a data point and m_i the centroid of the cluster C_i). When we increase the number x of clusters, the WSS tends to 0 and the BSS tends to the TSS. This is absolutly logical, because when $x = N$ (with x the number of clusters and N the number of data points), the percentage of variance explained is obviously 100% because

$$\lim_{x \to N} F(x) = \frac{BSS}{BSS} = 1$$

Conversely, when we decrease the number x of clusters, the BSS tends to 0 and the WSS tends to the TSS. It's also logical, because if there is only one cluster, there cannot be a Between-Group Variability.

To use the Elbow method and determine the "elbow", two equivalent techniques have been implemented: the first one compare the second order derivative of the F-Test for each $k \in [k_{min}, k_{max}]$, and the second one compare the measurements of the angles formed by the vectors generated by two points $(x, F(x))$ in a graphic representation of the percentage of variance explained, with x the number of cluster and $F(x)$ the F-Test of the KMeans algorithm for x number of clusters.

II. Second Order Derivative

—The idea here is to use the second order derivative to compare the percentage of variance explained for each KMeans algorithm applied to a number x of clusters. We want to find were the difference between two consecutive second order derivative is maximum, because that's where the x number of clusters could be the more likely to be optimal k.

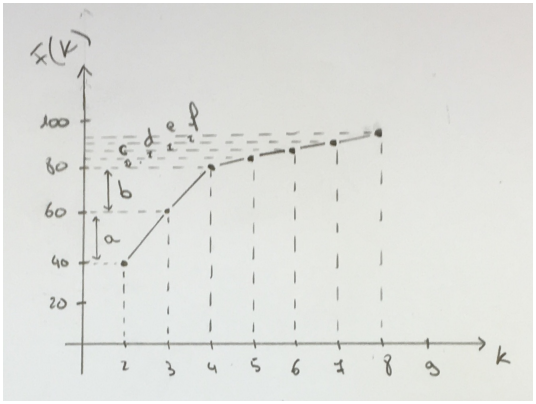

FIG.1 : Representation of Elbow Method with distances

Here, we have to compare the variances of the different k.

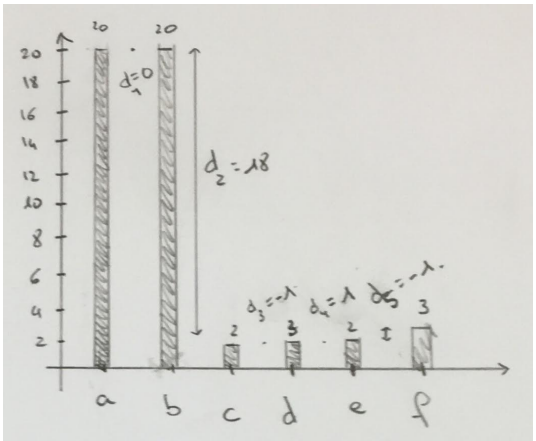

FIG.2 : Comparison of the distances

Let's use the function F for the F-Test defined by $F(x) = \frac{BSS(x)}{TSS}$, with x the number of clusters for the KMeans algorithm. We can see in the $fig.2$ that there is an important drop for $k = 4$, which correspond to the "elbow" in the $fig.1$. Actually, this is where the difference between $F(x)-F(x-1)$ and $F(x+1)-F(x)$ is maximum. Let's name this function ψ.

$$\psi(x) = (F(x) - F(x-1)) - (F(x+1) - F(x))$$
$$\psi(x) = F(x) - F(x-1) - F(x+1) + F(x)$$
$$\psi(x) = 2F(x) - F(x-1) - F(x+1)$$

We have defined a function named ψ.

$$\boxed{\psi(x) = 2F(x) - F(x-1) - F(x+1)}$$

with x the number of cluster, and $F(x)$ the F-Test applied to the KMeans algorithm for x clusters. We apply this ψ function to all the $x \in [k_{min}+1, k_{max}-1]$ (the bounds are excluded because the elbow cannot be one of the bounds, but we can apply this algorithm for $[k_{min}-1, k_{max}+1]$ to include the bounds that the user chose). The $y \in [k_{min}+1, k_{max}-1]$ for which $\psi(y)$ is the maximum value of ψ will be the more likely to be optimal number of clusters k for the data set.

We can test this algorithm in our example of the FIG.1, with the points $k_2(2, 40)$, $k_3(3, 60)$, $k_4(4, 80)$, $k_5(5, 82)$, $k_6(6, 85)$, $k_7(7, 87)$, $k_8(8, 90)$.

We have :

$$F(2) = 40;$$
$$\mathrm{F}(3) = 60;$$
$$\mathrm{F}(4) = 80;$$
$$\mathrm{F}(5) = 82;$$
$$\mathrm{F}(6) = 85;$$
$$\mathrm{F}(7) = 87;$$
$$\mathrm{F}(7) = 90$$

So :

$$\psi(3) = 2 * 60 - 40 - 80 = 0;$$
$$\psi(4) = 2 * 80 - 60 - 82 = 18;$$
$$\psi(5) = 2 * 82 - 80 - 85 = -1;$$
$$\psi(6) = 2 * 85 - 82 - 87 = 1;$$
$$\psi(7) = 2 * 87 - 85 - 90 = -1;$$

We find that $\psi(4) = 18$ is the maximum value of ψ for $[2, 8]$. So $k = 4$ is the optimal number of clusters.

III. Angle Comparison

—The idea here is to use a theoretical comparison between the angles formed by the vectors generated by two consecutive points $(x; F(x))$ and $(x+1; F(x+1))$ in the graphic representation of the Elbow method.

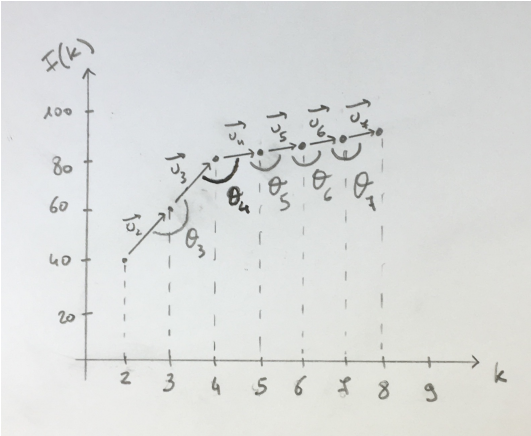

FIG.3 : Representation of Elbow Method with vectors

We can see that the "elbow" is in fact the smaller angle formed by the vectors of the graph (here θ_4). We will name $\overrightarrow{u_x}$ the vector generated by the points of abscissa x and $x+1$. To find the measuring of an angle with x at its top, we have to use the vectors $\overrightarrow{u_x}$ and $-\overrightarrow{u_{x-1}}$. The $\overrightarrow{u_i}$ vectors have for first coordinate 1 (they are vectors generated by the point $(x; F(x))$ and the point $(x+1; F(x+1))$). Their second coordinate is the difference between $F(x+1)$ and $F(x)$. Let's name this function Δ. So the coordinates of $\overrightarrow{u_x}$ are $(1;\Delta(x))$.

The mathematical formula to calculate the measuring of an angle between two vectors $\overrightarrow{v}(v_1; v_2)$ and $\overrightarrow{w}(w_1; w_2)$ is :

$$\arccos(\frac{< v; w >}{||u|| * ||w||})$$

which is equal here to :

$$\arccos(\frac{v_1w_1 + v_2w_2}{\sqrt{(v_1)^2 + (v_2)^2} * \sqrt{(w_1)^2 + (w_2)^2}})$$

We just have to apply this formula to our vectors $\overrightarrow{u_x}(1; \Delta(x))$ and $-\overrightarrow{u_{x-1}}(-1; -\Delta(x-1))$. That give us :

$$\arccos(\frac{-1 - \Delta(x-1) * \Delta(x)}{\sqrt{1 + (\Delta(x-1))^2} * \sqrt{1 + (\Delta(x))^2}})$$

Let's name this function ϕ.

$$\boxed{\phi(x) = \arccos(\frac{-1-\Delta(x-1)*\Delta(x)}{\sqrt{1+(\Delta(x-1))^2}*\sqrt{1+(\Delta(x))^2}})}$$

with x the number of cluster. We apply this ϕ function to all the $x \in [k_{min} + 1, k_{max} - 1]$ (the bounds are excluded because the elbow cannot be one of the bounds, but we can apply this algorithm for $[k_{min} - 1, k_{max} + 1]$ to include the bounds that the user chose). The $y \in [k_{min} + 1, k_{max} - 1]$ for which $\phi(y)$ is the minimal value of ϕ will be the more likely to be optimal number of clusters k for the data set.

We can test this algorithm in our example of the FIG.3, with the points $k_2(2, 40)$, $k_3(3, 60)$, $k_4(4, 80)$, $k_5(5, 82)$, $k_6(6, 85)$, $k_7(7, 87)$, $k_8(8, 90)$.

We have :

$$\begin{gathered}\Delta(2) = 20;\\ \Delta(3) = 20;\\ \Delta(4) = 2;\\ \Delta(5) = 3;\\ \Delta(6) = 2;\\ \Delta(7) = 3\end{gathered}$$

So :

$$\begin{gathered}\phi(3) = 180;\\ \phi(4) = 156.6;\\ \phi(5) = 171.9;\\ \phi(6) = 171.9;\\ \phi(7) = 171.9;\end{gathered}$$

We find that $\phi(4) = 156.6$ is the minimal value of ϕ for $[2; 8]$. So $k = 4$ is the optimal number of clusters. We can see that visually in the $fig.3$, the "elbow" is for $k = 4$ too.

VI. Conclusion

—We found those two method to determine the "elbow" in the gain of percentage of variance explained. But the F function have to be monotonic, and more specifically a non-decreasing function. Actually, we need to find if $F(x + 1) \geq F(x)$, $\forall x \in [k_{min}, k_{max}]$. We can reduce this inequality to $BSS(x + 1) \geq BSS(x)$, because $F(x) = \frac{BSS(x)}{TSS}$, and TSS is constant $\forall x \in [k_{min}, k_{max}]$.

Like we said in the introduction, when we decrease the number of clusters, the BSS tends to 0, and when we increase the number of clusters, the BSS grows and tends to the TSS. So we can say that BSS is a non-decreasing monotonic function. So we have $BSS(x+1) \geq BSS(x)$, which implies $F(x+1) \geq F(x)$, $\forall x \in [k_{min}, k_{max}]$. So we can say that F is a non-decreasing monotonic function.

In theory, our methods find the "elbow" in the gain of the percentage of variance explained, finding the maximum of second order derivative or the minimal angle. But what if there is no maximum of second order derivative or minimal angle ? Imagine that the gain between to consecutive percentage of variance explained is strictly equal; how can our methods be effective ?

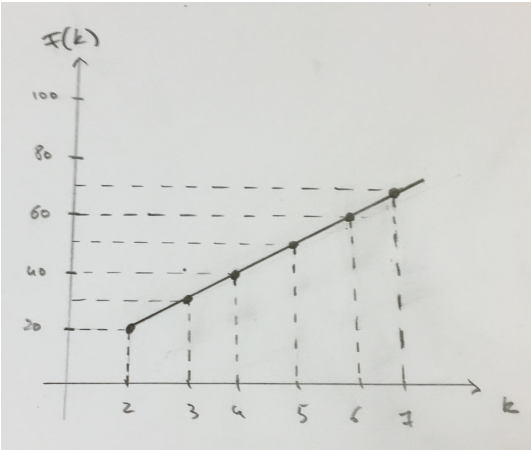

FIG.4 : Representation of Elbow Method , case where the gain is always equal

We can deal with this situation by adding this case: $\forall x \in [k_{min}+1, k_{max}-1]$, if all the $\psi(x)$ (or $\phi(x)$) are equal, we take the first iteration (in fact $k_{min}+1$) as the more likely to be optimal k. Finding k clusters for a data set may require a lot of calculus; so less we have clusters, more efficient and shorter could be our algorithm. That's why we use the first iteration ($k_{min}+1$).

We can also ask ourselves if it is useful to use the "elbow" that we found if the drop of gain is really negligible, i.e we can't really detect it visually ? In other terms, is it worth it to use the more likely to be optimal number of clusters y if there actually is an other number of clusters $y' < y$ whose percentage of variance explained is not so different ? As previously said, y' would be more efficient because it requests less calculation, for a result quite similar. It can really help us to make our algorithm faster.

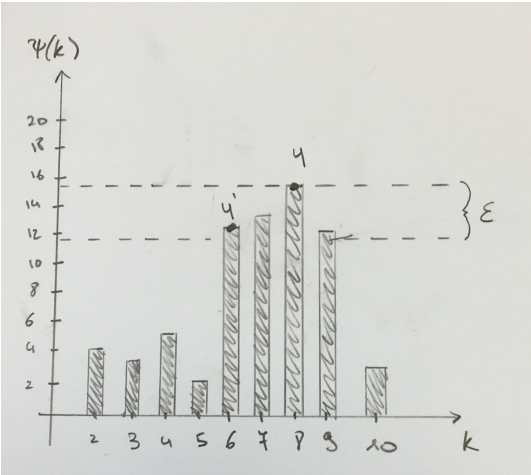

FIG.5 : Representation of $\psi(x)$

We could introduce an ε criteria, like we could see in $fig.5$. When we find a $y \in [k_{min}+1, k_{max}-1]$ number of clusters for which $\psi(y)$ is the maximum value of ψ for $[k_{min}+1, k_{max}-1]$ (so y is the elbow), we could try to find if there is any $y' \in [k_{min}+1, y]$ such that $\psi(y) \geq \psi(y') \geq \psi(y) - \varepsilon$. In other terms, it could be more efficient if we use the y' for which $\psi(y') \in [\psi(y) - \varepsilon, \psi(y)]$ is minimal, always in order to maker our algorithm shorter, faster and more efficient. But what value should take ε ?

A first idea might be to use the standard deviation of ψ. But in fact, if we use this standard deviation σ, there can be cases where the elbow is really important and where the difference between $\psi(y)$ and $\psi(y')$ is really too important to say that y' is good enough ($fig.6$).

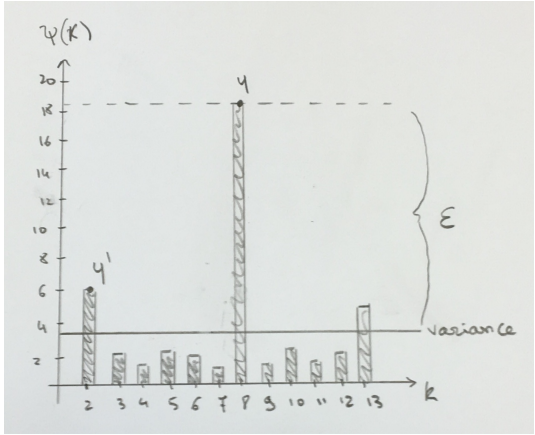

FIG.6 : Representation of $\psi(x)$

The solution that we found is to use the minimum between a percentage of $\psi(y)$ and the variance of ψ; we used to set it at $\varepsilon = min(10\% * \psi(y), \sigma(\psi))$. It could be sufficient to approximate the more likely to be optimal number of clusters k.

With the Angle Comparison method, we did not find yet any way to determine if there is an other number of clusters $y' < y$ whose percentage of variance explained is not so different and could be good enough. So, although these two methods are equivalent, it could be more interesting to use the Second Order Derivative method, because we can adapt the "elbow" that we found to the situation and use another one which is good enough.