The Kernel Trick and clustering: the kernel k-means method

Mateus Maia[1,*] **and Anderson Ara**[2]

[1]Student of Master in Statistics, IME-UFBA, Salvador, Brazil
[2]Assistant Professor, IME-UFBA, Salvador, Brazil

ABSTRACT

The main objective of clustering methods is to segment the observations in subgroups or groupings, from measures of similarity between them. One of the classic examples is K-means, that Euclidean distance is used between similarity measures. However, there are some minor limitations, especially with regard to data that are not linearly separable. Kernel K-Means are as alternative to solve this problem. The methods were implemented in the R, and, as a result, it was noticed that in addition to solving well the issue of lack of linearity Kernel K-Means, is also a result of clustering when compared to classic K-Means.

Introduction

The goal of clustering is to segment and divide a dataset in sub-groups, using a from similarity measure, in order to group similar observations to try to recognize patterns, or reveals insights from data. One of most classical clustering techniques is the K-Means, a lot of results, research, and extensions involves this method along the lasts decades (1). This technique operationalize this classification minimizing the within-groups distances (2). Despite the efficiency of this method, there are some limitations to identify non-linear separable clusters. Analyzing this problem, some authors proposed some ways to deal with the non-linearity problem (3); (4). One approach to deal with that situation it's the Kernel K-Means, as an extension from K-Means where the observations are taken to a higher dimensional space where they can be linearly separable. Although the Kernel trick was used initially in the SVM context (5), he could be extended to others distance-based algorithms (6), which includes the K-Means clustering technique. Ir order to evaluate the efficiency from the Kernel Trick to solve the non-linearity problem in clustering, several types of kernels were evaluated, as well differents hyperparameters for each one.

K-Means

Considering a dataset $\mathbf{X} = (x_1, ..., x_n)$, where $\mathbf{x}$ are the observations from data, the main objective from k-means it's to minimize

$$L = \sum_{j=1}^{K} \sum_{x_i \in C_j} ||x_i - m_j||^2, \tag{1}$$

where K is the numbers of clusters and

$$m_j = \sum_{xi \in C_j} \frac{x_i}{n_j} \tag{2}$$

is the the centroid that belongs j cluster from n_j observations. Initially the centroids are randomly defined. Then, the euclidean distance between each observation from each centroid is calculated, where the shortest distance between x_i and m_j associates the observation to the j cluster. This process is repeated iteratively, updating the centroids form Equation 2., until the moment where the convergence it secured.

Kernel K-Means

The kernel k-means can be defined as an extension from the k-means technique where, using the kernel trick, it's possible to take the observations and the centroids to higher dimensions where the distances can be calculated. Then , defining $\mathbf{X} = (x_1, ..., x_n)$ as the observations, exists the clusters C_k where $k = 1, ..., n$, and centroids $\mathbf{m}_k$ to each cluster C_k in dimension $\mathbb{R}^n$, where $\Phi : \mathbb{R} \to \mathbb{R}^n$. Therefore, we can define, assuming that each cluster have at least one observation.

$$\mathbf{m}_k = \sum_{xi \in C_k} \frac{\Phi(x_i)}{|C_k|} \tag{3}$$

where $|C_k|$ refers to the numbers of observations that belongs to C_k.

Algorithm 1: K-Means

Input: The matrix of observations **X** and the numbers of clusters **K** are parameters of kernel function, the convergence criterion **p**

Output: The data matrix **X** and the class that refers to the clusters.

while $\varepsilon \geq p$ **do**
  Calculate randomly **K** centroids m_j;
  Calculate the distances D_{ij} between x_i and each centroid;
  Associate the shortest distance D_{ij} to the cluster j ;
  Recalculate the centroids using the Equation 2;
  Calculate ε, as the distance between the new cluster the previous one.;
end
Return the matrix **X** with the observations and their respective clusters.

Then determining the squared distance as $D(\mathbf{x}_i, \mathbf{m}_k) = ||\phi(\mathbf{x}_i) - \mathbf{m}_k||^2$ we can rewrite it as

$$D(\mathbf{x}_i, \mathbf{m}_k) = \phi(\mathbf{x}_i)^\intercal \phi(\mathbf{x}_i) - 2\phi(\mathbf{x}_i)^\intercal \mathbf{m}_k + \mathbf{m}_k^\intercal \mathbf{m}_k \tag{4}$$

substituting $\mathbf{m}_k$ from equation (3) in (4)

$$D(\mathbf{x}_i, \mathbf{m}_k) = \phi(\mathbf{x}_i)^\intercal \phi(\mathbf{x}_i) - 2\frac{\sum_{x_j \in C_k} \phi(\mathbf{x}_i)^\intercal \phi(\mathbf{x}_j)}{|C_k|} + \frac{\sum_{x_l \in C_k} \sum_{x_j \in C_k} \phi(\mathbf{x}_j)^\intercal \phi(\mathbf{x}_l)}{|C_k|^2} \tag{5}$$

to calculate the inner products we can use the kernel trick and obtain

$$D(\mathbf{x}_i, \mathbf{m}_k) = \kappa(x_i, \mathbf{x}_i) - 2\frac{\sum_{x_j \in C_k} \kappa(x_i, \mathbf{x}_j)}{|C_k|} + \frac{\sum_{x_l \in C_k} \sum_{x_j \in C_k} \kappa(x_j, \mathbf{x}_l)}{|C_k|^2} \tag{6}$$

The Gram-Matrix (6), also called kernel matrix, is the matrix of dimension $n_x n$ that computes all the inner products between $\phi(x)$ observations. The variable $\kappa(x_i, xj)$ corresponds to each element from this matrix. Then, isn't necessary calculate every $\kappa(i, j)$ during the convergence process, and the distance is given by:

$$D(\mathbf{x}_i, \mathbf{m}_k) = K(x_i, \mathbf{x}_i) - 2\frac{\sum_{x_j \in C_k} K(x_i, \mathbf{x}_j)}{|C_k|} + \frac{\sum_{x_l \in C_k} \sum_{x_j \in C_k} K(x_j, \mathbf{x}_l)}{|C_k|^2} \tag{7}$$

Therefore, after associate each observation from each one of k clusters, the information is updated in the dataset, in order to repeat this process until it reaches the convergence, that is, there are no more changes on observation classification. The process can be finished too if reach the maximum number of iterations. To secure the initialization of this algorithm it's necessary to randomly assign a group to each observation in the beginning of the process.

There are a lot of types of kernels that were developed along the decades, the most popular are the Gaussian Kernel, Linear, and Sigmoidal that were implemented in the article, besides them, was also used the *Exponential Kernel* and *Cauchy Kernel*, although they are not so common.

- ***Gaussian Kernel*:** the kernel gaussian is a type of radial basis kernel and it's given by

  $$\kappa(x, y) = e^{-\frac{||x-y||^2}{2\sigma^2}} \tag{8}$$

  also can be rewrite as

  $$\kappa(x, y) = e^{-\gamma||x-y||^2} \tag{9}$$

  is interesting evaluet the choice of the hyperparameterσ that's imporant to the clustering process, once that high values can result in *overfitting*, whie lower values can reduce the kernel's capacity to deal with non-linear situations.

- ***Polynomial Kernel:*** The polynomial kernel is given by

  $$\kappa(x, y) = (x^\intercal y + c)^d \tag{10}$$

  where d is the degree of the polynomial. Higher degrees can deal better with the non-linearity but they can cause *overfitting* and increase exponentially the computational cost.

- ***Linear Kernel:*** Can be treat as a particular case of *polynomial kernel* where $d = 1$.

$$\kappa(x,y) = x^{\intercal}y + c \tag{11}$$

- ***Exponential Kernel:*** similar to *Gauss Kernel*, but isn't used the square distance

$$\kappa(x,y) = e^{-\frac{||x-y||}{2\sigma^2}} \tag{12}$$

- ***Cauchy Kernel:*** This kernel cam from Cauchy Distribution (7). It is a long tail kernel and can be used to provide long-range influence and sensitivity over high-dimensional space.

$$\kappa(x,y) = \frac{1}{1+\frac{||x-y||^2}{\sigma}} \tag{13}$$

Algorithm 2: Kernel K-Means

Input: The observations $\mathbf{X}$, numbers of clusters, and $\mathbf{K}$, $\mathbf{w}$ maximum number of iterations.
Output: The data matrix $\mathbf{X}$ and their cluster classifications

Give random class to each observation $\mathbf{x}_i$;
Calculate the kernel matrix $\mathbf{K}$ for the determined kernel;
while *The convergence isn't obtained or isn't reached the maximum number of iterations.* **do**
    Calculate the distances $D(\mathbf{x}_i, \mathbf{m}_k)$ between x_i and each centroid;
    Associate the shortest distance $D(\mathbf{x}_i, \mathbf{m}_k)$ to cluster k ;
    Recalculate the distances $D(\mathbf{x}_i, \mathbf{m}_k)$;
end
Return the matrix $\mathbf{X}$ and their respective clusters.

Methodology

The *K-Means* and *Kernel K-Means* clustering methods were applied to both synthetic (Figure 1) and real data sets. The artificial are described in Table 1, while the real data provided by *UCI Repository of Machine Learning* (Murphy 1994) is described in Table 2. The aforementioned methods were used and each was evaluated for accuracy. Mainly synthetic data covering the issue of nonlinearity was used.

Table 1. Synthetic Data

Dataset	Nº of observations	Nº Clusters	Dimension
Jain	373	2	2
Flame	240	2	2
PathBased	300	3	2
Circles	399	2	2

Table 2. Real Datasets

Dataset	Nº of observations	Nº Clusters	Dimension
Iris	150	3	4
Seeds	199	3	7
Glass	300	6	9

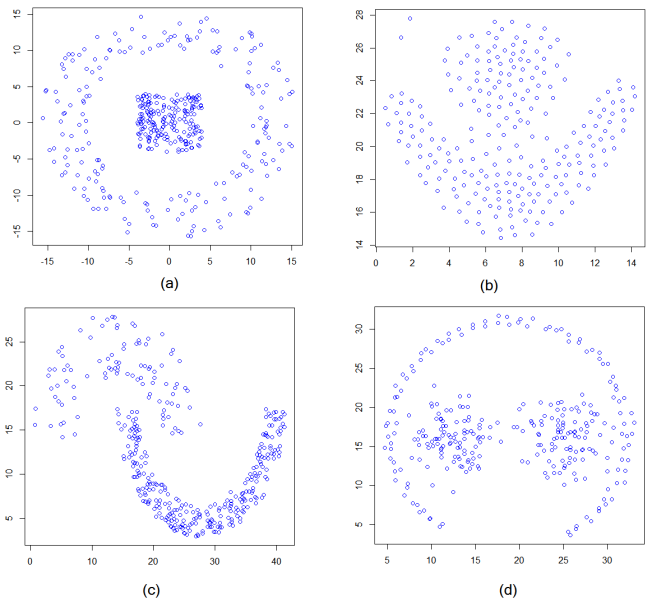

Figure 1. Synthetic Datasets. Respectively: (a) Circles, (b) Flame, (c) Jain, (d) PathBased

Results

All results were summarized in Table 3 which puts the accuracy measure for each of the methods for all databases used. It can be seen that for some datasets, especially the artificial ones, the kernels showed some improvement. The *Polynomial Kernel* used was degree 2, and the sigma parameter of the other methods was defined as $sigma = 1$. It can be seen that some Kernels showed very low accuracy, and this may be caused by a wrong choice of sigma parameter for the case, so another analysis regarding the choice of this value may be necessary. However, when the parameter fits well, as was the case with the Exponential Kernel for the given circles, it is possible to clearly see the difference of the clustering result for a data that is not linearly separable (Figure 2).

Table 3. Accuracy for each method for all databases

Banco de Dados	K-Means	RBF	Poly.	Lin.	Exp.	Cauchy
Jain	**0,767**	0,528	0,662	0,743	0,759	0,649
Flame	0,841	0,942	0,821	0,829	**0,975**	**0,975**
PathBased	0,743	0.853	0,797	0,773	**0.970**	0.953
Circles	0,531	0,917	**1,000**	0,579	**1,000**	0,912
Smiley	0,551	**1,000**	**1,000**	0,560	**1,000**	**1,000**
Cassini	0,854	**1,000**	0,928	0,928	**1,000**	**1,000**
Iris	0,920	0,893	0,853	0,887	**0,940**	0,900
Seeds	0,889	**0,909**	0,864	0,899	0,884	0,899
Glass	0,887	0,912	0,853	0,798	**0,921**	0,833

The others example over the synthetic datasets are shown in Figure 2

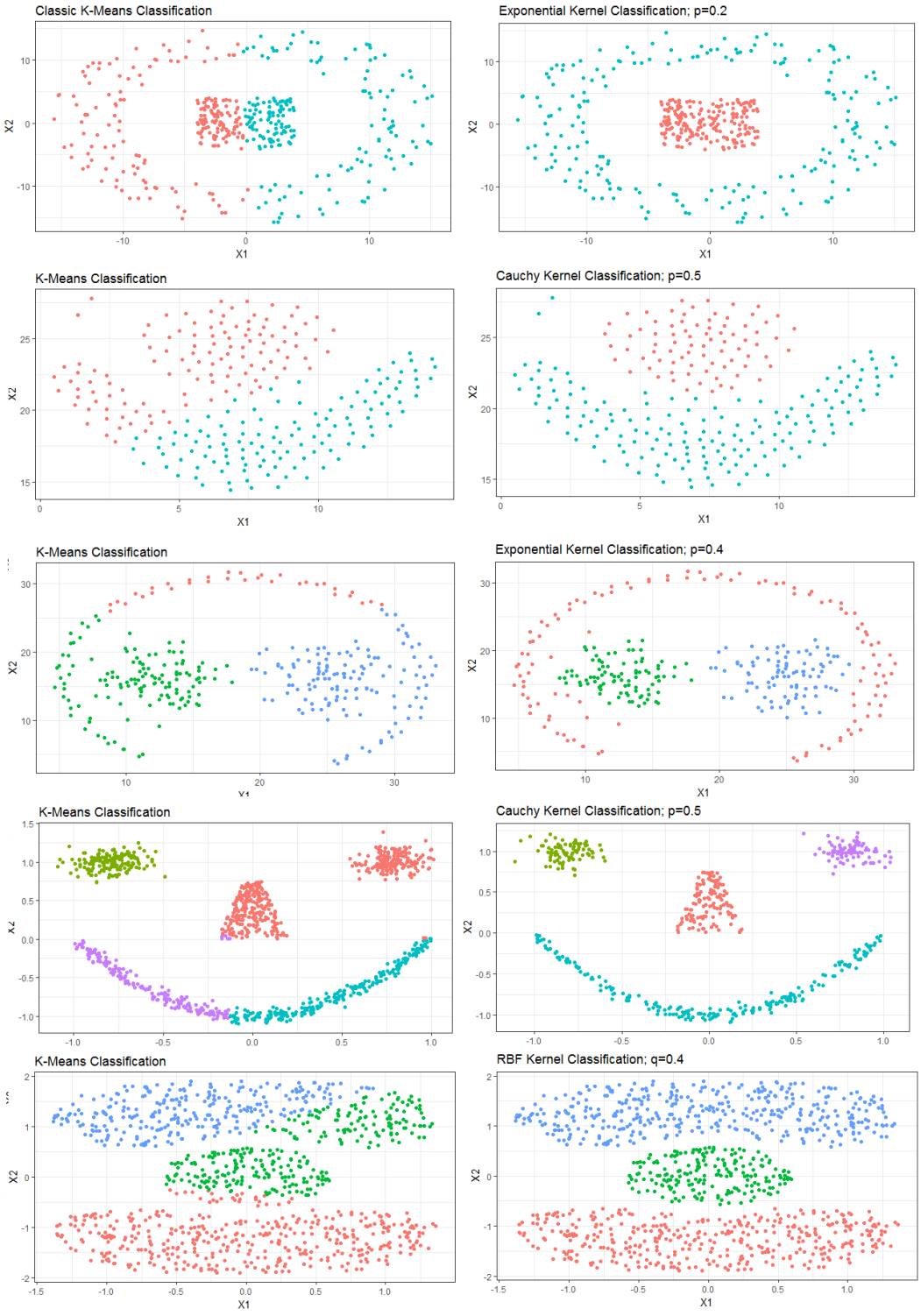

Figure 2. Comparison of the classic K-Means and the Kernel K-Means that presented the highest accuracy for each database.

Conclusion

When compared with the classic K-Means, the Kernel K-Means technique presented bests results, evidenced by greater values in accuracy, and a better methodology to deal with non-linear classification problems. The study from parameter selection was efficient and give a good framework to select the hyperparameters for each method. To future works it's interesting the possibility to ensemble the kernels to figure out if betters results can be achieved.

References

1. Steinley, D. K-means clustering: a half-century synthesis. *Br. J. Math. Stat. Psychol.* **59**, 1–34 (2006).

2. Hartigan, J. A. Clustering algorithms, new york: John willey and sons. *Inc. Pages113129* (1975).

3. Dhillon, I. S., Guan, Y. & Kulis, B. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 551–556 (ACM, 2004).

4. Ramesh, C. R., DVManjula, D. G. J. & Sastry, C. Dimensionality reduction for optimal clustering in data mining. *Int. J. on Comput. Sci. Eng.* **3**.

5. Vapnik, V. *The nature of statistical learning theory* (Springer science & business media, 1995).

6. Schölkopf, B. The kernel trick for distances. In *Advances in neural information processing systems*, 301–307 (2001).

7. Basak, J. A least square kernel machine with box constraints. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 1–4 (IEEE, 2008).