# K-Means Clustering para base de dados Íris

**Gustavo Silva Costa**

[1]Ciência da Computação – Pós-Graduação em Ciência da Computação (UFU)

{gustavosc91@gmail.com

## 1. Introdução

Algoritmos de aprendizado de máquinas visam melhorar com experiências alguma tarefa, comumente são representados por duas classes, os supervisionados e os não supervisionados. Algoritmos supervisionados dizem respeito a classificação de novas instancias de hipóteses com base em instancias previamente classificadas, sendo mais adequados para problemas de regressão e classificação, já os não supervisionados apresenta uma abordagem para problemas onde não existe hipóteses previamente classificadas, estes algoritmos se ajustam melhor para problemas de agrupamentos.

O problema apresentado neste trabalho, visa agrupar a base de dados Iris Data Set de acordo com a características de cada flor, tais características são referente a largura e comprimento de suas sépalas e pétalas. A base de dados em questão contém 3 classes de 50 instâncias cada.

O algoritmo não supervisionado K-means, será apresentado neste trabalho como solução otimizada para agrupar instâncias semelhantes, tal agrupamento utiliza-se do conceito de clusterização, onde um conjunto de observações é particionado em agrupamentos naturais ou agrupamento de padrões, de modo que a medida de similaridade entre qualquer par de observações atribuído a cada agrupamento minimize uma função de custo especificada.

## 2. Algoritmo K-Means

Apresentado por em [MacQueen et al. 1967], o algoritmo K-means está entre os mais conhecidos algoritmos de clusterização, por sua fácil implementação e eficiência em seu desempenho. Neste tópico daremos uma breve explicação para o algoritmo, em [Haykin 2009] é possível encontrar uma visão mais detalhada do algoritmo.

O algoritmo em questão aborda da seguinte maneira o problema de agrupamento; Dado um conjunto de dados multidimensional, de quantidade N, que deve ser particionado em um conjunto de K clusters propostos, em que K é menor que o número de dados, temos a seguinte relação:

$$j = C(i), i = 1, 2, 3, ..., N \tag{1}$$

Onde j é um cluster onde a i-ésima instância do conjunto de dados convergiu para a mais adequada classificação.

Em seguida, deve-se relacionar a i-ésima observação $x_i$ ao j-ésimo cluster, em uma releção de muitos-para-um, cada unidade do conjunto de clusters deve ser inicializada

com valores aleatórios, para classificar a observação $x_i$ em relação a um cluster, devemos utilizar a seguinte equação:

$$\pi_{ml} = argmin \sum_{i=1}^{m} (x_i - \pi)^2 \tag{2}$$

A equação 1 pode ser lida da seguinte maneira: Dado um conjunto de N observações, encontre C que atribui essas observações aos clusters K de forma que, dentro de cada cluster, a medida média de dissimilaridade das observações atribuídas a partir da média do cluster seja minimizada.

O ultimo passo realizado pelo algoritmo, é de ajustar cada instancia do cluster de acordo com o agrupamento de seus elementos, para isso faz-se a média dos elementos pertencente ao cluster, representado na equação 3:

$$\pi_{ml} = \frac{1}{m} \sum_{i=1}^{m} x_i \tag{3}$$

Note que a implementação do algoritmo [1] representa o agrupamento dos dados e ajustes dos clusters pelas respectivas funções: agruparDados e ajusteCluster.

## 3. Utilizando K-means para Base de dados Iris

O algoritmo K-means foi escolhido para execução deste problema pela sua fácil implementação e um desempenho satisfatório, nesta sessão irei apresentar a utilização do K-means para a base de dados Iris presente no diretório UCI.

A base de dados Íris conta com a representação de 150 flores, no qual estão divididas em 3 classes, a figura 1 expõem a distribuição das instâncias de acordo com sua classe, cruzando as medidas das sépalas com as das pétalas,

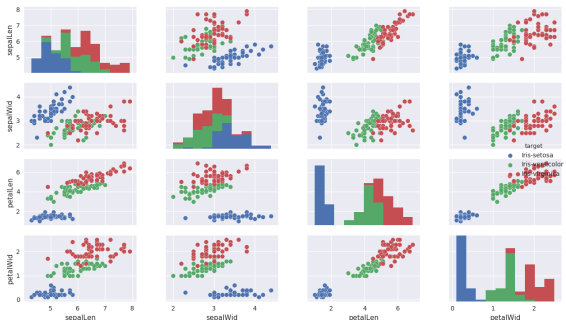


**Figura 1. Representadas em azul estão as Iris-setosa, em verde as Iris-versicolor e em vermlho Iris-virginica.**

O valor de K para o algoritmo K-means, foi inicializado com 3 para esse problema, porém existem técnincas para encontrar o valor ótimo de k, quando este é desconhecido [Mitchell 1997]. Assumiu os seguintes valores iniciais para cada instancia de K:

[1] Link para o algoritmo implementado https://github.com/gustavodev91/kmeansML

$$k = [[2, 2, 2, 2], [3, 3, 3, 3], [4, 4, 4, 4]] \tag{4}$$

O critério de parada do algoritmo é exposto pela equação 5, onde leva em consideração se o número de itarações é maior que o número de épocas pré estabelecido, ou se valor do cluster antigo é igual novo cluster.

$$(iteracao > epocas)||(klusterAntigo == klusterNovo) \tag{5}$$

Para execução do algoritmo foi definido o número de épocas máximo sendo 50 e para o valor de k inicial definido em eq. 4, foram necessário 15 iterações para que os cluster começassem a se repetir (figura 2).

```
[[4.7, 3.1, 1.39, 0.19], [5.33, 3.17, 2.51, 0.66], [6.47, 2.96, 5.18, 1.8]]
[[4.96, 3.36, 1.44, 0.23], [5.35, 2.89, 3.02, 0.89], [6.36, 2.91, 5.05, 1.74]]
[[5.0, 3.41, 1.46, 0.24], [5.48, 2.52, 3.76, 1.14], [6.43, 2.94, 5.15, 1.79]]
[[5.0, 3.41, 1.46, 0.24], [5.57, 2.63, 3.98, 1.23], [6.55, 2.97, 5.3, 1.86]]
[[5.0, 3.41, 1.46, 0.24], [5.67, 2.67, 4.07, 1.26], [6.59, 2.98, 5.37, 1.9]]
[[5.0, 3.41, 1.46, 0.24], [5.71, 2.67, 4.12, 1.27], [6.61, 2.99, 5.4, 1.93]]
[[5.0, 3.41, 1.46, 0.24], [5.75, 2.7, 4.15, 1.3], [6.63, 2.99, 5.44, 1.94]]
[[5.0, 3.41, 1.46, 0.24], [5.79, 2.71, 4.22, 1.34], [6.67, 3.0, 5.51, 1.97]]
[[5.0, 3.41, 1.46, 0.24], [5.82, 2.72, 4.25, 1.36], [6.7, 3.01, 5.55, 1.99]]
[[5.0, 3.41, 1.46, 0.24], [5.82, 2.73, 4.31, 1.39], [6.76, 3.03, 5.59, 2.0]]
[[5.0, 3.41, 1.46, 0.24], [5.83, 2.73, 4.33, 1.4], [6.79, 3.04, 5.62, 2.01]]
[[5.0, 3.41, 1.46, 0.24], [5.85, 2.74, 4.34, 1.4], [6.8, 3.04, 5.64, 2.03]]
[[5.0, 3.41, 1.46, 0.24], [5.88, 2.74, 4.37, 1.41], [6.82, 3.06, 5.69, 2.06]]
[[5.0, 3.41, 1.46, 0.24], [5.88, 2.74, 4.38, 1.43], [6.85, 3.07, 5.71, 2.05]]
[[5.0, 3.41, 1.46, 0.24], [5.88, 2.74, 4.38, 1.43], [6.85, 3.07, 5.71, 2.05]]
```

**Figura 2. Valores de clusters encontrados pelo algoritmo**

Ao final das 15 iterações podemos comparar a figura 3 com a figura 1, no caso a figura 3 representa o agrupamento final do algoritmo, após o ponto de parada ser atingido.

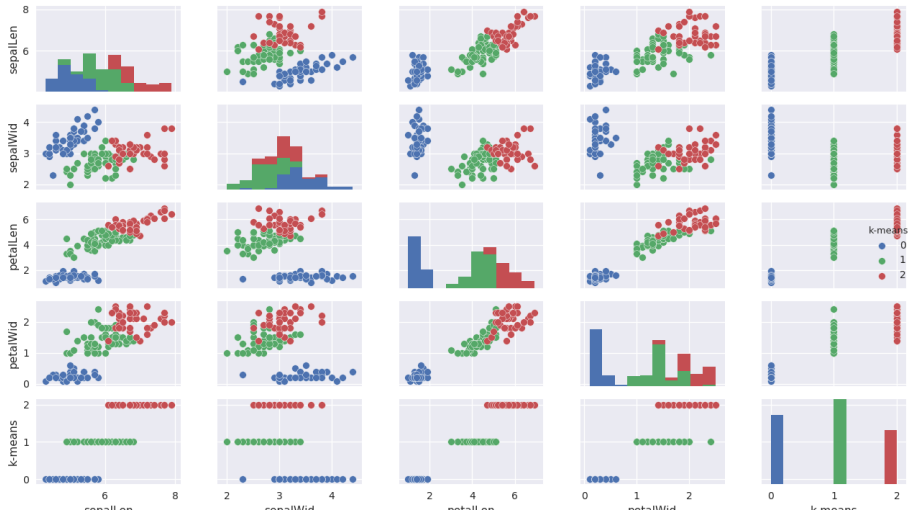


**Figura 3. Agrupamento final**

Podemos ver que o agrupamento final apresenta um valor semelhante à função alvo de cada instância, a diferença de valores se dá pela diversidade de caracteristicas das flores.

O algoritmo K-means implementado, ao final retorna os clusters que melhor classificam as instâncias, para isso a figura 4 demonsta o gráfico da instancia final dos clusters, comparando com a figura 1, podemos ver uma relação bem próxima com os dados originais.

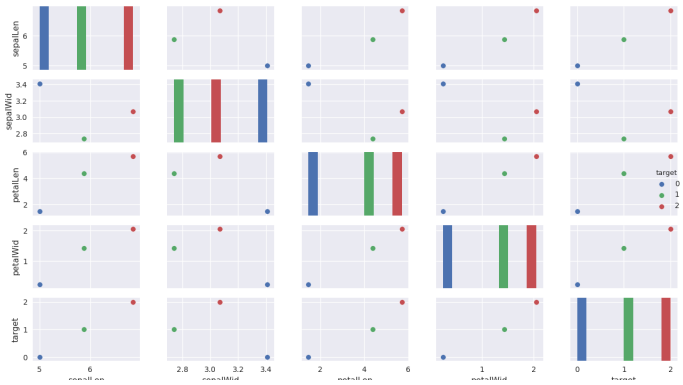


**Figura 4. Valores finais dos clusters**

## 4. Considerações Finais

O algoritmo mostrado na Seção 2 ilustra o agrupamento de dados realizado por um algoritmo não supervisionado, é um algoritmo de fácil implementação e que retorna resultados satisfatórios, podendo ser empregado a diversas bases de dados.

Para o caso do agrupamento da base de dados Íris, ele trouxe resultados relativamente satisfatório, podendo ser comparado a classificação original dos dados.

A implementação do algoritmo realizada em linguagem de programação Python pode ser melhorada em níveis assintóticos, assim como a representação inicial de suas k instâncias de cluster pode ser realizado de forma dinâmica seguindo o algoritmo EM [Mitchell 1997].

## Referências

Haykin, S. S. (2009). *Neural networks and learning machines*. Pearson Education, Upper Saddle River, NJ, third edition.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.