Identificação de Clusters Utilizando o K-Means

ENG121 - Reconhecimento de Padrões

Autor: Matheus Castro Silva

RESUMO

O trabalho a seguir apresenta uma discussão do algoritmo o K-Means que utiliza a teoria de Conjuntos Ordinários, nas abordagens de agrupamento e classificação clássicas.

Palavras Chaves **- algoritmos, K-Means, classificação, reconhecimento de padrões.**

I. INTRODUÇÃO

O presente trabalho busca identificar a eficiência do algoritmo K-means, para encontrar os clusters presente em um conjunto de dados no plano xy de 2 dimensões.

II. DEFINIÇÃO DO PROBLEMA

Dado um conjunto de pontos no plano 2D, com as coordenadas x e y, identifique os clusters presente neste conjunto por meio do algoritmo K-Means.

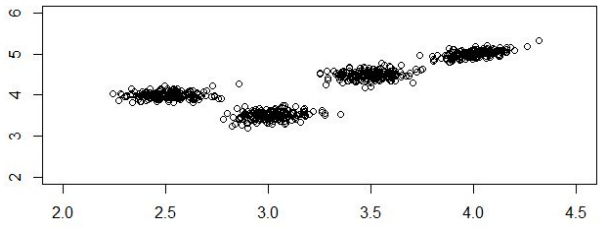

Imagem 1: Ilustração do conjunto de pontos no plano xy a serem utilizados.

III. IMPLEMENTAÇÃO K-MEANS

O K-Means é um dos algoritmos não-supervisionados mais simples que resolve o problema de clustering.

O procedimento segue um jeito fácil e intuitivo de classificar um conjunto de dados em um certo número de grupos (assuma k grupos) fixados previamente.

A ideia principal é definir k centróides, um para cada grupo. Esses centróides devem ser colocados em lugares bem escolhidos pois locais diferentes produzem resultados diferentes. Na implementação realizada, os centróides foram inicializados de forma aleatória.

A seguir, é preciso associar cada dado do espaço de entrada ao centróide mais próximo. Quando não há mais dados para serem associados, o primeiro passo acaba e um agrupamento inicial está feito.

Neste passo, é necessário recalcular k novos centróides como sendo os baricentros dos grupos resultantes do passo anterior. Depois de feitos os cálculos, uma nova associação dos dados ao centróide mais próximo é realizada.

Um laço de repetição é gerado. Como resultado desse laço, pode-se notar que os k centróides mudam sua localização passo a passo até que nenhuma mudança seja feita, ou seja, até que os centróides parem de se mover.

Finalmente, a meta dessa algoritmo é minimizar uma função objetivo, nesse caso uma função de erro quadrática. A função objetivo é:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left|\left| x_i^{(j)} - c_j \right|\right|^2$$

onde $\left|\left| x_i^{(j)} - c_j \right|\right|$ é uma medida de distância escolhida entre o ponto *xi* e o centro do grupo *cj*, é um indicador da distância entre os n dados do espaço de entrada e os seus respectivos centros de grupos.

Resumindo, o algoritmo segue os seguintes passos:

1. Coloque k pontos no espaço representado pelos dados que estão sendo agrupados. Esses pontos representam os centróides iniciais de cada grupo.
2. Associe cada dado ao grupo que tenha o centróide mais próximo;
3. Quando todos os dados tiverem sido associados, recalcule as posições dos k centróides;
4. Repita os passos 2 e 3 até que os centróides parem de se mover;
5. Isso produz uma separação dos dados em grupos dos quais a métrica a ser minimizada pode ser calculada.

O critério de parada é feito da seguinte forma: a posição dos centróides em cada iteração é armazenada em um vetor de centróides. A cada nova iteração, o cálculo dos novos centróides é armazenado em um vetor auxiliar. No final da iteração, uma comparação é feita entre os dois vetores: se todos os valores forem iguais, o programa para; se algum valor diferir, o vetor

auxiliar é copiado para o vetor de centróides e uma nova iteração começa.

IV. RESULTADOS

Utilizou-se um conjunto de dados apresentando a priori 4 clusters separados visualmente, então o parâmetro K ajustado para o algoritmo foi de K = 4.

Os resultados estão presentes nas imagens de 2 à 3.

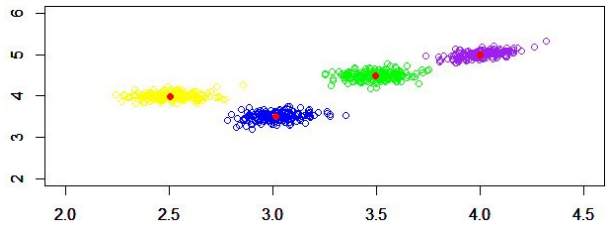

Imagem 2: Ilustração do resultado de um ótimo global.

A imagem 2 reflete muito bem o resultado esperado, o algoritmo foi capaz de encontrar os 4 clusters de maneira eficiente. Portanto sabemos que nem sempre acontece o ótimo global utilizando este algoritmo.

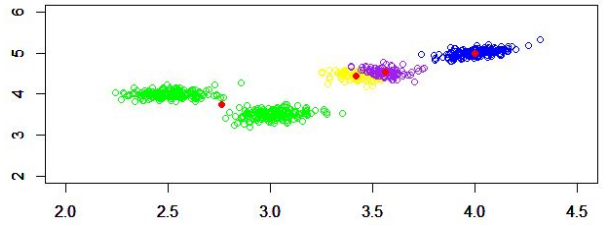

Imagem 3: Ilustração do resultado de um ótimo local.

A imagem 3 mostra que o algoritmo mesmo setando o K = 4, é possível não obter os mesmos clusters que o esperado. O algoritmo identificou 4 clusters, porém de maneira errada. Isto deve-se ao critério de parada do algoritmo e também da sua abordagem clássica binária e não uma lógica nebulosa fuzzy.

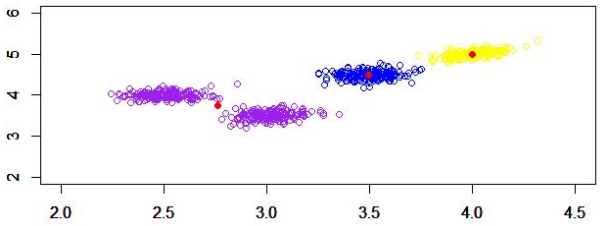

Imagem 4: Ilustração do resultado de um ótimo local.

A imagem 4 mostra que o algoritmo mesmo setando o K = 4, é possível não obter os mesmos clusters que o esperado. E ainda vemos que o algoritmo identificou apenas 3 clusters e não 4 como o esperado. Isto deve-se ao critério de parada do algoritmo e também da sua abordagem clássica binária e não uma lógica nebulosa fuzzy.

O número de iterações perante os teste ficou entre 2 à 15 iterações, considerando o algoritmo rápido para a abordagem proposta.

V. CONCLUSÕES

O algoritmo K-Means foi bastante satisfatório para encontrar a separação dos clusters, mas também foi visto que o algoritmo nem sempre acerta de primeira, podendo convergir para ótimos locais invés do ótimo global.

Para obter uma melhora no algoritmo basta alterar o critério de parada do algoritmo e também a implementação da sua abordagem clássica binária para uma abordagem lógica nebulosa fuzzy.

Utilizando este algoritmo e do conjunto de amostras referente a cada cluster, foi possível obter ótimos resultados e bastante eficiência perante ao algoritmo K-Means.

VI. REFERÊNCIAS

[1] R tutorial, http://www.r-tutor.com/r-introduction, acessado 30/08/2017.

[2] K-means - https://pt.wikipedia.org/wiki/K-means, acessado 17/09/2017.