Relatório: implementando o algoritmo de *fuzzy c-means* em Python

Juan Belieni

2 de novembro de 2022

1 O algoritmo de *fuzzy c-means*

O algoritmo de *fuzzy c-means* é um algoritmo baseado no tradicional algoritmo de *k-means*. Nele, cada ponto pode estar associado a mais de um cluster, com uma certa probabilidade, que deve somar 1.

No algoritmo, uma matriz de particionamento $U \in [0,1]^{C \times P}$ é criada, onde C é o número de clusters e P é o número de pontos. Cada elemento u_{ij} da matriz representa a probabilidade do i-ésimo ponto pertencer ao j-ésimo cluster. Também é definido um vetor $\mathbf{v} \in \mathbb{R}^{C \times M}$, que representa os centróides dos clusters, onde M é a dimensão dos pontos. A clusterização é controlada por um parâmetro $m \in [1, \infty)$, chamado de coeficiente de fuzzificação.

O *fuzzy c-means* é um algoritmo iterativo, e busca minimizar a função objetivo

$$J = \sum_{i=1}^{P} \sum_{j=1}^{C} u_{ij}^{m} \|\mathbf{z}_i - \mathbf{v}_j\|^2, \tag{1}$$

onde $\mathbf{z}_i$ é o i-ésimo ponto e $\mathbf{v}_j$ é o centro do j-ésimo cluster. Para que essa minimização seja feita, o algoritmo define, a cada iteração, novos valores para a matriz de particionamento U e para os centros dos clusters $\mathbf{v}$ da seguinte forma:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|\mathbf{z}_i - \mathbf{v}_j\|}{\|\mathbf{z}_i - \mathbf{v}_k\|} \right)^{\frac{2}{m-1}}}. \tag{2}$$

e

$$\mathbf{v}_j = \frac{\sum_{i=1}^{P} u_{ij}^m \mathbf{z}_i}{\sum_{i=1}^{P} u_{ij}^m}. \tag{3}$$

A cada iteração, o algoritmo verifica se $J^{(k+1)} - J^k < \varepsilon$, onde ε é um parâmetro de tolerância.

2 Implementação

A implementação do algoritmo de *fuzzy c-means* foi feita em Python, utilizando a biblioteca *numpy* para a manipulação de matrizes e vetores.

3 Métodos

4 Resultados