



Ciclo 05 - O sistema não-supervisionado de aprendizado



Fundamentos Machine Learning

[Aprendizado não-supervisionado](#)

[Tarefas de Clusterização](#)

[K-Means](#)

[Treinamento](#)

[Premissas](#)

[Métricas de avaliação de clusters](#)

[Elbow Method](#)

[Maldição da dimensionalidade no K-Means](#)

Aprendizado não-supervisionado

No aprendizado não-supervisionado, temos um conjunto de dados de treinamento com características observadas do fenômeno, mas não temos uma variável numérica ou um rótulo de classificação das observações. Podemos pensar em a aprendizado não-supervisionado como um jogo onde não é passado nenhuma instrução ou regra, e portanto nós temos que descobrir como jogar o jogo sozinhos, em outras palavras, aprendizado não supervisionado é um tipo de aprendizado no qual o algoritmo tenta aprender os padrões ou relações nos dados sem que lhe tenha passado explicitamente o que procurar.



No **aprendizado não-supervisionado**, nosso único objetivo é **descobrir padrões**. Padrões, também chamados de segmentações, são comportamentos ou características similares de pessoas ou produtos dentro de um determinado grupo.

Tarefas de Clusterização

A técnica de clusterização agrupa objetos similares em grupos que apresentam padrões de comportamento e/ou características parecidas. Esses grupos de objetos similares entre si formam padrões que são usados para entender melhor estruturas de dados, segmentação de clientes por características ou comportamentos similares.



A clusterização cria grupos de objetos com comportamentos similares, ela nos permite criar os próprios rótulos para cada um dos grupos.

Exemplos:

1. **Segmentação de clientes:** usando dados de transações, a clusterização pode identificar grupos de clientes com comportamento de compra semelhantes, como frequência de compra, valor de compra ou produtos comprados. Esses grupos podem ser usados para personalizar ofertas, campanhas de marketing ou para entender melhor as preferências dos clientes.
2. **Identificação de fraudes:** a clusterização pode ser usada para detectar atividades suspeitas ou fraudes em transações financeiras. Por exemplo, grupos de transações atípicas ou fora do padrão podem ser identificados como potenciais fraudes.
3. **Segmentação de imagens:** a clusterização pode ser usada para segmentar imagens em regiões com características similares. Por exemplos, em imagens de satélite, a clusterização pode ser usada para segmentar áreas com características similares, como uso do solo, cobertura vegetal ou densidade populacional.

4. **Análise de textos:** a clusterização pode ser usada para agrupar textos similares com base em tópicos ou temas. Isso pode ser usado para análise de sentimentos, identificação de tendências ou para entender melhor a opinião pública sobre determinado assunto.

Algoritmos de clusterização

1. K-Means
2. Hierarchical Clustering Analysis (HCA)
3. DBSCAN
4. Meanshift
5. Gaussian Mixture Model (GMM)



A clusterização é uma forma de agrupar os dados, mas como não possuímos os rótulos não é possível dar o feedback ao algoritmo e de modo que não conseguimos ajustar o aprendizado e portanto o **aprendizado em clusterização não é garantido**.

K-Means

O K-Means é um algoritmo de clusterização (agrupamento) que divide um conjunto de dados em k grupos (clusters) baseados nas suas similaridades. Cada grupo é representado por um centroide, que é ponto médio de todos os pontos do grupo. O algoritmo funciona iterativamente, alocando cada ponto ao cluster mais próximo e recalculando os centroides até que a convergência seja alcançada.

Treinamento

O primeiro passo para utilizar o K-Means é informar ao algoritmo quantos cluster nós achamos que o conjunto de dados possui. Em conjuntos de dados com 2 ou 3 features é relativamente simples chutar um bom valor, pois é possível plotar os dados em um gráfico e ver a posição que cada ocupa. No gráfico ao abaixo é dá para observar que $k = 3$ é um valor razoável.

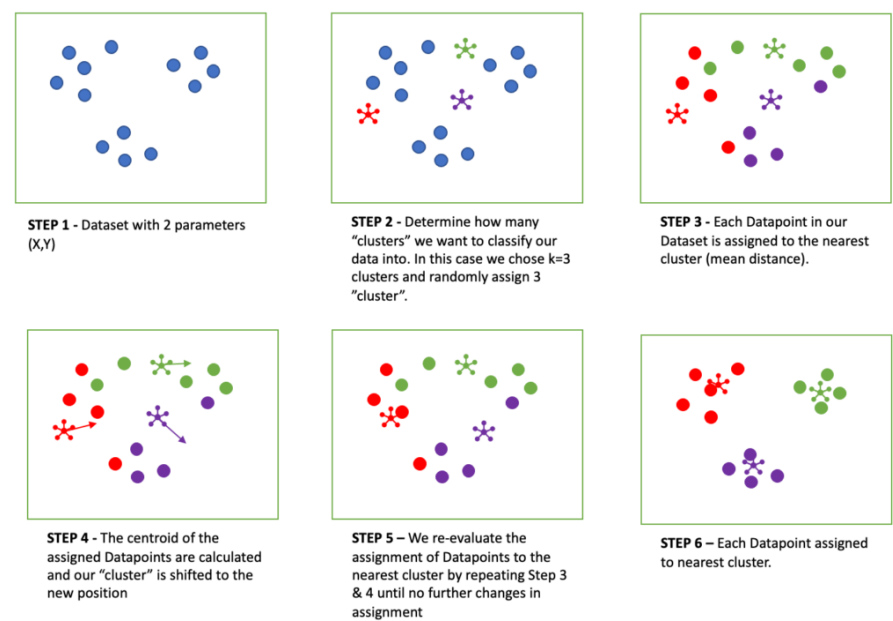
https://external-content.duckduckgo.com/iu/?u=https%3A%2F%2Fmiro.medium.com%2Fmax%2F1200%2F1*rw8IUza1dbffBhiA4i0GNQ.png&f=1&nofb=1&ipt=44bf518ae6deba8bce6193ca310ad9ba348bf95208d93b06ebfdf0f3a2c85170&ipo=images

Após escolhermos o valor de k , o algoritmo escolhe e insere no gráfico k pontos em locais aleatórios, esses pontos são chamados de centroides. O K-Means inicialmente calcula as distâncias de cada ponto em relação a cada um dos centroides, após o cálculo de todas as distâncias, o algoritmo associará os pontos que ficarem mais próximos de um centroide a ele, em outras palavras, os datapoints são atribuídos ao cluster definido pelo centroide mais próximo.

Após encontrarmos os k clusters (passo 3 na imagem), devemos calcular a posição da centroide definido pelo o cluster e mover o antigo centroide para a posição calculada.

Mais uma vez (com os centroides na nova posição), calculamos a distância entre os novos centroides e todos os datapoints e, novamente atribuímos os pontos ao cluster definido pelo centroide mais próximo. Com os novos clusters definidos, recalculamos a posição dos centroides dos clusters e movemos os antigos centroides para a nova posição.

O processo é repetido até que os centroides não mudem mais de posição, ou seja, nenhum datapoint mudará mais de cluster.



Passos:

1. **Passo 01:** Carregue os dados.
2. **Passo 02:** Inicialize os k centroides aleatoriamente no espaço de dados.
3. **Passo 03:** Calcule a distância entre os pontos e os k centroides.
4. **Passo 04:** Atribua os pontos aos centroides mais próximos.
5. **Passo 05:** Calcule o ponto médio dentro de cada cluster e reposicione o centroide para o centro.
6. **Passo 06:** Refaça os passos de 3 a 5 até não haver mais mudanças de pontos entre cluster.

Premissas

1. **O número de clusters k é conhecido.**
2. **Os clusters têm forma esféricas e são igualmente densos:** o algoritmo K-Means assume que todos os clusters têm forma esférica e têm a mesma densidade.
3. **Todos os pontos de dados pertencem a pelo menos um cluster:** o algoritmo assume que cada ponto pertence a exatamente um cluster.
4. **A variância dos dados é a mesma em todos os clusters:** o algoritmo pressupõe que os clusters têm tamanhos semelhantes e distribuições semelhantes.
5. **Os pontos de dados são independentes e idênticos:** isso significa que cada datapoint é tratado como uma observação aleatória e não relação entre eles.

Vantagens

- Fácil de entender e implementar
- Eficiente computacionalmente
- Funciona bem com grandes conjuntos de dados

Desvantagens

- Sensível a inicialização dos centroides.
- Não funciona bem com dados em alta dimensionalidade.
- Pode produzir clusters com tamanhos muito diferentes.
- Apresenta baixa performance para clusters não-circulares.

Métricas de avaliação de clusters

Coesão (Compactness / Cohesion)

A coesão mede o quanto os pontos em um cluster estão próximos uns dos outros. Ela é medida pela soma do erro das distâncias quadráticas entre cada ponto do cluster e o seu centroide dentro do próprio cluster. Essa media de coesão é conhecida como “**Within-Cluster Sum of Squares**”, do inglês, WCSS.

$$\text{WCSS}_j = \sum_{i=1}^n \|x_i - c_j\|^2$$



Quanto menor o WCSS de um cluster, mais compacto ele é, ou seja, os pontos estão mais próximos uns dos outros

Separação

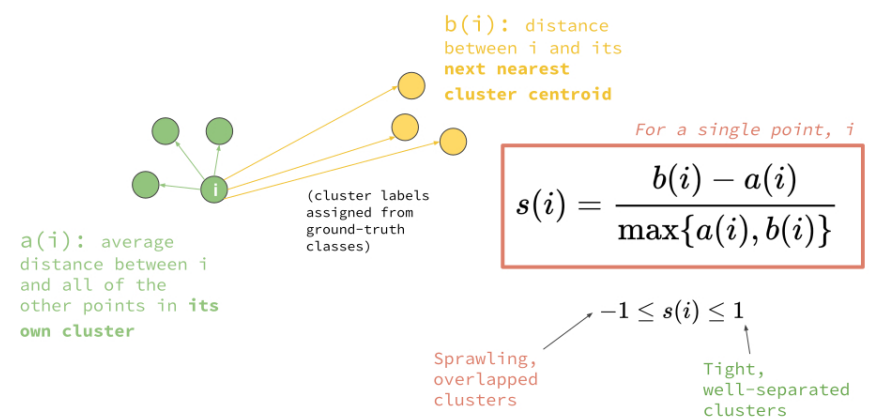
A separação mede o quão distante os pontos em um cluster estão de outros clusters. Ela é medida pela distância entre os centróides dos clusters. Quanto maior a distância entre os centroides, mais separados os clusters estão.

Em outras palavras, a separação mede o quão bem definidos os clusters estão uns dos outros. Um baixo valor de separação indica que os clusters podem estar muito próximos uns dos outros, o que pode tornar difícil a sua distinção. Por outro lado, um valor alto de separação sugere que os clusters são bem definidos e distintos uns dos outros.

Silhouette Score

O Silhouette Score é uma medida de avaliação para medir a qualidade de um agrupamento. Ele é calculado para cada observação e mede o quão bem ela se encaixa em seu cluster atual, em comparação com os outros clusters.

O Silhouette Score varia de -1 a 1 , onde o valor mais próximo de 1 indica que a observação está bem ajustada ao seu cluster e mal ajustada aos outros, e um valor mais próximo de -1 indica que uma observação está mal ajustada ao seu cluster e bem ajustada aos outros clusters. Um valor próximo de 0 indica que a observação pode ser atribuída a qualquer um deles.



$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

- Se $b_i > a_i \Rightarrow s_i > 0$ \therefore ponto se **ajusta bem** ao cluster atual
- Se $b_i < a_i \Rightarrow s_i < 0$ \therefore ponto se **ajusta mal** ao cluster atual



O Silhouette Score geralmente é usado em conjunto com outras medidas de avaliação, como o WCSS, para escolher o número ideal de clusters para um conjunto de dados.

Nos nossos problemas de clusterização nós usamos o silhouette score médio para avaliar a performance do algoritmo. Para calcular o silhouette score médio basta somar o valor do silhouette score para cada ponto e dividir tudo pelo número pontos.

$$SS = \frac{\sum_{i=1}^N s_i}{N} = \frac{1}{N} \sum_{i=1}^N \left(\frac{b_i - a_i}{\max(a_i, b_i)} \right)$$

Elbow Method

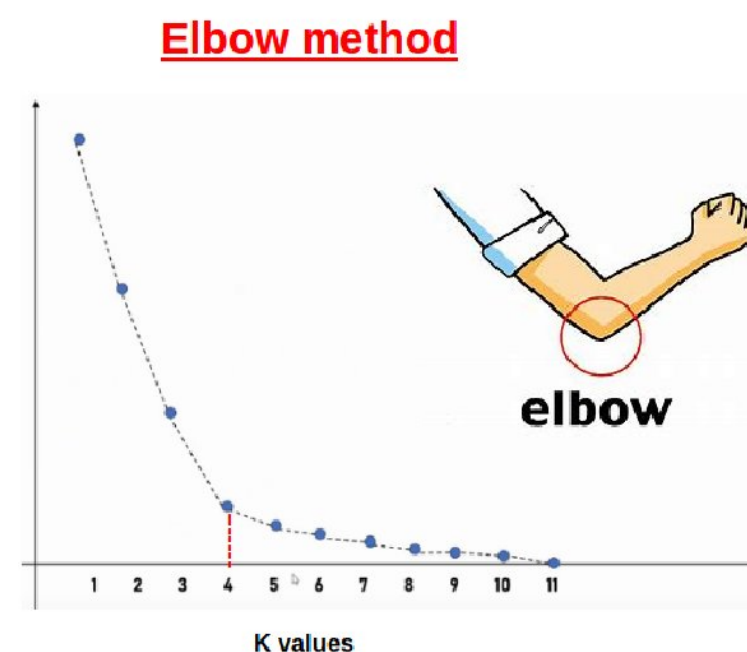
Como escolher o melhor valor de K?

É possível encontrar o valor de K para o algoritmo K-Means de maneira empírica, ou seja, variando o valor de K e observado a métrica de avaliação dos clusters, de modo a determinar o valor de K que resulte na melhor performance. Esse método empírico é chamado de **Elbow Method** ou método de cotovelo.

O Elbow Method é um método que ajuda a determinar o número ideal de clusters (ou grupos) em uma análise de clusterização. Ele é chamado de Elbow Method porque a curva formada pelo número de clusters em relação à medida de coesão se assemelha a um braço dobrado no cotovelo.

Os 4 passos de funcionamento do Elbow Method

1. Realize a clusterização dos dados para diferentes valores de K .
2. Para cada valor de K , calcule a soma dos quadrados das distâncias de cada ponto ao centro do cluster mais próximo (coesão ou WCSS).
3. Trace um gráfico com o número de clusters no eixo x e a soma das distâncias no eixo y.
4. O objetivo é escolher o valor de K que resulte em uma redução significativa da soma das distâncias em relação ao número de clusters. Isso é indicado por uma curva que se assemelha a um braço dobrado em um cotovelo, onde o ponto de dobra indica o valor ideal de K .




Maldição da dimensionalidade no K-Means

Algebra Linear SVD e PCA

Since clustering algorithms such as K-means operate only on distances, the right distance metric to use (theoretically) is the distance metric which is preserved by the dimensionality reduction. This way, the dimensionality reduction step can be seen as a computational shortcut to cluster the data in a lower dimensional space. (also to avoid local minima, etc)


How do I know my k-means clustering algorithm is suffering from the curse of dimensionality?

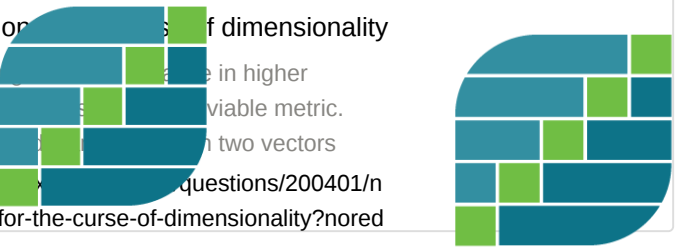
I believe that the title of this question says it all.

 <https://stats.stackexchange.com/questions/232500/how-do-i-know-my-k-means-clustering-algorithm-is-suffering-from-the-curse-of-dim?noredirect=1&lq=1>

Need more intuition for the curse of dimensionality


People despise using Euclidean distance in higher dimensional spaces as it is not a viable metric. People argue that this is because the distance between two vectors


 <https://stats.stackexchange.com/questions/200401/need-more-intuition-for-the-curse-of-dimensionality?noredirect=1&lq=1>



Why is Euclidean distance not a good metric in high dimensions?

I read that 'Euclidean distance is not a good distance in high dimensions'. I guess this statement has something to do with the curse of dimensionality, but what exactly? Besides, what is 'high


 <https://stats.stackexchange.com/questions/99171/why-is-euclidean-distance-not-a-good-metric-in-high-dimensions/>



Exemplos geométricos artigo acima


Calculator Suite - GeoGebra

Interactive, free online calculator from GeoGebra: graph functions, plot data, drag sliders, create triangles, circles and much more!

 <https://www.geogebra.org/calculator/advwz8ur>

3D Calculator - GeoGebra

Free online 3D grapher from GeoGebra: graph 3D functions, plot surfaces, construct solids and much more!

 <https://www.geogebra.org/3d/e9ydck3s>