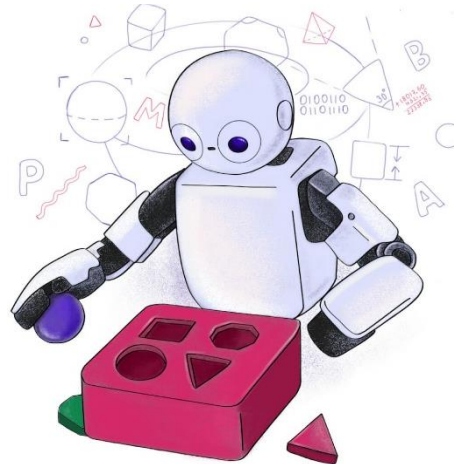


TP555 - Inteligência Artificial e Machine Learning: *k-Médias*



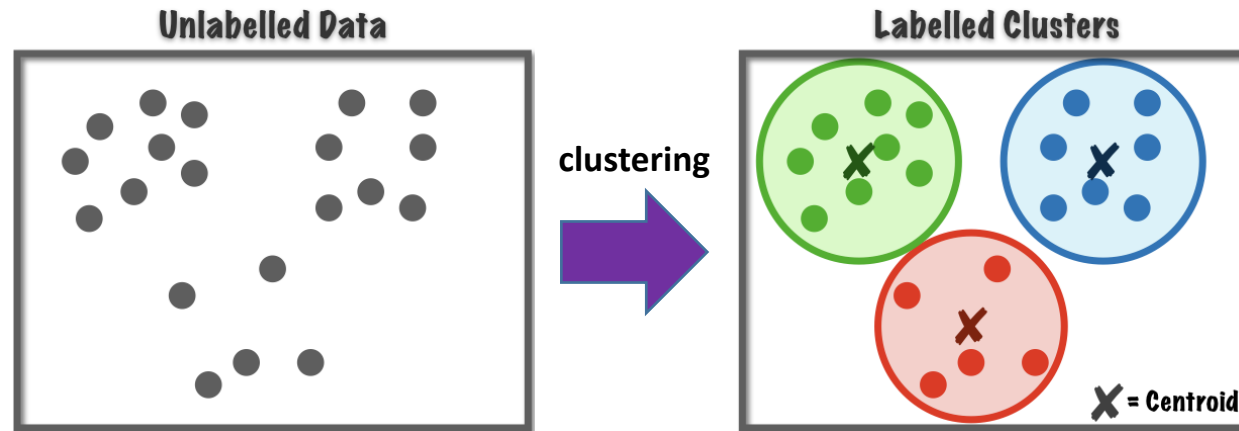
Inatel

Felipe Augusto Pereira de Figueiredo
felipe.figueiredo@inatel.br

Recapitulando

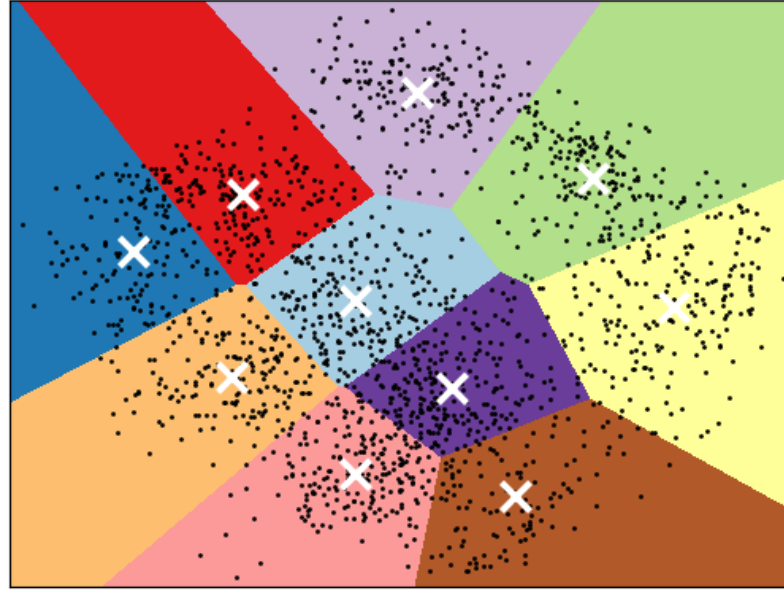
- Até o momento, todos os algoritmos que aprendemos seguiam o paradigma do aprendizado supervisionado.
- Hoje, falaremos sobre ***clustering***, que são algoritmos de ***aprendizado não-supervisionado*** que visam criar agrupamentos de dados (chamados de ***clusters*** ou ***grupos***) segundo seu grau de semelhança.
- Em seguida, aprenderemos sobre o algoritmo chamado de ***k-Médias*** (ou ***k-Means***, em inglês) que é um dos algoritmos mais simples de ***clustering***.
- O vídeo sobre esta apresentação pode ser encontrado neste [link](#).

Motivação



- O que podemos fazer se não tivermos informações sobre as classes (i.e., rótulos) a que pertencem os exemplos de entrada?
- Veremos que informações úteis podem ser obtidas mesmo de exemplos cujas classes não são conhecidas.
- Enquanto o ***aprendizado supervisionado*** se concentra na ***indução de classificadores***, o ***aprendizado não-supervisionado*** está interessado em ***descobrir propriedades úteis*** dos dados disponíveis.

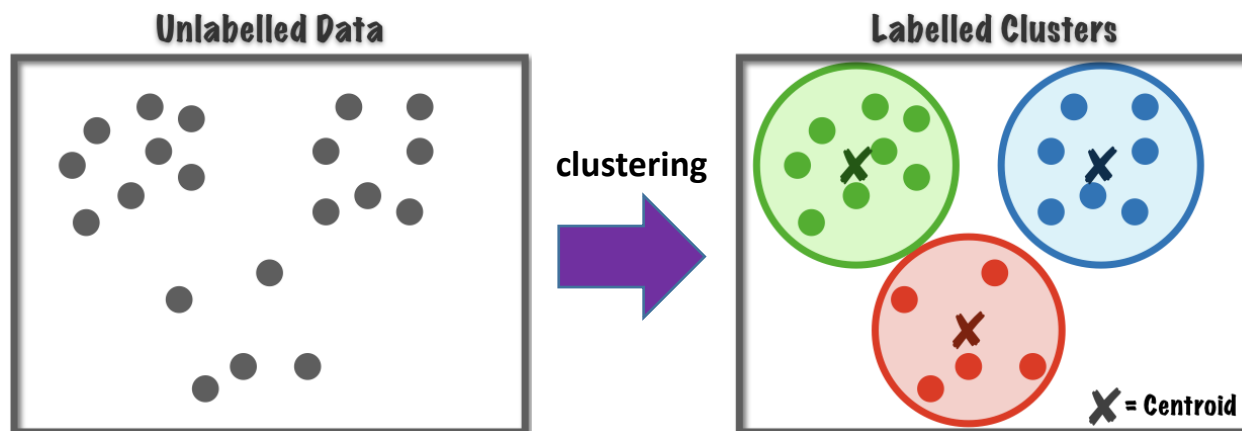
Motivação



- Talvez a tarefa mais popular dos algoritmos deste paradigma seja a procura por **grupos** (chamados **clusters**) de **exemplos semelhantes**.
- Os **centroides** desses **clusters** podem então ser usados como
 - centros para redes Bayesianas ou de Função de base radial (RBF),
 - estimativas de valores de atributos desconhecidos (ou ausentes),
 - ferramentas de visualização de dados multidimensionais,
 - auxiliares para criação de classificadores mais simples.

Identificação de clusters

- A tarefa fundamental do ***aprendizado não-supervisionado*** é a ***identificação de clusters***.
- Nessa tarefa, a entrada é um conjunto de vetores de atributo (i.e., exemplos), ***mas sem rótulos***.
- A saída é um conjunto com os ***clusters*** a que pertencem cada um dos exemplos.

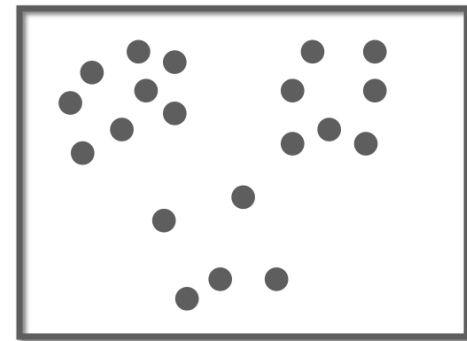


OBS.: A identificação visual de clusters em um espaço bidimensional é fácil, mas em quatro ou mais dimensões isso já não é mais possível. Nesses casos, apenas algoritmos de identificação de clusters conseguem agrupar os dados.

Como representar os clusters?

- Para realizar a identificação, temos que decidir como os **clusters** serão representados.
- Existem algumas opções como a localização dos clusters, tamanhos, limites, etc.
- Porém, a abordagem mais simples usa os **centroides** (i.e., centros) dos clusters.
- Se os atributos forem numéricos, o **centroide** é obtido através das **médias individuais dos atributos**.
- Por exemplo, suponhamos os seguintes **vetores de atributos** em um espaço bidimensional: (2, 5), (1, 4), (3, 6).
- Nesse caso, o **centroide** é representado pelo vetor (2, 5), pois
 - A média do primeiro atributo é $\frac{2+1+3}{3} = 2$.
 - A média do segundo atributo é $\frac{5+4+6}{3} = 5$.
- Se os atributos não forem numéricos, devemos transformá-los em numéricos.

Como devem ser os clusters?



- Os clusters ***não devem se sobrepor***: cada exemplo deve pertencer a um e apenas a um cluster.
- Porém, dentro do mesmo cluster, os exemplos devem estar relativamente próximos uns dos outros e distantes dos exemplos dos outros clusters.
- Aí surge uma dúvida. Quantos clusters um conjunto de exemplos contém?
- Na figura acima conseguimos identificar três clusters.
- No entanto, o número de opções existentes não se limita a essa única possibilidade:
 - em um extremo, todo o conjunto de dados pode ser pensado como formando um grande cluster;
 - no outro, cada exemplo pode ser visto como representando seu próprio cluster de um único exemplo.
- Implementações práticas geralmente evitam esse problema pedindo ao usuário que forneça o número de clusters.

Medindo distâncias

- Algoritmos para identificação de clusters geralmente precisam de um mecanismo para **avaliar a distância** entre um exemplo e o centroide de um cluster.
- Uma forma de fazer isso quando os atributos são contínuos é usar a **distância euclidiana** entre os dois vetores.
- Para exemplos com atributos discretos ou uma mistura de ambos, usamos uma equação mais geral para calcular as distâncias:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^K d_a(x_i, y_i)},$$

onde K é o número de atributos, $d_a(x_i, y_i) = (x_i - y_i)^2$ para atributos contínuos e $d_a(x_i, y_i) = 0$ se $x_i = y_i$ e $d_a(x_i, y_i) = 1$ se $x_i \neq y_i$ para atributos discretos.

A qual cluster um exemplo deve pertencer?

- Vamos supor que existam N clusters cujos centroides são denotados por $\mathbf{c}_i, \forall i \in (1, N)$.
- Um exemplo \mathbf{x} tem uma certa distância $d(\mathbf{x}, \mathbf{c}_i)$ para cada centroide.
- Se $d(\mathbf{x}, \mathbf{c}_p)$ é a menor dessas distâncias, então, é natural colocarmos \mathbf{x} como pertencente ao centroide \mathbf{c}_p , ou seja, o p -ésimo cluster.
- Portanto, escolhemos a **menor distância** para definir a qual cluster um exemplo pertence.

k-Means

- É talvez o algoritmo mais simples para identificação de clusters.
- O " k " no nome denota o número **solicitado** de clusters, ou seja, o número de clusters é um parâmetro definido pelo usuário.
- O pseudocódigo do algoritmo é mostrado abaixo.

Entradas: conjunto de exemplos e número de clusters, k .

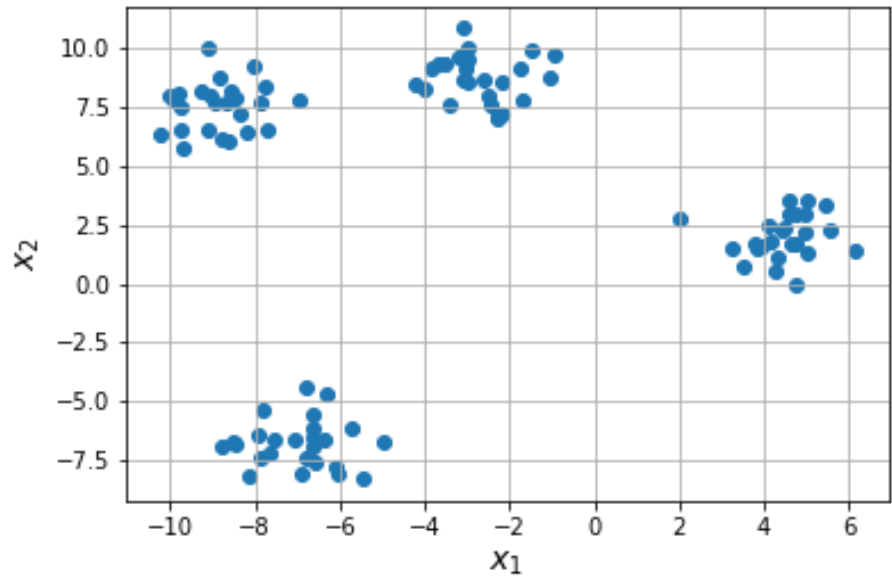
1. **Defina** k centroides iniciais (os centroides representam e definem o número de clusters).
2. **Repita**
 - a) Calcule a distância de cada exemplo, x , para cada um dos k centroides e atribua cada exemplo ao cluster mais próximo.
 - b) Calcule o novo centroide de cada cluster.
3. **Enquanto** as posições dos centroides continuarem mudando.

- O algoritmo garantidamente chega a uma situação em que cada exemplo se encontra no cluster mais próximo, de modo que, a partir deste momento, os centroides não mudem mais.

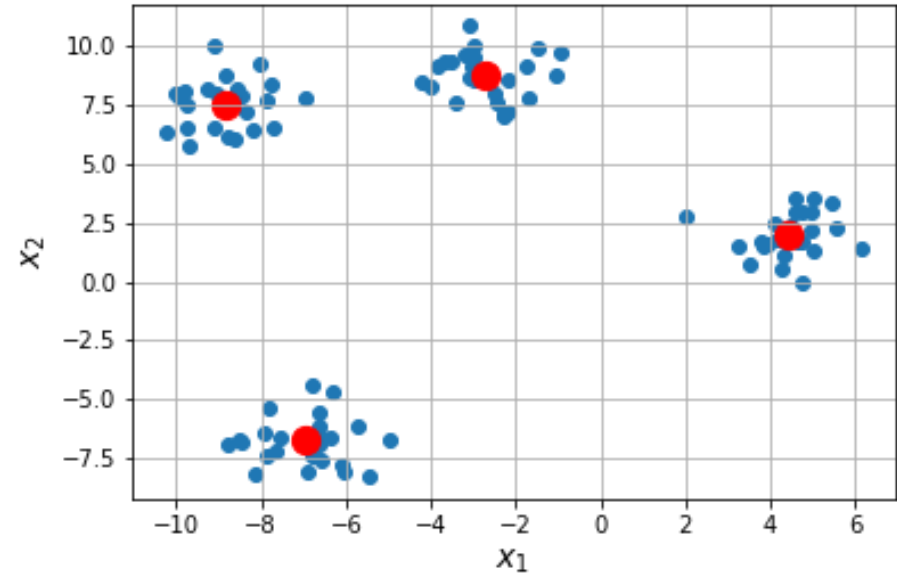
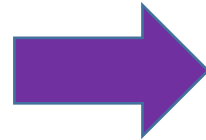
Como inicializar os centroides?

- O procedimento mais simples para inicializar os centroides é escolher k exemplos de treinamento aleatórios e os considerar como os centroides iniciais.
- Os clusters iniciais são então criados associando cada um dos exemplos ao seu centroide mais próximo.
- O número de transferências de um exemplo de um cluster para outro depende dos centroides iniciais.
- Se os centroides iniciais já forem perfeitos, nenhum exemplo precisa ser atribuído a outro cluster e o algoritmo é encerrado.
- Portanto, a inicialização é importante no sentido de que um ponto de partida melhor garante que a solução seja encontrada mais rápido.

SciKit-Learn: k-Means

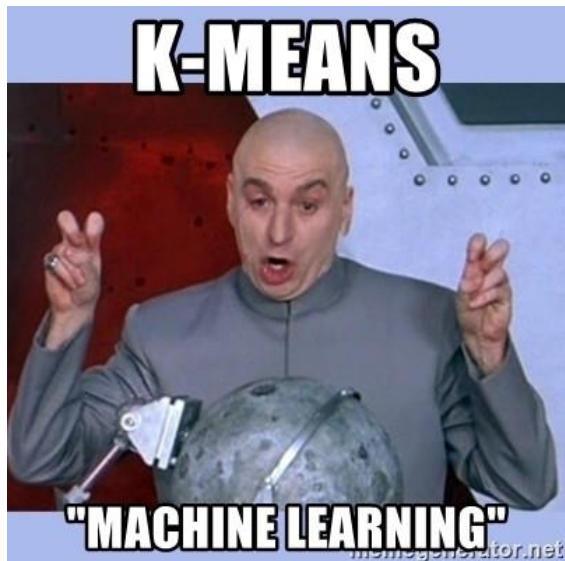


k-Means



[Exemplo: kmeans_example.ipynb](#)

Obrigado!



k-means be like:



K-MEANS CLUSTER

