

# Problemas Clássicos da Computação

## Aprendizado Não Supervisionado

Felipe Augusto Lima Reis

[felipe.reis@ifmg.edu.br](mailto:felipe.reis@ifmg.edu.br)



**INSTITUTO  
FEDERAL**  
Minas Gerais

# Sumário



- 1 Introdução
- 2 Similaridade
- 3 Clustering
- 4 Redes Neurais

# INTRODUÇÃO

# Introdução

- Muitos algoritmos de aprendizado de máquinas necessitam de dados rotulados para treinamento
  - Rótulos são úteis, porém podem ser difíceis ou caros de serem produzidos;
  - Muitos rótulos são gerados manualmente, por um especialista humano, o que encarece a construção de uma base de dados;
- Para casos onde os rótulos inexistem, são úteis os algoritmos de aprendizado não supervisionados.

# Introdução

- Aprendizado não supervisionado pode ser utilizado em tarefas de classificação
  - Tarefas de regressão não podem ser realizadas por esse tipo de algoritmo;
  - Uma vez que não existem classes corretas, o algoritmo irá avaliar a **similaridade** dos elementos;
  - As similaridades serão utilizadas para **clusterizar** (agrupar) elementos similares, provendo classificação [Marsland, 2014].

# MÉTRICAS DE SIMILARIDADE

# MÉTRICAS DE SIMILARIDADE PARA VARIÁVEIS NUMÉRICAS

# Métricas de Similaridade

- Segundo [Cha, 2007], podemos dividir as similaridades nas seguintes famílias:
  - 1 Família Minkowski  $L_p$ ;
  - 2 Família  $L_1$ ;
  - 3 Família de Interseção;
  - 4 Família de Produto Interno;
  - 5 Família de Fidelidade ou Família Squared-chord;
  - 6 Família Quadrática  $L_2$  ;
  - 7 Família de Entropia de Shannon;
  - 8 Combinações;



# Métricas de Similaridade

## 1 Família Minkowski $L_p$

- Família originada a partir da distância Euclidiana;
  - A distância City Block<sup>1</sup>, corresponde à distância absoluta entre coordenadas cartesianas;
  - A generalização da distância City Block corresponde à distância Minkowski [Cha, 2007].

Table 1. $L_p$ Minkowski family		
1. Euclidean $L_2$	$d_{Euc} = \sqrt{\sum_{i=1}^d  P_i - Q_i ^2}$	(1)
2. City block $L_1$	$d_{CB} = \sum_{i=1}^d  P_i - Q_i $	(2)
3. Minkowski $L_p$	$d_{Mk} = \sqrt[p]{\sum_{i=1}^d  P_i - Q_i ^p}$	(3)
4. Chebyshev $L_\infty$	$d_{Cheb} = \max_i  P_i - Q_i $	(4)

Fonte: [Cha, 2007]

<sup>1</sup>Também conhecida como Manhattan Distance, Distância  $L_1$  ou Taxicab metric.

# Métricas de Similaridade

## 2 Família $L_1$

- Família utilizada para cálculo da diferença absoluta ( $L_1$ );
  - Sørensen e Canberra são destaques da classe, e usadas na área de biologia;
  - Gower realiza escala do espaço vetorial no espaço normalizado para cálculo da distância [Cha, 2007].

## 3 Família de Interseção

- A interseção entre duas Funções Densidade de Probabilidade são muito usadas para similaridades onde não há sobreposição;
  - A maioria das distâncias do grupo podem ser transformadas em distâncias da família  $L_1$  [Cha, 2007].

# Métricas de Similaridade

## 4 Família de Produto Interno

- Família de similaridades onde há produto interno entre os elementos  $P$  e  $Q$ ;
  - O produto interno normalizado é chamado coeficiente Cosseno, devido ao ângulo entre os dois vetores;
  - Dice é relacionado a uma série de outras medidas, como Sørensen e Czekanowski, e é frequentemente usado para taxonomias biológicas [Cha, 2007].

## 5 Família de Fidelidade

- A soma da média geométrica é conhecida como Similaridade de Fidelidade, e métricas relacionadas a essa medida podem ser agrupadas nesta classe [Cha, 2007].

# Métricas de Similaridade

## 6 Família Quadrática $L_2$

- Família agrupa métricas usando a distância Euclidiana Quadrática;
  - Formas alternativas das distâncias de Jaccard e Dice pertencem a essa família [Cha, 2007];

## 7 Família de Entropia de Shannon

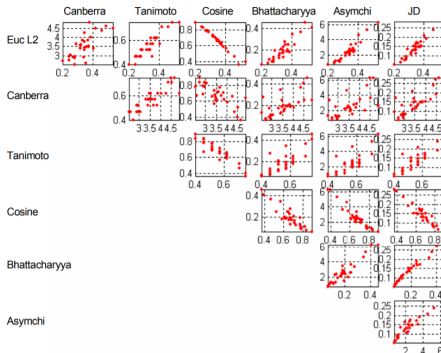
- A família corresponde ao conceito de incerteza ou entropia, proposto por Shannon [Cha, 2007];

## 8 Combinações

- A família contém medidas de distância que contém múltiplas ideias ou medidas [Cha, 2007].

# Métricas de Similaridade

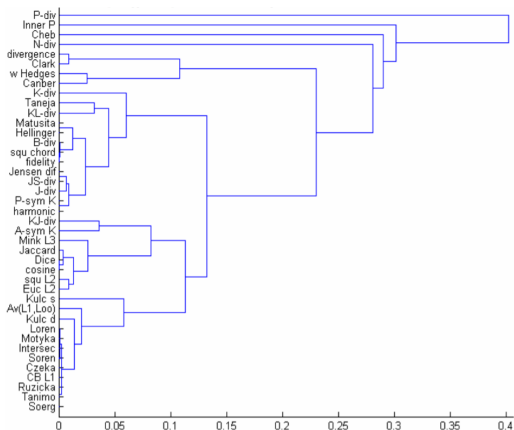
- Podemos indicar a força e a direção entre duas medidas de distâncias pela imagem abaixo
  - Se as distâncias não são similares, o valor tende a 0;
  - Caso contrário, os valores tendem a 1.



Fonte: [Cha, 2007]

# Métricas de Similaridade

- Métricas de similaridade podem ainda serem agrupadas com o auxílio do dendrograma abaixo.



Fonte: [Cha, 2007]

# MÉTRICAS DE SIMILARIDADE EM DADOS CATEGÓRICOS

# Métricas de Similaridade - Dados Categóricos

- Segundo [Santos, 2014], as medidas de similaridade em dados categóricos podem ser classificadas em 3 tipos:
  - ① Medidas que atribuem valor 1 para *matching* e 0 para *mismatching*;
  - ② Atribuem valor 1 para para *matching* e valores entre 0 e 1 para *mismatching*;
  - ③ Medidas que atribuem valores entre 0 e 1 quando ocorrem *matching* e *mismatching*;



# Métricas de Similaridade - Dados Categóricos

- São exemplos de métricas de similaridade para dados categóricos: [Santos, 2014]
  - Métricas do Tipo 1
    - Similaridades de Gower (GOW);
    - Similaridades de Eskin (ESK)
    - Similaridades de Gambaryan (GAM);
  - Métricas do Tipo 2
    - *Inverse Occurrence Frequency* (IOF);
  - Métricas do Tipo 3
    - Similaridade de Lin (LIN);
    - Similaridade de Smirnov (SMI).

# CLUSTERING

# Clustering

- **Clusterização** (*clustering*) ou **agrupamento** é uma das técnicas mais amplamente utilizadas para análise exploratória de dados [Shalev-Shwartz and Ben-David, 2014];
  - O método busca agrupar elementos similares e separar elementos dissimilares;
  - A distância entre elementos de um mesmo grupo devem ser a menor possível, enquanto a distância entre elementos de grupos distintos devem ser a maior possível<sup>2</sup>.

---

<sup>2</sup>Esse mesmo objetivo foi apresentado previamente na seção de Redução de Dimensionalidade.

# Clustering

- Uma das dificuldades da clusterização é que o processo pode ser entendido como uma relação de equivalência<sup>3</sup>
  - Dentre as características das relações de equivalência, destaca-se a transitividade;
  - Para transformar uma relação não transitiva em um relação transitiva é necessário adicionar novos elementos<sup>4</sup>;
  - No entanto, a medida em que novos elementos são adicionados à relação, esta precisa ser revisada, para verificar se os novos elementos não irão causar efeitos colaterais;
  - Novos passos podem ser necessários até que a relação se torne de equivalência ou que um limite de passos seja executado e o algoritmo termine [Shalev-Shwartz and Ben-David, 2014].

---

<sup>3</sup>Em matemática discreta, uma relação de equivalência deve ser simétrica, reflexiva e transitiva.

<sup>4</sup>Conceitualmente, esses elementos fazem parte de um fecho transitivo.

# Clustering

- O método mais simples para criação de *clusters* é utilizando **Algoritmos de Clusterização Baseados em Ligação**<sup>5</sup>;
- Outro método popular é a definição de uma função de custo, com objetivo de encontrar uma partição (*cluster*) de menor custo possível [Shalev-Shwartz and Ben-David, 2014]
  - Nesta técnica, destaca-se o algoritmo *k-means*;
  - Outros algoritmos similares, como *k-medoides*, *k-median* e *k-modes* também são utilizados.

---

<sup>5</sup>Tradução direta do inglês: *Linkage-Based Clustering Algorithms*.

# Linkage-Based Clustering Algorithms

- Esses algoritmos realizam em uma sequência de iterações;
- Começam com um agrupamento trivial, considerando cada ponto no conjunto de dados como um *cluster* de um único elemento;
- Repetidamente, adicionam *clusters* “mais próximos”, fazendo fusão de *clusters*
  - Consequentemente, o número de *clusters* diminui a cada iteração;
  - Parâmetros são utilizados para definir a distância máxima entre *clusters* e limitar o número máximo de iterações [Shalev-Shwartz and Ben-David, 2014].

# Linkage-Based Clustering Algorithms

- A distância  $d$  entre elementos avaliados pelos algoritmos de clusterização podem ser calculadas de diversas formas:  
[Shalev-Shwartz and Ben-David, 2014]
  - **Single Linkage Clustering**: a distância entre *clusters* é definida como a distância mínima entre membros de dois *clusters*;
  - **Average Linkage Clustering**: a distância entre *clusters* é definida como a distância média entre um ponto em um dos *clusters* e um ponto no outro *cluster*;
  - **Max Linkage Clustering**: a distância entre *clusters* é definida como a distância máxima entre seus elementos.

# Linkage-Based Clustering Algorithms

- Os algoritmos de clusterização baseados em ligação são classificados como **aglomerativos**
  - Iniciam seu processo a partir de dados fragmentados;
  - *Clusters* adicionam novos elementos à medida em que o algoritmo é executado.
- São critérios de parada dos algoritmos de clusterização: [Shalev-Shwartz and Ben-David, 2014]
  - Número fixo de *clusters*;
  - Distância máxima entre *clusters*.

---

Nota: Além dos algoritmos aglomerativos, existem ainda algoritmos divisivos, no qual o procedimento inicia com um *cluster* de tamanho máximo, que é dividido durante as iterações.



# Algoritmo $k$ -means

- O algoritmo  $k$ -means é um método de clusterização com objetivo de particionar  $n$  elementos em  $k$  grupos, de modo que cada elemento pertença ao grupo mais próximo da média
  - É definida uma função de custo e cada cluster deve ter custo mínimo;
  - O problema de clusterização é transformado em um problema de otimização;
  - O problema pode ser classificado como NP-difícil, porém, existem heurísticas comumente empregadas para solução mais rápida [Shalev-Shwartz and Ben-David, 2014].

# Algoritmo $k$ -means

- O algoritmo  $k$ -means requer a definição à priori da quantidade de *clusters* que serão utilizados para separação dos grupos;
- O algoritmo pode ser dividido nas seguintes etapas:
  - 1 Inicialização;
  - 2 Atribuição de Elementos aos *Clusters*;
  - 3 Movimentação de Centroides;
  - 4 Otimização dos Centroides;

# Algoritmo $k$ -means

- Inicialização

- São escolhidos o número de *clusters*  $k$ ;
- São escolhidas  $k$  posições aleatórias no espaço de dados;
- Os centros de cada um dos  $k$  *clusters* são associadas às  $k$  posições aleatórias escolhidas
  - Os pontos centrais dos *clusters* são chamados de **centroides**.

- Atribuição de Elementos aos *Clusters*

- Para cada elemento do conjunto de dados, são computadas as distâncias (Euclidianas) em relação aos centroides;
- Cada elemento é atribuído ao centroide mais próximo [Marsland, 2014].

# Algoritmo *k*-means

- **Movimentação de Centroides**
  - Após atribuição de elementos, a posição dos centroides é recalculada;
  - O novo ponto médio é definido como o valor médio entre os elementos do *cluster* [Marsland, 2014].



Fonte: [Santana, 2017]

# Algoritmo *k*-means

- Otimização dos Centroides
  - O algoritmo executa repetidamente a atribuição de elementos ao *cluster* e a movimentação de centroides;
  - O algoritmo finaliza quando o centro do *cluster* para se mover ou quando o algoritmo atinge algum critério de parada.
- Avaliação de Desempenho e Uso
  - Após o término do aprendizado, o algoritmo pode ser avaliado em um conjunto de testes para análise de desempenho;
  - O algoritmo também pode ser aplicado diretamente a uma situação real, de forma a classificar elementos [Marsland, 2014].

# Algoritmo $k$ -means

- O algoritmo  $k$ -means pode ser resumido em:

---

## The $k$ -Means Algorithm

---

- **Initialisation**

- choose a value for  $k$
- choose  $k$  random positions in the input space
- assign the cluster centres  $\mu_j$  to those positions

- **Learning**

- repeat
    - \* for each datapoint  $\mathbf{x}_i$ :
      - compute the distance to each cluster centre
      - assign the datapoint to the nearest cluster centre with distance
- $$d_i = \min_j d(\mathbf{x}_i, \mu_j).$$
- \* for each cluster centre:
    - move the position of the centre to the mean of the points in that cluster ( $N_j$  is the number of points in cluster  $j$ ):

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \mathbf{x}_i$$

- until the cluster centres stop moving

- **Usage**

- for each test point:
  - \* compute the distance to each cluster centre
  - \* assign the datapoint to the nearest cluster centre with distance

$$d_i = \min_j d(\mathbf{x}_i, \mu_j).$$

---

Fonte: [Marsland, 2014]

# Algoritmo $k$ -means - Escolha Parâmetro $k$

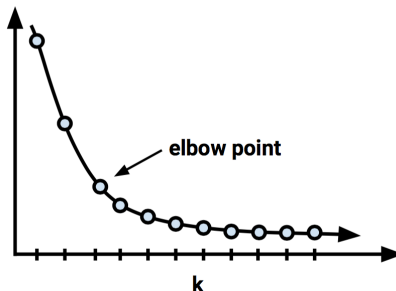
- Como informado previamente, o algoritmo  $k$ -means precisa, obrigatoriamente, de um valor fixo  $k$ ;
  - Em alguns problemas o número de *clusters* já é previamente definido;
  - No entanto, em alguns cenários, o número de *clusters* precisa ser descoberto pelo próprio algoritmo;
- Em problemas sem um valor de  $k$  previamente definido, como escolher, de forma ideal, a quantidade de *clusters*?
  - Uma regra prática é utilizar o **Elbow Method**<sup>6</sup>.

---

<sup>6</sup>Tradução literal: “Método do Cotovelo”.

# Elbow Method

- O **Elbow Method** é uma heurística para determinar o número de *clusters* em uma base de dados;
  - Consiste em variar o número de *clusters*, e testar a variância dos dados;
  - Os registros são plotados em um gráfico e é escolhido o ponto que representa o “cotovelo (ou joelho)” da curva.



Fonte: [Santana, 2017]



## *k-medoids, k-median e k-modes*

- *k-medoids, k-median e k-modes* são variações do *k-means*, usando métricas diferentes para definição dos centroides:
  - *k-medoids*:
    - Utiliza um exemplar (medoid) como centro do *cluster*;
    - Possui como vantagem a melhor interpretabilidade do centro do *cluster*, uma vez que o ponto representa um elemento real, e não um local onde pode não haver nenhum elemento;
  - *k-median*:
    - Calcula a mediana para definição dos centroides;
    - Possui como vantagem a minimização da distância  $L_1$  (1-norm ou City Block) e a menor susceptibilidade a ruídos;
  - *k-modes*:
    - Utiliza a moda para definição do centroide;
    - Para alguns cenários pode ser usado para representar os elementos mais comuns.

# REDES NEURAIS NÃO SUPERVISIONADAS

# Redes Não Supervisionadas



- Redes neurais podem ser utilizadas para tarefas de Aprendizado Não Supervisionado;
- As redes devem descobrir por si só padrões, características, correlações ou categorias a partir dos dados de entrada;
- Os padrões descobertos devem ser reproduzidos na saída da rede [Barreto, 1998].

# Aplicações



- Segundo [Barreto, 1998], as redes não supervisionadas podem ter as seguintes aplicações:
  - Agrupamento (*clustering*): agrupa as entradas em classes, de acordo com a similaridade (retorna apenas uma classe);
  - Prototipação: semelhante ao anterior, porém gera um protótipo (exemplo) da classe apropriada;
  - Codificação: gera uma versão codificada da entrada<sup>7</sup>, mantendo a informação mais relevante;
  - Mapas auto-organizáveis: organiza dimensionalmente dados complexos em grupos, de forma a manter a representação com propriedades relevantes a partir da entrada.

---

<sup>7</sup> possivelmente com menos bits

# Aprendizagem Competitiva

- Neurônios de cada camada competem pelo direito de responder a uma determinada entrada [Coppin, 2004] [Barreto, 1998];
- A resposta é dada pelo neurônio que tiver o maior valor de ativação [Coppin, 2004];
  - Modelo “vencedor-leva-tudo” (*the winner-take-all*, WTA);
  - A unidade vencedora tem sua saída fixada em 1, enquanto as demais são fixadas em 0 [Barreto, 1998];
- Problema comum: unidades mortas (*dead units*)
  - Unidades mal inicializadas podem nunca ganhar competições;
  - Para solução, podem ser criados mecanismos para limitar a vitória de neurônios [Barreto, 1998].

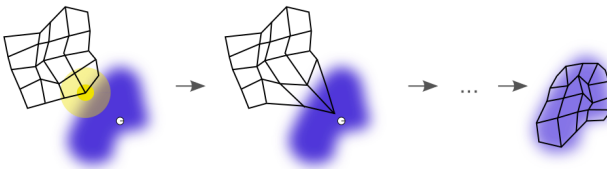
# MAPAS DE KOHONEN

# Mapa de Kohonen

- Corresponde a uma arquitetura de rede proposta por Teuvo Kohonen em 1982 [Coppin, 2004];
  - Também chamado de mapa de *features* (características) auto-organizado;
  - Utiliza o algoritmo “Vencedor-leva-tudo”;
  - Busca subdividir as entradas em *clusters*;
  - A rede é alimentada com vetores de exemplo e a distância Euclidiana entre eles é calculada e os pesos são computadas;
- A rede possui, basicamente, 2 camadas: [Coppin, 2004]
  - Camada de entrada;
  - Camada de *clusterização* (ou Kohonen), que serve à saída.

# Mapa de Kohonen - Treinamento

- O neurônio cujo vetor de pesos é mais parecido (semelhante) à entrada é chamado de melhor unidade de “melhor unidade correspondente” (*Best Matching Unit*, BMU);
- Os pesos do BMU e dos neurônios são ajustados com base no vetor de entrada.



Fonte: [Wikipedia contributors, 2020b]

[Link: versão animada do Mapa de Kohonen](#) [Wikipedia contributors, 2020a]

[Link: animação do treinamento Mapa de Kohonen](#) [Wikipedia contributors, 2020a]



# Referências I



Barreto, G. d. A. (1998).

Redes Neurais não-supervisionadas para processamento de sequências temporais.

PhD thesis, Universidade de São Paulo - Escola de Engenharia de São Carlos.

[Online]; acessado em 07 de Setembro de 2020. Disponível em: [https://teses.usp.br/teses/disponiveis/18/18133/tde-25112015-111953/publico/Dissert\\_Barreto\\_GuilhermeA.pdf](https://teses.usp.br/teses/disponiveis/18/18133/tde-25112015-111953/publico/Dissert_Barreto_GuilhermeA.pdf).



Cha, S.-H. (2007).

Comprehensive survey on distance/similarity measures between probability density functions.

International Journal of Mathematical Models and Methods in Applied Sciences, 1(4):300–307.

[Online]; acessado em 23 de Março de 2021. Disponível em:

<https://www.naun.org/main/NAUN/ijmmas/mmms-49.pdf>.



Coppin, B. (2004).

Artificial Intelligence Illuminated.

Jones and Bartlett illuminated series. Jones and Bartlett Publishers, 1 edition.



Kopec, D. (2019).

Classic Computer Science Problems in Python.

Manning Publications Co, 1 edition.



Marsland, S. (2014).

Machine Learning: An Algorithm Perspective.

CRC Press, 2 edition.

Disponível em: <https://homepages.ecs.vuw.ac.nz/~marslast/MLbook.html>.

# Referências II



Richert, W. and Coelho, L. P. (2013).  
Building Machine Learning Systems with Python.  
Packt Publishing Ltd., 1 edition.



Santana, F. (2017).  
Entenda o algoritmo k-means e saiba como aplicar essa técnica.  
[Online]; acessado em 24 de Março de 2021. Disponível em:  
<https://minerandodados.com.br/entenda-o-algoritmo-k-means>.



Santos, T. (2014).  
Uma Análise comparativa de Medidas de Similaridade para Agrupamento de dados Categóricos.  
PhD thesis, Pontifícia Universidade Católica de Minas Gerais- Programa de Pós-Graduação em Informática.  
[Online]; acessado em 23 de Março de 2021. Disponível em:  
[http://www.biblioteca.pucminas.br/teses/Informatica\\_SantosTRL\\_1.pdf](http://www.biblioteca.pucminas.br/teses/Informatica_SantosTRL_1.pdf).



Shalev-Shwartz, S. and Ben-David, S. (2014).  
Understanding Machine Learning: From Theory to Algorithms.  
Cambridge University Press, 1 edition.  
Disponível em: <http://www.cs.huji.ac.il/shaish/UnderstandingMachineLearning>.



Wikipedia contributors (2020a).  
Mapas de kohonen.  
[Online]; acessado em 07 de Setembro de 2020. Disponível em:  
[https://pt.wikipedia.org/wiki/Mapas\\_de\\_Kohonen](https://pt.wikipedia.org/wiki/Mapas_de_Kohonen).

## Referências III



Wikipedia contributors (2020b).

Self-organizing map.

[Online]; acessado em 07 de Setembro de 2020. Disponível em:

[https://en.wikipedia.org/wiki/Self-organizing\\_map](https://en.wikipedia.org/wiki/Self-organizing_map).