APRENDIZADO NÃO SUPERVISIONADO (*CLUSTERING*)

Inteligência Artificial

André Câmara

UFRPE Universidade Federal Rural de Pernambuco

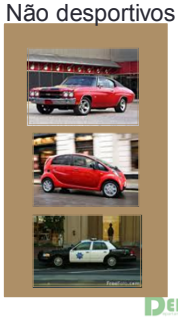
DEINFO Departamento de Estatística e Informática

Alguns slides foram extraídos do material de aula do prof. Valmir Macário (DEINFO/UFRPE)

Qual o agrupamento natural destes dados?

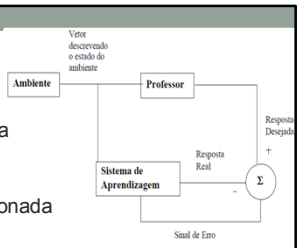
Desportivos

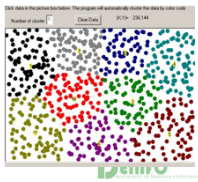
Não desportivos



Contexto

- Aprendizagem Supervisionada
  - Dados rotulados
- Aprendizagem Não Supervisionada
  - Dados Não Rotulados
  - Categorização
  - Algoritmos de Agrupamento
  - Redução de dimensionalidade





Qual o agrupamento natural destes dados?

Europeus

Asiáticos

Americanos

É claramente um processo subjetivo…

Qual o agrupamento natural destes dados?

Aprendizado Supervisionado

- No aprend. supervisionado, todas as amostras de treinamento estavam rotuladas, ou seja, com o valor do conceito alvo associado

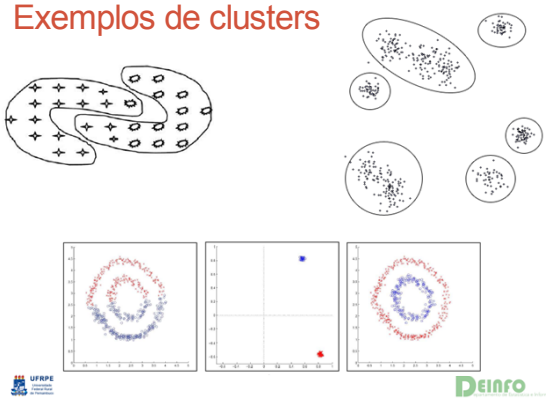
Vetor de atributos	Classe
0,43 0,03 0,40 0,19 0,12 0,16 0,04 0,01 0,00 0,01 0,40 0,02 ,	Bart

- Estes exemplos são ditos “supervisionados”, pois, contém tanto a entrada (atributos), quanto a saída (valor do conceito alvo).

Aprendizado não supervisionado

- Porém, muitas vezes temos que lidar com exemplos "não–supervisionados", isto é, exemplos não rotulados, ou seja:
  - sem um atributo alvo associado ou
  - sem um valor de atributo alvo associado
- Por que?
  - Coletar e rotular um grande conjunto de exemplos pode custar muito (tempo, esforço, dinheiro).
  - Às vezes essa informação não é conhecida

Exemplos de clusters



Aprendizado não supervisionado

- Porém, podemos utilizar grandes quantidades de dados não rotulados para treinamento e somente então "usar supervisão" para rotular os agrupamentos encontrados.
- Isto é apropriado para aplicações de mineração de dados (*datamining*), onde o conteúdo de grandes bases de dados não é conhecido antecipadamente.

Aprendizagem Não Supervisionada

- Classificação não supervisionada
  - Se propõe a encontrar grupos homogêneos a partir de um conjunto de indivíduos
  - **Objetivo**: os indivíduos semelhantes devem pertencer ao mesmo grupo
  - É um objetivo intuitivo mas não é uma definição precisa da noção de grupo

Aprendizado não supervisionado

O interesse principal é desvendar a organização dos padrões em clusters (agrupamentos) consistentes, os quais permitirão descobrir similaridades e diferenças entre padrões bem como derivar conclusões úteis a respeito deles.

- ***Clustering*** = *Aprendizagem Não Supervisionada = Aprendizado Sem Professor = Taxonomia Numérica = Tipologia = Partição.*

Aprendizagem Não Supervisionada

- Agrupar para que?
  - Existem classes "naturais" e o desafio é encontrá-las
  - Deseja-se construir as classes segundo estruturas classificatórias (impostas)
  - Encontrar classes úteis para o usuário
  - Simplificação dos dados
  - Geração de Hipóteses
  - Predição com base nos grupos formados

Aplicações de Clustering

- Reconhecimento de Padrões
- Análise de Dados
- Processamento de Imagens
- Bioinformática
- Economia (especialmente pesquisa de mercado)
- Internet
  - Classificação de documentos
  - Agrupamento de dados provenientes do Weblog para descobrir grupos de acesso similares

Exemplo 2

Input: 10 million images (sampled frames from YouTube)

Output:

Impact: state-of-the-art results on object recognition (22,000 categories)

Exemplos de Aplicações de Clustering

- Marketing: Ajuda a descobrir grupos de clientes e usa esse conhecimento para orientar as campanhas publicitárias
- Geoprocessamento: Identificação de áreas de propriedades similares
- Seguro: Identificação de grupos de segurados com um custo médio elevado de reembolso
- Planejamento Urbano: Identificação de grupos de habitação segundo o tipo, valor e localização geográfica

Exemplo 2 (cont.)

Fonte: http://arxiv.org/abs/1112.6209

Building High-level Features Using Large Scale Unsupervised Learning

Quoc V. Le QUOCLE@CS.STANFORD.EDU
Marc'Aurelio Ranzato RANZATO@GOOGLE.COM
Rajat Monga RAJATMONGA@GOOGLE.COM
Matthieu Devin MDEVIN@GOOGLE.COM
Kai Chen KAICHEN@GOOGLE.COM
Greg S. Corrado GCORRADO@GOOGLE.COM
Jeff Dean JEFF@GOOGLE.COM
Andrew Y. Ng ANG@CS.STANFORD.EDU

Figure 6. Visualization of the cat face neuron (left) and human body neuron (right).

Exemplo 1

Input: raw text (100 million words of news articles)...

Output:

Cluster 1: Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends Sundays Saturdays
Cluster 2: June March July April January December October November September August
Cluster 3: water gas coal liquid acid sand carbon steam shale iron
Cluster 4: great big vast sudden mere sheer gigantic lifelong scant colossal
Cluster 5: man woman boy girl lawyer doctor guy farmer teacher citizen
Cluster 6: American Indian European Japanese German African Catholic Israeli Italian Arab
Cluster 7: pressure temperature permeability density porosity stress velocity viscosity gravity tension
Cluster 8: mother wife father son husband brother daughter sister boss uncle
Cluster 9: machine device controller processor CPU printer spindle subsystem compiler plotter
Cluster 10: John George James Bob Robert Paul William Jim David Mike
Cluster 11: anyone someone anybody somebody
Cluster 12: feet miles pounds degrees inches barrels tons acres meters bytes
Cluster 13: director chief professor commissioner commander treasurer founder superintendent dean custodian
Cluster 14: had hadn't hath would've could've should've must've might've
Cluster 15: head body hands eyes voice arm seat eye hair mouth

Definição formal

- Dado um conjunto de dados *X:*

$$X = \{x_1, x_3, x_3, \ldots x_n\}$$

- definimos como um *m*–agrupamento de X a partição de X em *m* conjuntos (clusters ou grupos) C_1 , C_2 , ..., C_m tal que as três condições seguintes sejam satisfeitas:

$C_i \neq \emptyset$, i= 1,2,..., m (nenhum cluster pode ser vazio)

$\bigcup_{i=1}^{m} C_i = X$ (a união de todos os clusters é igual ao conjunto que os gerou)

$C_i \cap C_j = \emptyset$, $i \neq j$ (a interseção de dois clusters é vazia, i.e, não devem ter padrões em comum) -> HARD CLUSTERING

O que é um bom agrupamento?

- Um bom método de agrupamento fornece grupos de alta qualidade com
  - **Alta similaridade intra-grupo**
  - **baixa similaridade inter-grupo**
- A qualidade do resultado de um agrupamento depende tanto da **medida de similaridade** usada pelo **método** como da sua implementação.
- A qualidade de um método de agrupamento é também medido pela sua habilidade para descobrir os padrões escondidos.

Principais Etapas

- e) Análise e Interpretação dos Resultados
  - Diferentes escolhas de atributos, medidas de proximidade, critérios de agrupamento e algoritmos de clusterização levam a **resultados totalmente diferentes**.
  - Qual resultado é o correto?

Principais Etapas

- a) Aquisição dos dados
  - 1) Seleção das observações (indivíduos, objetos, casos, itens)
  - 2) Seleção das variáveis (caracteres, descritores) e das correspondentes escalas
  - 3) Construção da Tabela de Dados
- b) Pré-processamento dos dados
  - 1) Mudança de escala
  - 2) Normalização
  - 3) Extração de caracteres

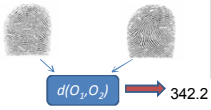
Índices de proximidade

- *Significado:*
  - *Qualidade do que é similar; que é da mesma natureza; semelhante; homogéneo* (dicionário Priberam)
- Índices de Proximidade
  - Similaridade
  - Dissimilaridade

Principais Etapas

- c) Cálculo da Proximidade
  - Medida para quantificar quão **similar** ou **dissimilar** são dois vetores de atributos.
  - 1) Escolha de um Índice de Proximidade
  - 2) Construção da Matriz de Proximidades
- d) Seleção de um Algoritmo de Formação de Grupos em função do tipo de agrupamento desejado
  - Depende da interpretação que o especialista dá ao termo **sensível** com base no tipo de cluster que são esperados.
  - Por exemplo, um cluster compacto de vetores de atributos pode ser sensível de acordo com um critério enquanto outro cluster alongado, pode ser sensível de acordo com outro critério.

Propriedades das medidas de (dis)similaridade



Estas caixas representativas das medidas implement
uma função de duas variáveis.
Estas funções podem ser simples ou complexas.
No entanto há algumas propriedades a considerar.

- $d(i,j) = d(j,i)$ — *Simetria*
- $d(i,i) = 0$ — *Preservação de auto-dissimilaridade*
- $d(i,j) = 0 \leftrightarrow i = j$ — *Positividade (Separação)*
- $d(i,j) \leq d(i,k) + d(j,k)$ — *Desigualdade triangular*

Dissimilaridade entre objetos

- Distâncias são normalmente usadas como medida de dissimilaridade entre objetos
- Entre as mais populares: distância de *Minkowski*

$$d(i,j)=\sqrt[q]{(|x_{i1}-x_{j1}|^q+|x_{i2}-x_{j2}|^q+...+|x_{ip}-x_{jp}|^q)}$$

  - onde $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ e $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ são dois vetores *p*-dimensionais, e *q* é um inteiro positivo
- Se *q* = *1*, *d* é a distância de Manhattan

$$d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+...+|x_{ip}-x_{jp}|$$

Variáveis nominais

- Variável de escala nominal que pode assumir mais de 2 categorias, e.x., vermelho, amarelo, azul, verde
- Método 1: Concordâncias simples
  - *m*: # das concordâncias, *p*: número de variáveis

$$d(i,j)=\frac{p-m}{p}$$

- Método 2: usa um grande número de variáveis binárias
  - Criação de uma nova variável binária para cada uma das *M* categorias

Dissimilaridade entre objetos

- *Se q = 2, d é a distância euclidiana*:

$$d(i,j)=\sqrt{(|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+...+|x_{ip}-x_{jp}|^2)}$$

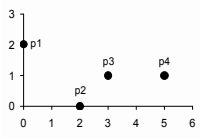
- Outras alternativas: distância ponderada, correlação (similaridade), cosseno, etc.

Outros aspectos relativos aos índices de proximidade

- Escala das Variáveis
- Correlação entre as Variáveis
- Descrições heterogêneas (Variáveis de diferentes tipos)
- Índices de proximidade entre padrões descritos por strings ou árvores
- Índices de proximidade dependentes do contexto
- Índices de proximidade conceptual

Dissimilaridade entre objetos

- Exemplos: Distâncias Manhattan e Euclidiana



L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

Matriz de dissimilaridade para dist. de Manhattan

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Matriz de Dados

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Matriz de dissimilaridade para dist. de Euclidiana

Medida de proximidade

- Medidas de distância
  - Métrica *Minkowski*
  - Métrica *Manhatan*
  - Métrica *Mahalanobis*
  - ...
- Medidas de Correlação
  - Cosseno
  - Correlação de *Pearson*
  - Correlação de *Spearman*
  - ...

Tipos de Clustering

- Algoritmos Flat (ou Particional)
  - Geram partição "plana", i.e. não existe relação hierárquica entre os clusters
- Algoritmos Hierárquicos
  - Geram uma hierarquia de clusters, i.e. cada cluster é associado a um cluster-pai mais genérico
    - Vantagem: diferentes visões dos dados

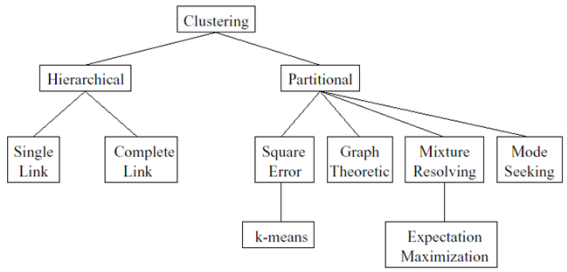
Tipos de Clustering

- Incremental
  - Partição é atualizada a cada novo objeto observado
    - Em geral, apenas um número pequeno de clusters é modificado
- Não-incremental
  - Partição é gerada de uma única vez usando todos os objetos disponíveis

Tipos de Clustering

- Hard
  - Cada objeto pertence exclusivamente a um único grupo na partição
- Fuzzy
  - Cada objeto está associado a um cluster com certo grau de pertinência
    - Partição Fuzzy pode ser convertida facilmente para uma partição hard

Classificações



Outras classificações também podem ser encontradas na literatura!!!
Ex.: Aglomerativos vs Divisivos, Hard vs Fuzzy, etc.

Tipos de Clustering

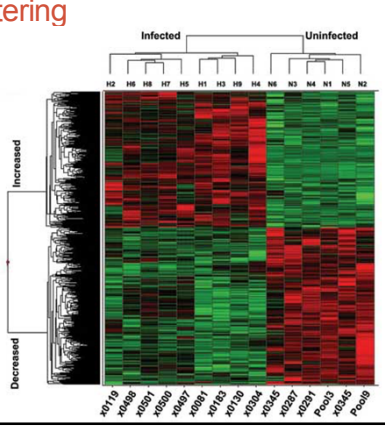
- Completos
  - Cada objeto pertence a pelo menos um cluster
- Parciais
  - Existem objetos que não estão associados a nenhum cluster (*outliers*, ruídos, sem interesse)

TÉCNICAS DE AGRUPAMENTO

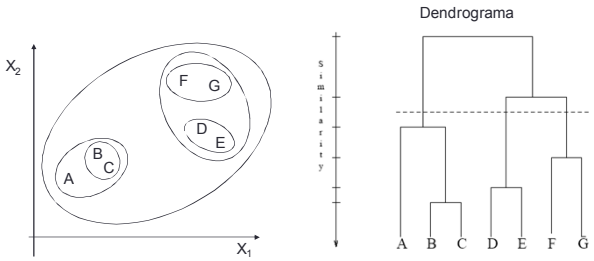
Algoritmos Hierárquicos

- Produzem uma hierarquia de agrupamentos
- Podem ser divididos em 2 subcategorias:
  - **Aglomerativos**:
    - Produzem uma seqüência de agrupamentos com um número decrescente de clusters, m a cada passo.
    - Os agrupamentos produzidos em cada passo resultam do anterior pela fusão de dois clusters em um.
  - **Divisivos**:
    - Atuam na direção oposta, isto é, eles produzem uma seqüência de agrupamentos com um número crescente de clusters, m a cada passo.
    - Os agrupamentos produzidos em cada passo resultam da partição de um único cluster em dois.

Exemplo: Clustering *Microarrays*



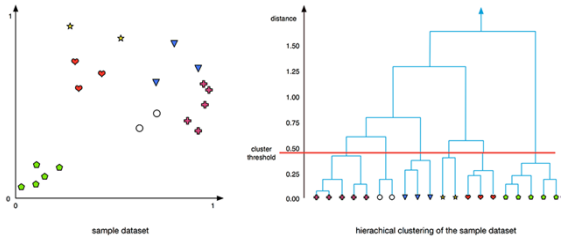
Diferentes visões



Tipos de Algoritmos Hierárquicos

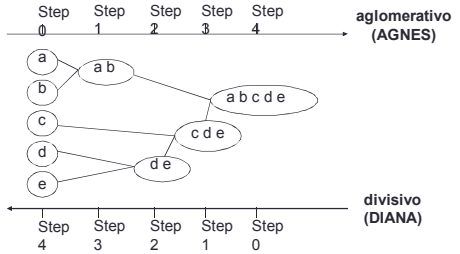
- Algoritmos Hierárquicos Divisivos ou Particionais
  - Assumem estratégia top-down
  - Iniciam com cluster mais geral que é progressivamente dividido em sub-cluster
- Algoritmos Hierárquicos Aglomerativos
  - Assumem estratégia bottom-up
  - Iniciam com clusters específicos que são progressivamente unidos

Diferentes visões



Métodos Hierárquicos

- Usa uma matriz de distâncias como critério de agrupamento. Esses métodos não requerem o número de grupos ***k*** como entrada, mas precisa de uma condição de parada

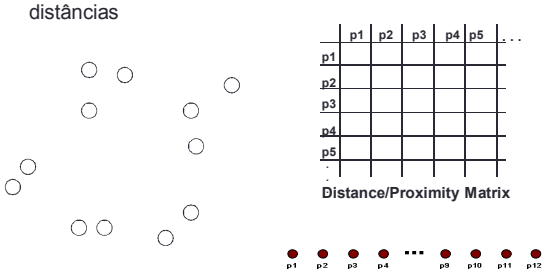


Algoritmos Hierárquicos Divisivos

- **Passo 1:** Inicie alocando todos os documentos em um cluster;
- **Passo 2:** A partir da estrutura existente de grupos, selecione um cluster para particionar;
  - Em geral, o maior cluster, ou o cluster menos homogêneo
- **Passo 3:** Particione o grupo em dois ou mais subgrupos;
- **Passo 4:** Repita os passos 2 e 3 até que um critério de parada seja verificado
  - e.g., até atingir um número desejado de grupos

Configuração Inicial

- Inicia com clusters de pontos individuais e uma matriz de distâncias

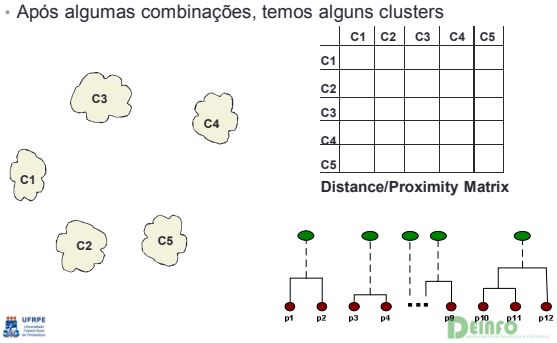


Algoritmos Hierárquicos Divisivos

- **Bi-Secting k-Means**
  - Uso do algoritmo k-Means na etapa de divisão dos clusters
  - Clusters são sucessivamente particionados em 2 sub-clusters
  - Complexidade: O(n log(n))

Estado intermediário

- Após algumas combinações, temos alguns clusters

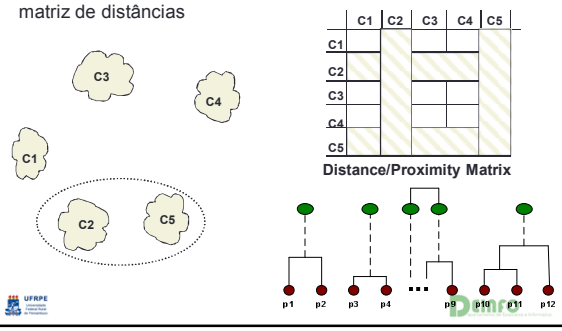


Algoritmos Hierárquicos Aglomerativos

- **Passo 1:** Inicie alocando cada documento como um cluster diferente;
- **Passo 2:** Selecionar o par de clusters mais similares entre si e os agrupe em um cluster mais geral;
- **Passo 3:** Repita o passo 2 até a verificação de um critério de parada
  - e.g., até que todos os documentos sejam agrupados em um único cluster
  - Complexidade: $O(n^2 \log(n))$

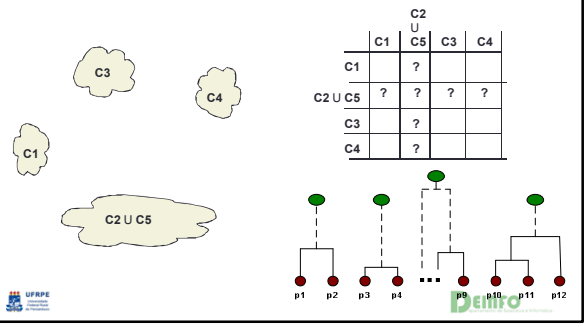
Estado intermediário

- Junte os dois clusters mais próximos (C2 e C5) e atualize a matriz de distâncias

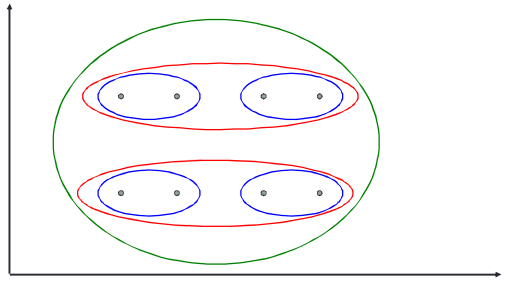


Depois da junção

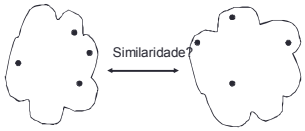
- "Como atualizar a matriz de distâncias?"



Single Link - Exemplo

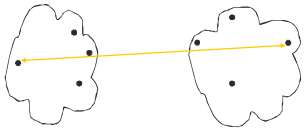


Como definir a similaridade entre dois clusters?



- MIN (Single Link)
- MAX (Complete Link)
- Média entre Grupos (Average-Link)
- Distância entre centróides
- Outros métodos guiados por uma função objetivo:
  - Método Ward's utiliza o erro quadrático

Max (Complete Link)



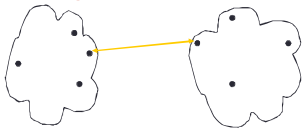
Similaridade entre clusters:

$$sim\_cluster_{CompleteLink}(C_1, C_2) = \max_{p_i \in C_1, p_j \in C_2} sim(p_i, p_j)$$

Efeito:

Produz clusters mais coesos e compactos

Min (Single Link)



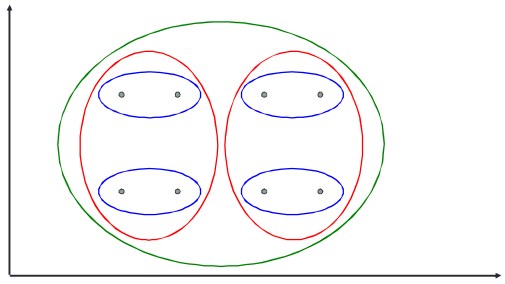
- Similaridade entre clusters:

$$sim\_cluster_{SingleLink}(C_1, C_2) = \min_{p_i \in C_1, p_j \in C_2} sim(p_i, p_j)$$

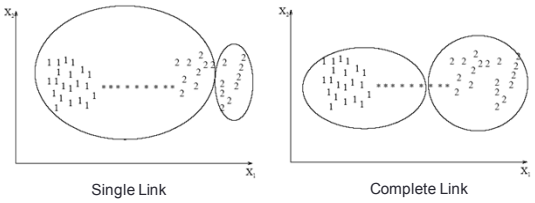
- Efeito:

Produz clusters mais alongados (efeito cadeia)

Complete Link - Exemplo



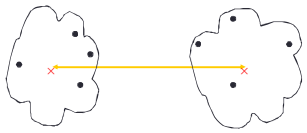
Single Link X Complete Link



Single Link Complete Link

Single-Link conecta pontos de classes diferentes através de uma cadeia de pontos com ruído (*)

Distância entre centróides

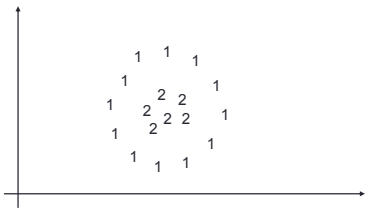


- Um centróide é um vetor que representa o cluster.
- A similaridade é entre os centróides de cada cluster

$$D_{centroids}(C_i, C_j) = sim(G_i, G_j)$$

- G_i é o centróide do cluster i

Single Link X Complete Link



Complete-Link não é capaz de identificar cluster de pontos (1)

Distância entre dois clusters

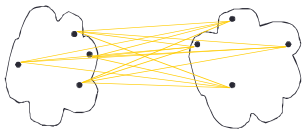
- **Distância de Ward's** é a ***soma total da diferença ao quadrado de cada cluster*** **A e B** menos a **soma ao quadrado da junção dos dois clusters**, resultado no cluster **A U B**

$$\Delta(A,B) = \sum_{i\in A\cup B} \|\vec{x}_i - \vec{m}_{A\cup B}\|^2 - \sum_{i\in A} \|\vec{x}_i - \vec{m}_A\|^2 - \sum_{i\in B} \|\vec{x}_i - \vec{m}_B\|^2 \quad (2)$$

$$= \frac{n_A n_B}{n_A + n_B} \|\vec{m}_A - \vec{m}_B\|^2 \quad (3)$$

where $\vec{m}_j$ is the center of cluster j, and n_j is the number of points in it. Δ is called the **merging cost** of combining the clusters A and B.

Average-Link



Similaridade entre clusters:

$$sim\_cluster_{AverageLink}(C_1, C_2) = \frac{1}{|C_1| * |C_2|} \sum_{p_i \in C_1, p_j \in C_2} sim(p_i, p_j)$$

Efeito:

Equilíbrio entre clusters coesos e flexíveis
Em alguns contextos (e.g., clustering de texto) tem se mostrado mais eficaz

Distância de Ward's

- Semelhante a média do grupo (Average-Link) e da distância entre centróides;
- Menos suscetível a ruídos e outliers
- Inclinado para clusters em forma de círculos
- Hierárquico análogo ao k-means
  - Pode ser usado para inicializar k-means

Algoritmos Hierárquicos

- Resumo:
  - Os algoritmos hierárquicos divisivos são menos custosos que os aglomerativos
  - Dentre os aglomerativos, o Average-Link funciona melhor em algumas aplicações
  - Desempenho pode ser melhorado através da combinação de técnicas

K-means

- Cada cluster *k=1, …, K* é representado por um centroide $\mu \in \mathbb{R}^d$
- Atribuir cada ponto $\phi(x_i)$ ao centróide mais próximo μ_{zi}
- **Função objetivo**

$$\text{Loss}_{\text{kmeans}}(z, \mu) = \sum_{i=1}^{n} \|\phi(x_i) - \mu_{z_i}\|^2$$

ALGORITMOS PARTICIONAIS

Algoritmo k-Means

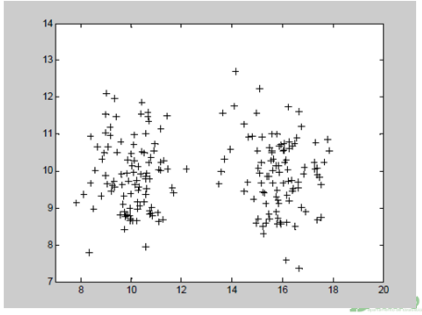
- **Passo 1:** Defina k centróides iniciais, escolhendo k objetos aleatórios;
- **Passo 2:** Aloque cada objeto para o cluster correspondente ao centróide mais similar;
- **Passo 3:** Recalcule os centróides dos clusters.
- **Passo 4:** Repita passo 2 e 3 até atingir um critério de parada
  - e.g. até um número máximo de iterações ou;
  - até não ocorrer alterações nos centróides (i.e. convergência para um mínimo local da função de erro quadrado)

K-means

- O algoritmo *k–Means* ou *k*–Médias é uma técnica iterativa muito simples e poderosa para particionar um conjunto de dados em grupos separados, onde o valor de k, deve ser pré–determinado
- Um dos mais antigos algoritmos de clustering
  - Também um dos mais usados

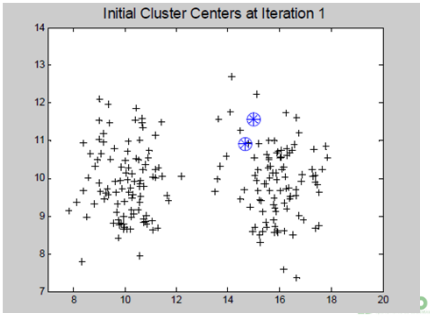
K-means Exemplo

- Conjunto de exemplos *D: (2-dimensional)*



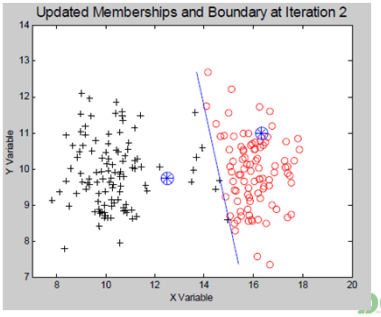
K-means
Exemplo

- Centros iniciais dos clusters (*k*=2)



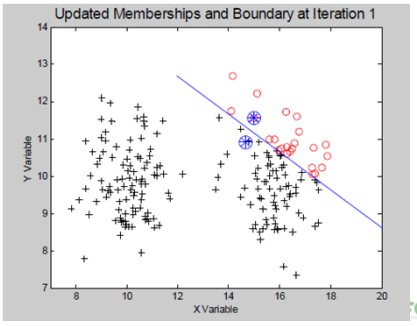
K-means
Exemplo

- Atualizando afiliações (iteração 2)



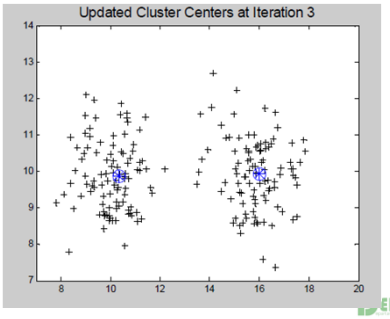
K-means
Exemplo

- Atualizando afiliações (iteração 1)



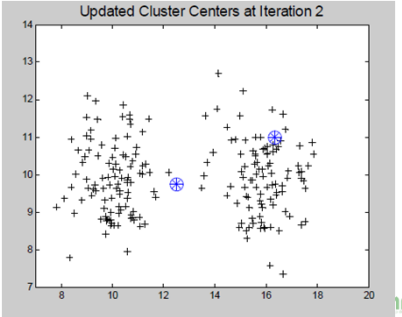
K-means
Exemplo

- Atualizando centros (iteração 3)



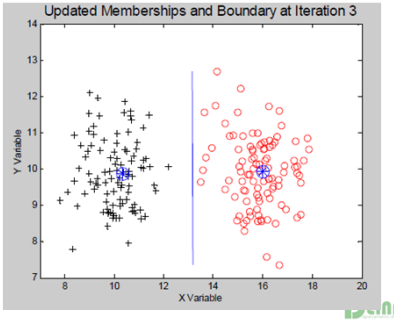
K-means
Exemplo

- Atualizando centros (iteração 2)



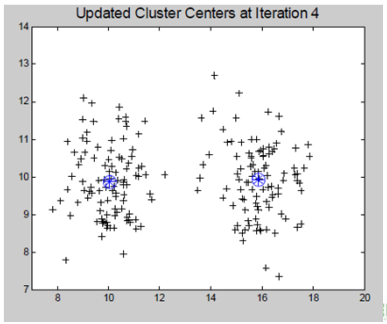
K-means
Exemplo

- Atualizando Alifiação (iteração 3)



K-means
Exemplo

- Atualizando centros (iteração 4)

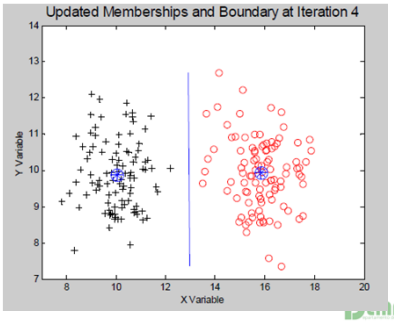


Algoritmo k-Means

- Pontos fortes
  - *Relativamente eficiente*: O(*tkn*), onde *n* é # objetos, *k* é # grupos, e *t* é # iterações. Normalmente, *k*, *t* << *n*.
  - Frequentemente termina em um *ótimo local*.
  - O *ótimo global* pode ser encontrado usando técnicas como: *deterministic annealing* e *algoritmos genéticos*
- Pontos fracos
  - Aplicável apenas quando a *média* é definida, o que fazer com dados categóricos?
  - É necessário especificar a priori *k, o número de grupos*
  - É sensível a ruídos e outliers
  - Não é apropriado para a descoberta de grupos não esféricos

K-means
Exemplo

- Atualizando Afiliação (iteração 4)



K-means
Agrupando pixels em uma imagem

- Podemos usar o algoritmo *k–Means* para agrupar a intensidade dos pixels de uma imagem em k clusters.
- É uma maneira simples de segmentar uma imagem em *k* regiões.
- É um método mais automático do que um limiar escolhido manualmente.

K-means

- Computacionalmente simples.
- O Erro Quadrático Total (TSE) decresce a cada iteração.
- Ele encontra um TSE mínimo global?
  - Não necessariamente
  - Os resultados são sensíveis ao ponto inicial (inicialização dos centróides)
  - Na prática, podemos executá–lo a partir de múltiplos pontos de partida e pegar a solução com menor erro (TSE).
- Clusters definidos com base nos *centróides* (*centro de gravidade*, ou o ponto médio dos cluster:

$$c''' = \frac{1}{|C|}\sum_{d_i \in C} d_i'''$$

- Alocação dos objetos nos clusters feita com base na similaridade com o centróide até critério de parada

K-means
Agrupando pixels em uma imagem

- Como fazer?
  - Tamanho (matriz de pixels) = *mx n*
  - Converter para um vetor com (*m x n)* linhas e 1 coluna
  - Executar o algoritmo *k–Means* com entrada = vetor de intensidades.
  - Atribuir para cada pixel a "cor ou nível de cinza"do *cluster a que ele for atribuído.*

K-means
Agrupando pixels em uma imagem

- Imagem Original

K-means
Agrupando pixels em uma imagem

- K-means (k=5)

K-means
Agrupando pixels em uma imagem

- K-means (k=2)

K-Means
Agrupando Imagens

- Exemplo: Podemos também agrupar conjuntos de imagens
- Cada vetor = uma imagem inteira (dimensão *m x n)*
- *N* imagens de tamanho m x n
  - Execute k–Means
  - *k–Means* está agora agrupando em um espaço de dimensão m x n
  - *k–Means* agrupará as imagens em k grupos

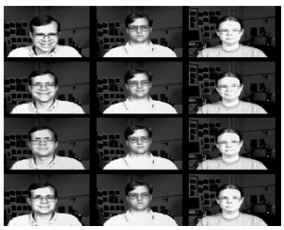
K-means
Agrupando pixels em uma imagem

- K-means (k=3)

K-Means
Agrupando Imagens

- 5 primeiros indivíduos, *k= 2*

Cluster 1



Cluster 2

K-Means
Agrupando Imagens

- Mais 5 indivíduos, k = 2

Cluster 1

Cluster 2

K-Means
Agrupando Imagens

- Todos indivíduos faces alegres, *k= 5*