Aprendizado de Máquina

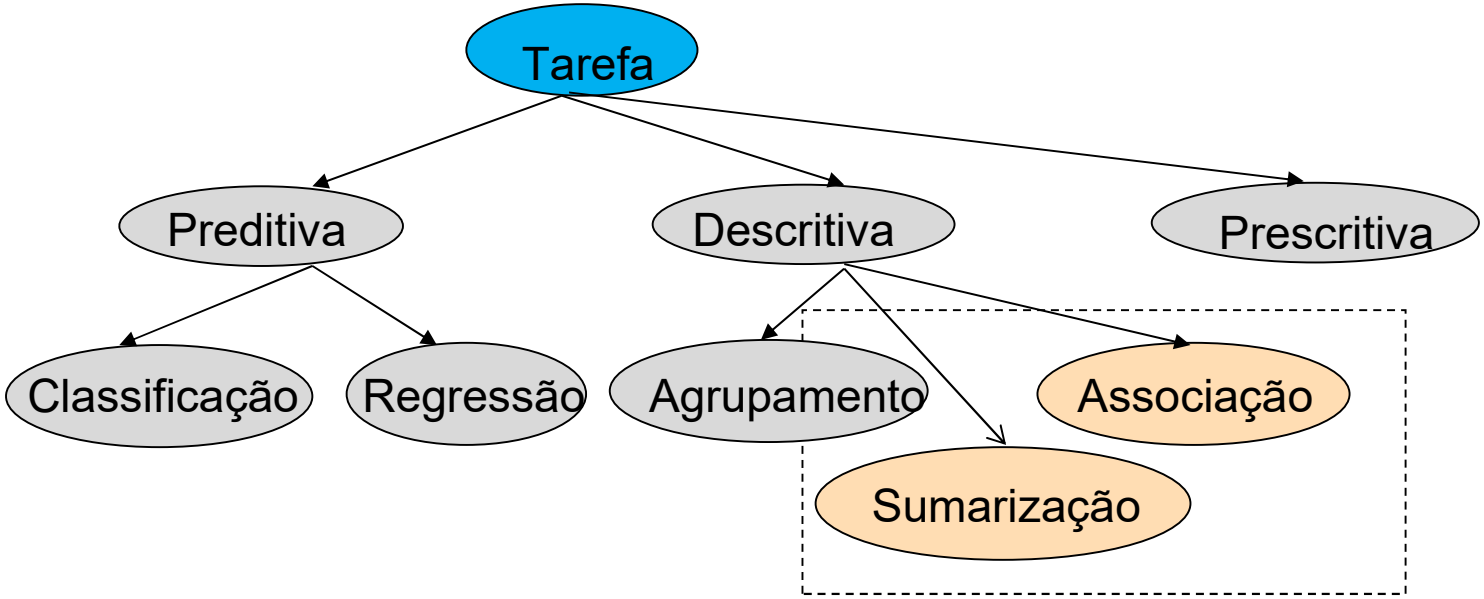
Aula 8: Agrupamento de dados

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br

Tópicos

- Agrupamento de dados
- Dificuldades em agrupamento
- Algoritmos de agrupamento
- Validação
- Aplicações

Tarefas de aprendizado



Introdução

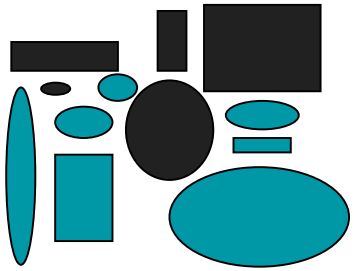
- Nem sempre os dados em um conjunto estão rotulados
  - Custo
  - Impossibilidade
- Conhecimento útil e relevante pode ser extraídos de dados não rotulados
  - Grupos de dados similares

Agrupamento

- Organização de um conjunto de objetos em grupos (clusters)
  - Não existe uma definição precisa
  - Particiona objetos de acordo com alguma relação entre eles
  - Busca partição que maximiza:
    - Similaridade entre objetos de um mesmo grupo e
    - Dissimilaridade entre objetos de grupos diferentes

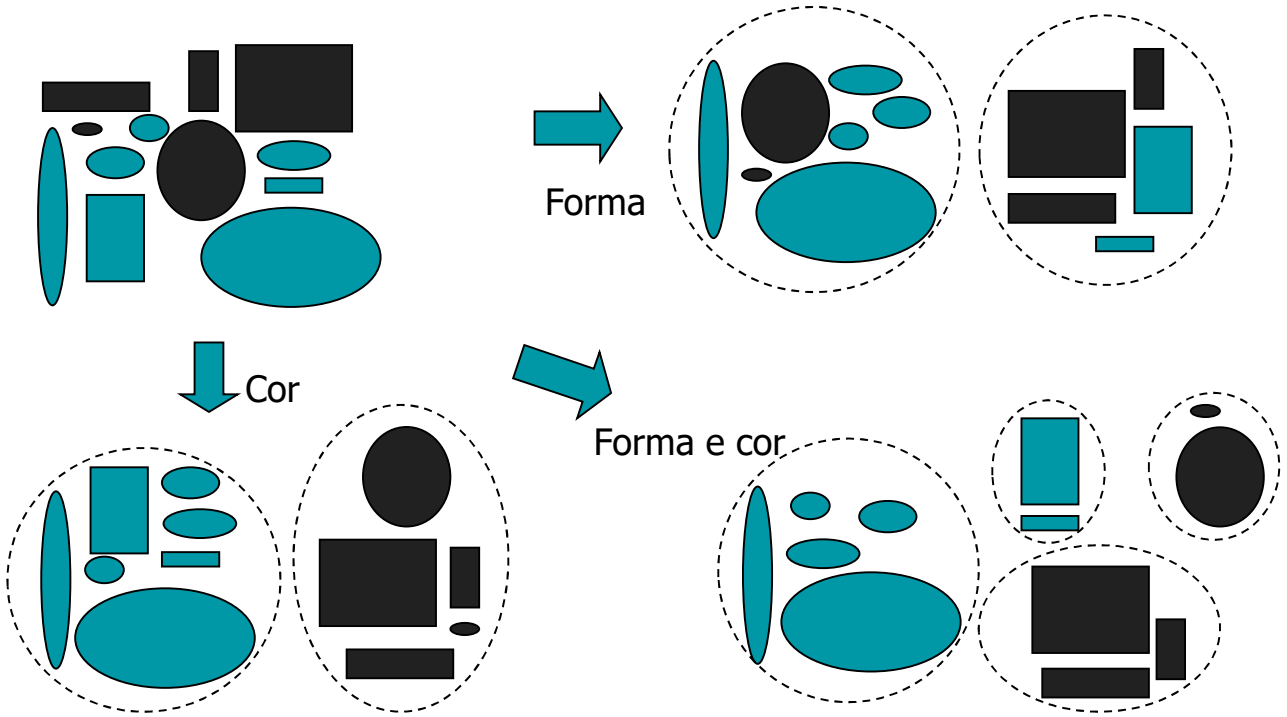
Agrupamento

- Supor os objetos:

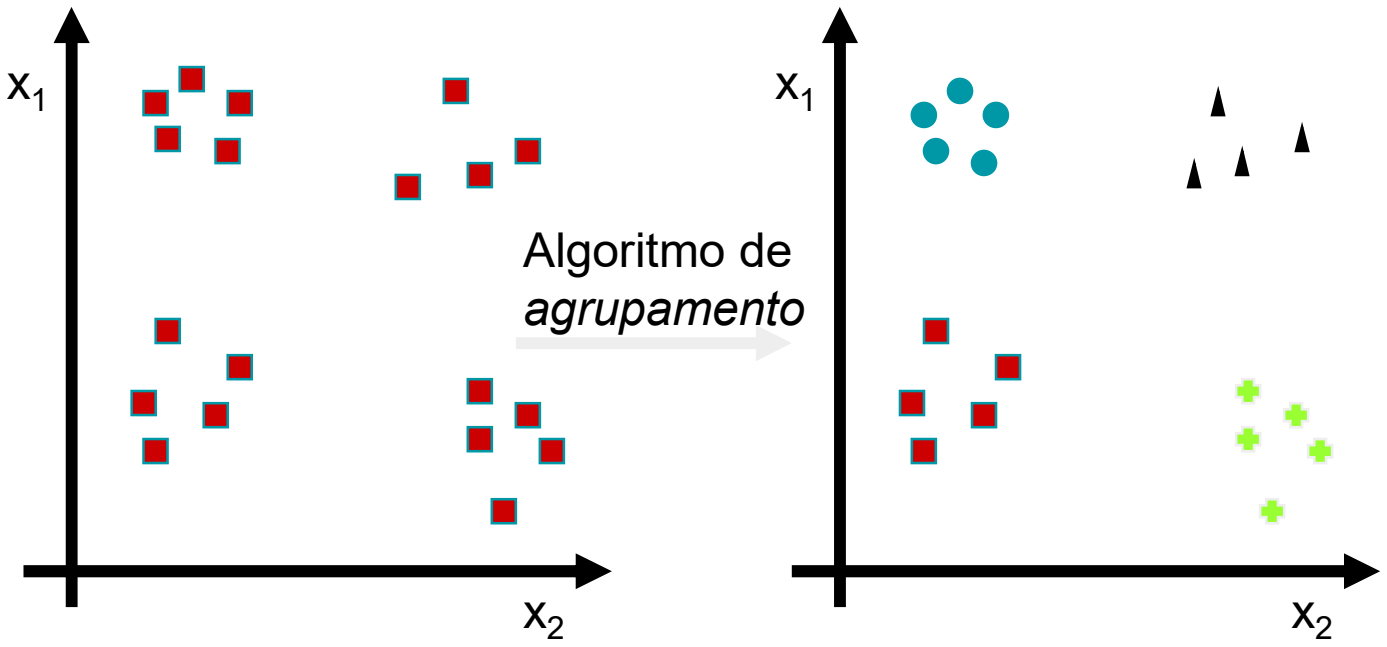


Como particionar?

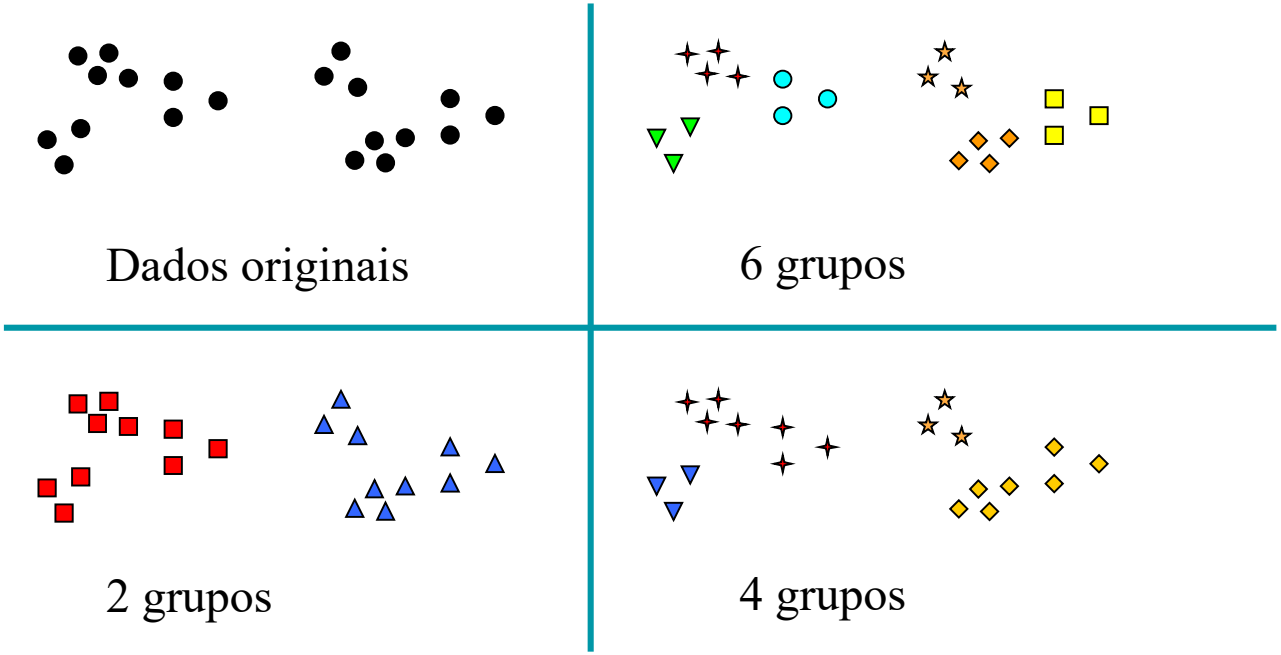
Agrupamento



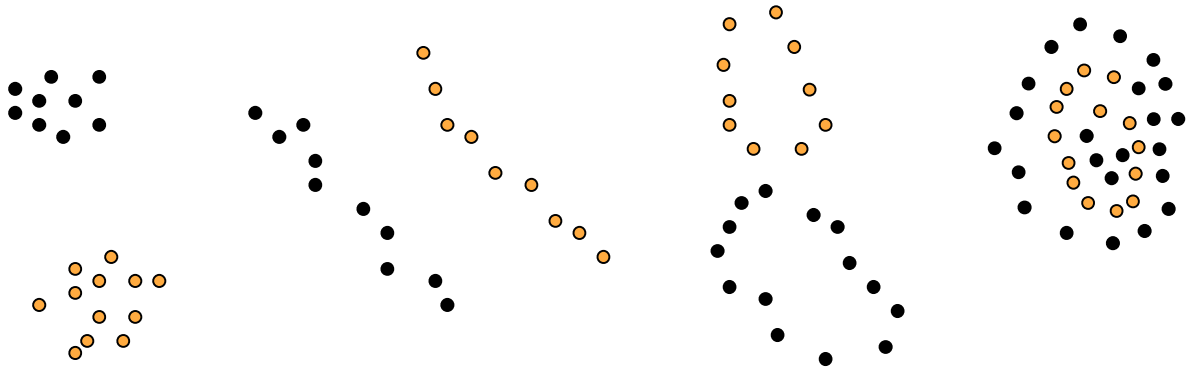
Agrupamento de dados



Quantos grupos?



Possíveis formatos



Agrupamento de dados

- Definição do que é um agrupamento
  - Imprecisa
  - Depende de:
    - Natureza dos dados
    - Resultados desejados
  - Existem várias

Tipos de agrupamento

- Seja $X = \{x_1, x_2, ..., x_n\}$ o conjunto de todos os objetos de um conjunto de dados
  - Tarefa: colocar cada X_i em um dos k clusters $C_1, C_2, ..., C_k$
- De acordo com a pertinência dos objetos, agrupamentos podem ser de dois tipos:
  - Tipo 1: duro (crisp)
  - Tipo 2: fuzzy

Tipos de agrupamento

- Agrupamento crisp
  - Cada objeto X_i pertence ou não a cada cluster C_j

$$C_i \neq \emptyset,\ i = 1, \ldots, k$$

$$C_i \cap C_j = \emptyset,\ i \neq j,\ i, j \in \{1,2, \ldots, k\}$$

  - Objeto em C_i é mais semelhante a outros objetos em C_i do que àqueles em C_j, $i \neq j$

Tipos de agrupamento

- Agrupamento fuzzy
  - Usa uma função de pertinência para definir o quanto um elemento pertence a um grupo

$$Pert_j : x_i \rightarrow [0, 1]$$

$Pert_j$ = pertinência ao grupo j
k = número de grupos
n = número de objetos

$$\sum_{j=1}^{k} Pert_j(x_i) = 1, \; i \in \{1, ..., n\}$$

$$0 < \sum_{i=1}^{n} Pert_j(x_i) \leq n, \; j \in \{1, ..., k\}$$

Objetivo

- Encontrar a partição que maximiza a similaridade em um grupo
  - Maximiza a dissimilaridade entre grupos
  - Quanto maior a homogeneidade dentro dos grupos e a diferença entre os grupos, melhor
- Alternativas
  - Busca exaustiva
  - Algoritmos de agrupamento de dados

Busca exaustiva

- Tentar todos os possíveis agrupamentos de *k* grupos (para vários valores de *k*)
- Números de Stirling do segundo tipo
    - Número de formas de particionar *n* dados em *k* subconjuntos não vazios

$$>> \binom{n}{k} \geq \left(\frac{n}{k}\right)^k$$

k = número de grupos
n = número de objetos

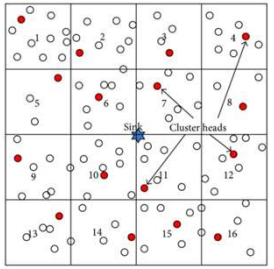
- Impraticável

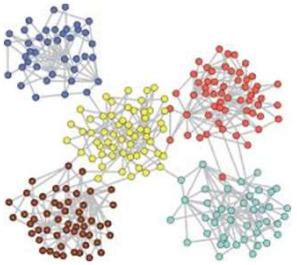
Agrupamento de dados

- Diferentes partições podem ser encontradas
  - Por diferentes algoritmos
    - Utilizam critérios diferentes para buscar uma boa partição
  - Pelo mesmo algoritmo
    - Diferentes inicializações
    - Diferentes números de clusters (grupos)

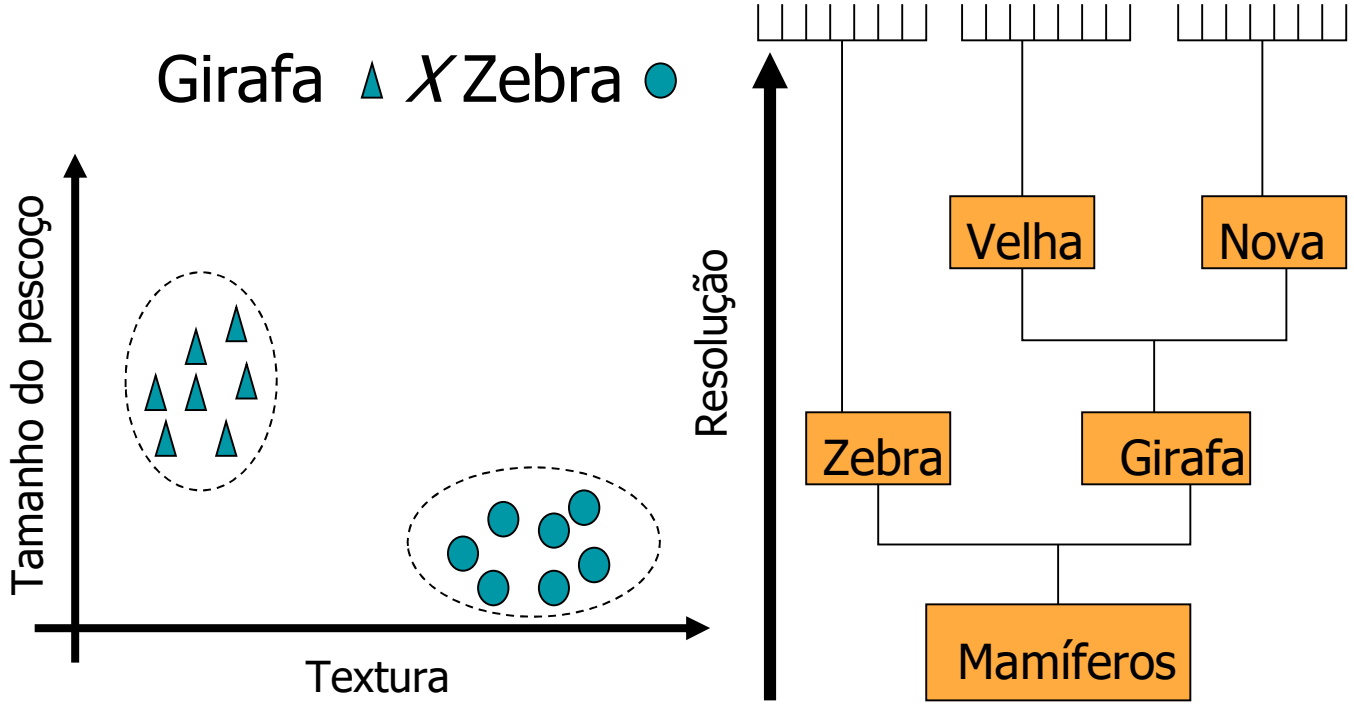
Algoritmos de agrupamento

- Principais abordagens
  - Particionais
    - Protótipos (erro quadrático médio)
    - Densidade
  - Hierárquicos
  - Baseados em grids (grades)
  - Baseados em grafos

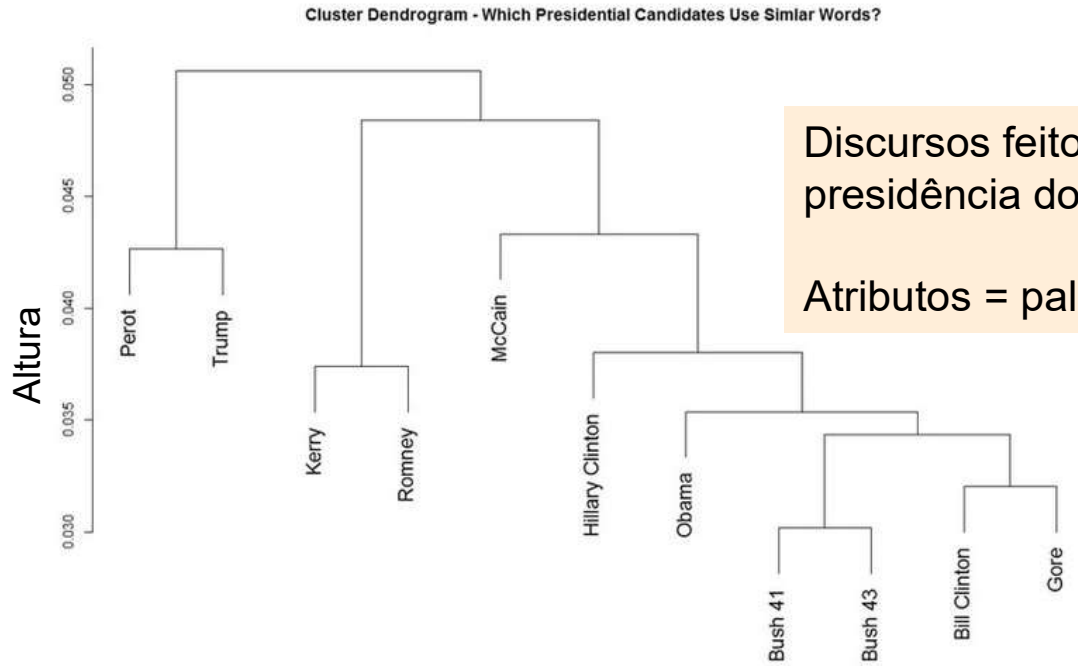




Particional X Hierárquico



Agrupamento hierárquico



Discursos feitos por candidatos a presidência dos EUA

Atributos = palavras usadas

http://www.realclearpoljtics.com/articles/2016/10/19/debate_text_mining_trump_has_a_cousin_in_perot_and_a_friend_in_obama.html

Algoritmos particionais

- Principais características
  - Produzem um único agrupamento (partição)
  - A maioria utiliza abordagem “gulosa” (*greedy*)
    - Busca pela melhor alternativa no momento, sem considerar futuras consequências
    - Uma vez tomada uma decisão, não volta atrás
    - Geralmente resultado depende da ordem de apresentação dos exemplos

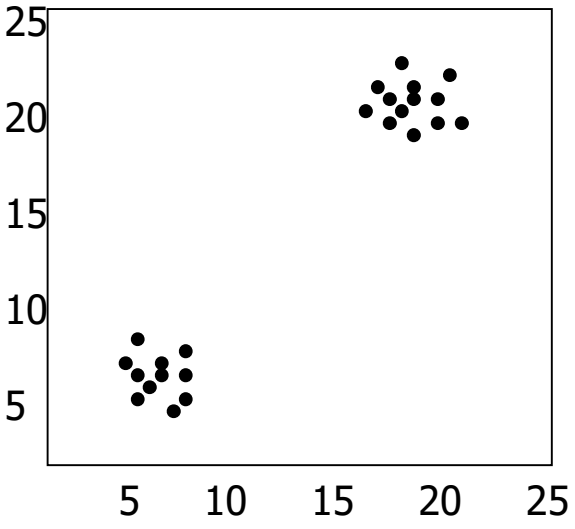
Algoritmo Particional Básico (APB)

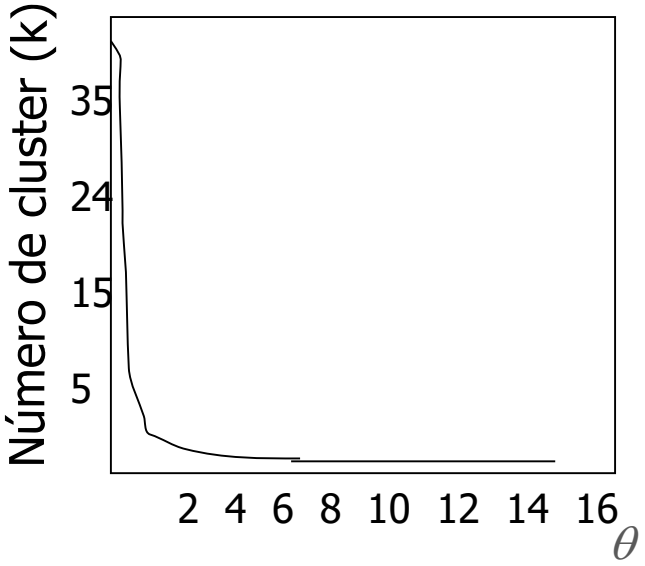
Entrada: θ, q
/ θ, distância máxima para um objeto entrar em um cluster*
/ q, número máximo de clusters, é opcional) */*
1 Inicializar k = 1, $C_k = \{x_1\}$
2 Para i = 2 até n faça
Encontrar o cluster C_j mais próximo de x_i
Se $d(C_j, x_i) > \theta$ e k < q / usar centros*
Então k = k + 1
$C_k = \{x_i\}$
Senão $C_j = C_j \cup \{x_i\}$ / atualizar centros*

Algoritmo Particional Básico (APB)

- Sensitividade (granularidade)
  - Se θ for grande, poucos (grandes) clusters são formados
    - E vice-versa
    - Como estimar valor de θ?
    - Executar APB para vários valores de θ e k
      - Plotar gráfico θ versus k

Algoritmo Particional Básico (APB)





Algoritmos particionais

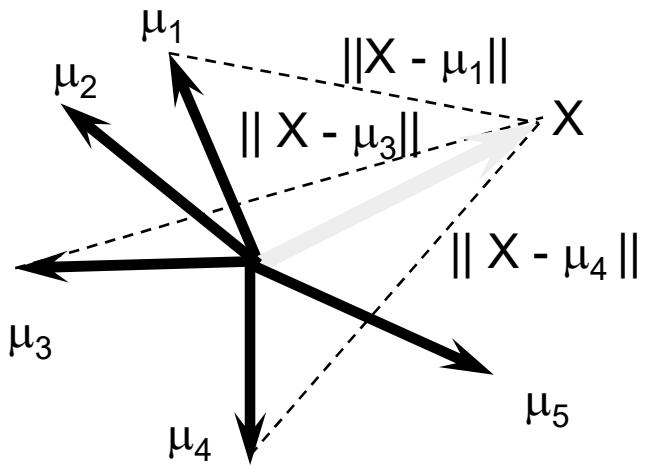
- K-médias (K-médias ótimo, K-médias sequencial)
- SOM
- FCM
- DENCLUE
- CLICK
- CAST
- SNN

Algoritmo k-médias

- Supor n objetos $x_1, x_2, ..., x_n$ a serem agrupados em k clusters, $k < n$
  - Seja μ_i a média dos objetos do cluster C_i
  - Seja d uma medida de distância
    - $\boldsymbol{x_p} \in$ cluster C_i se $d(\boldsymbol{x_p}, \mu_i)$ for menor que todas as k–1 distâncias entre $\boldsymbol{x_p}$ e μ_j, j = 1, 2, ..., k e i ≠ j

Medidas de distância

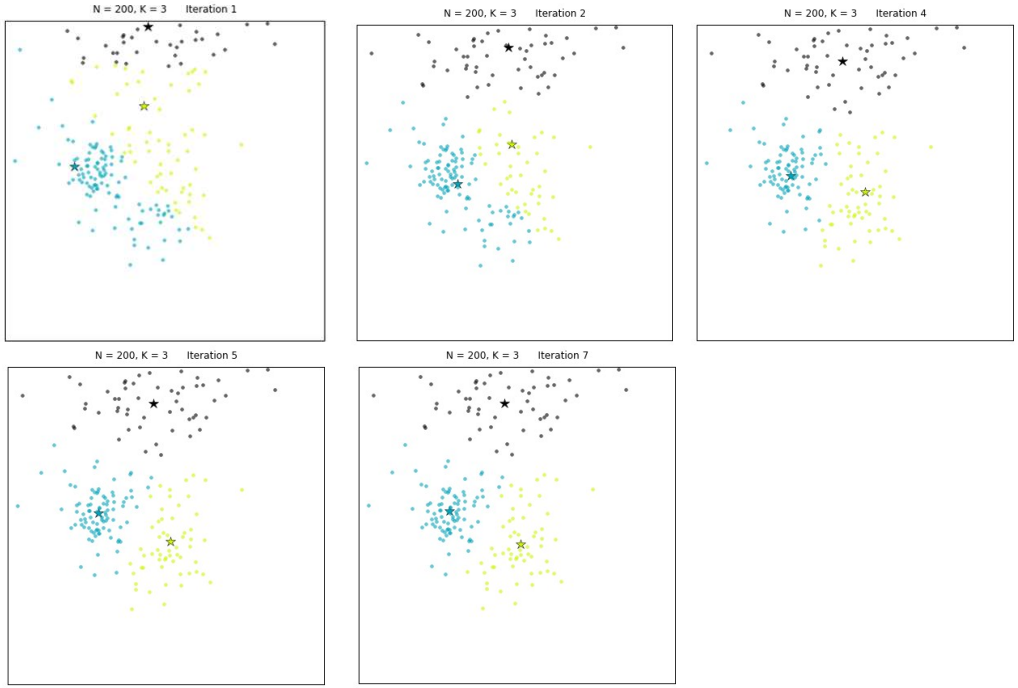
- Calcula || X - μ_i || para i = 1 até K
  - Escolher o grupo com menor distância



Algoritmo k-médias

1 Sugerir médias μ_1, μ_2, ..., μ_k iniciais
2 Repetir
/ Usar as médias sugeridas para agrupar*
/ os n objetos nos K clusters*
Para cada objeto x_j com j variando de 1 a n
Inserir x_j no cluster C_i mais próximo
Substituir μ_i pela média de todos os exemplos do cluster C_i
Até nenhuma das médias mudar

Algoritmo k-médias



Algoritmo k-médias

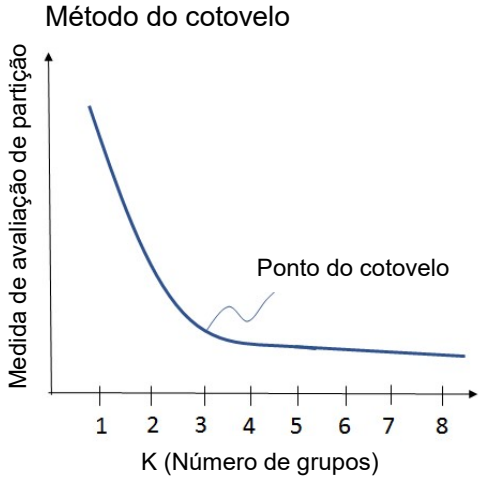
- Médias iniciais
  - Elementos podem ser aleatoriamente escolhidos
  - Objetos claramente diferentes

Limitações do k-médias

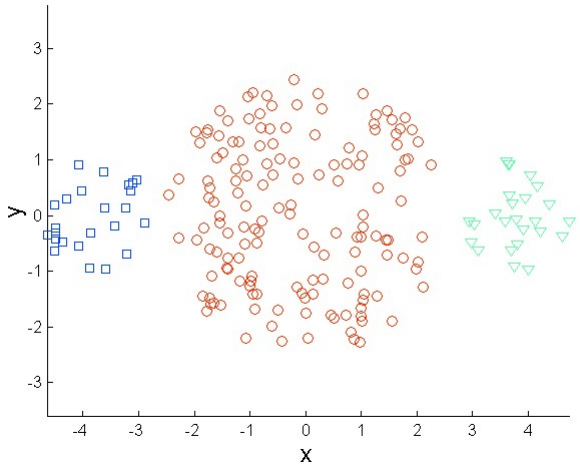
- Escolha do valor de K
  - Tentativa e erro ou automática
- Algoritmos K-médias tem problemas quando:
  - Grupos têm diferentes densidades
  - Grupos têm formatos não hiper-esféricos
  - Atributos estão em diferentes escalas
- Tem problemas também quando os dados contêm *outliers*

Quantos grupos

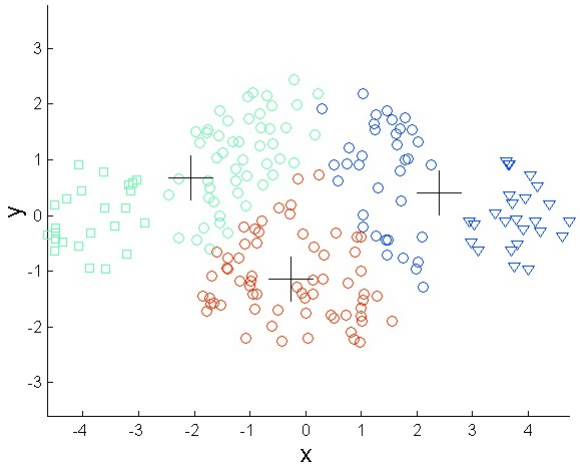
- Qual o melhor valor de k?
  - Vários métodos
- Método cotovelo (Elbow)
  - Traçar uma linha em um gráfico ligando o desempenho obtido para diferentes valores de k
    - Se o gráfico lembra um braço, o valor no cotovelo indica um bom valor de k



Grupos encontrados

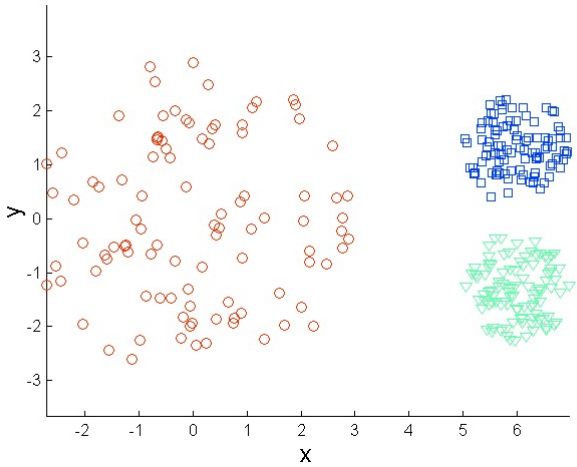


Grupos verdadeiros

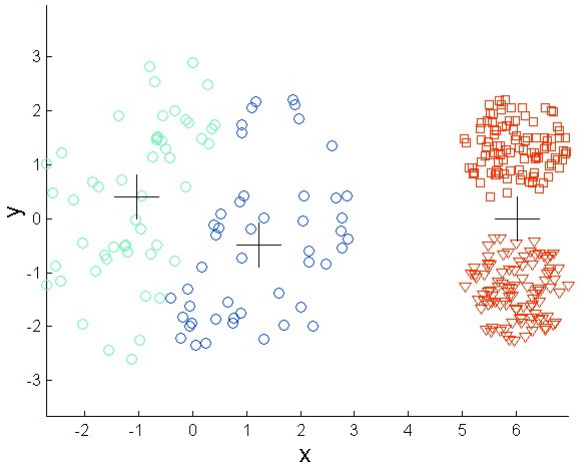


K-médias (3 grupos)

Densidades diferentes

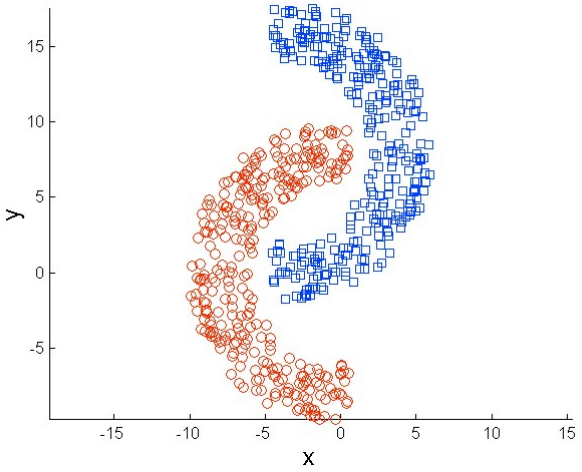


Grupos verdadeiros

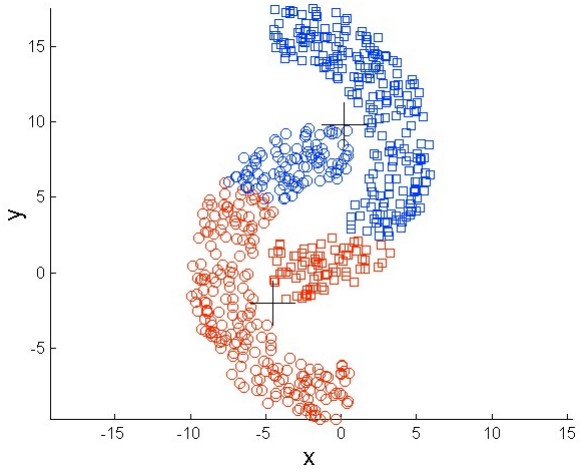


K-médias (3 grupos)

Formatos não hiperesféricos



Grupos verdadeiros



K-médias (2 grupos)

Algoritmos hierárquicos

- Utilizam diagrama de árvore (dendograma)
  - Produz uma sequência (hierarquia) de agrupamentos
- Historicamente usados em áreas que empregam estrutura hierárquica
  - Ex.: Biologia e arqueologia
  - Conceito de representação hierárquica de dados foi desenvolvido inicialmente na Biologia

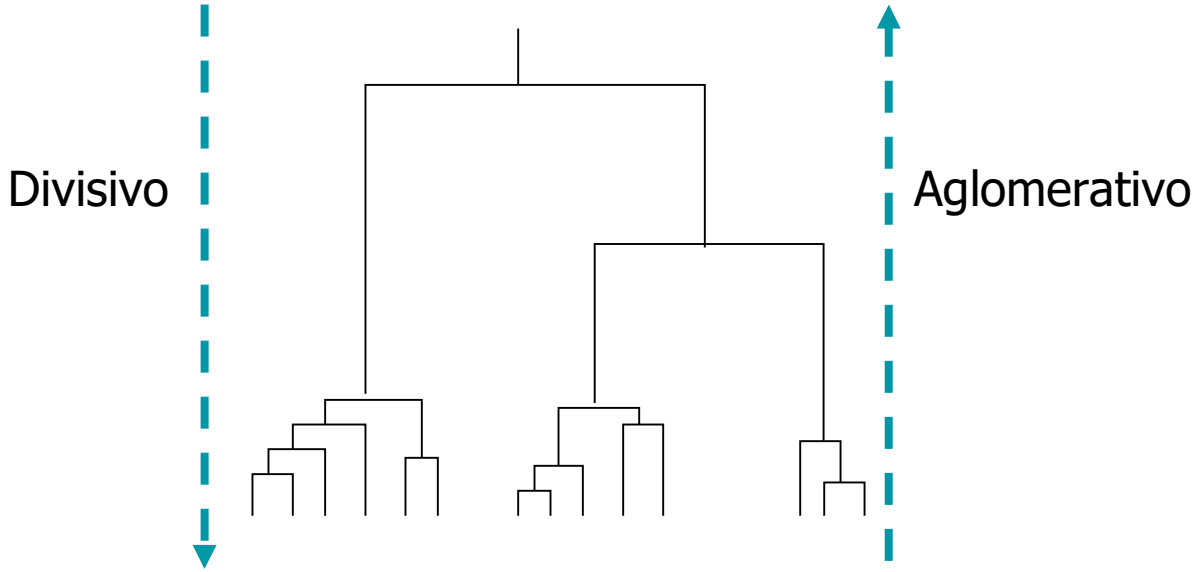
Algoritmos hierárquicos

- Algoritmos de agrupamento hierárquicos
  - Lembram a estrutura hierárquica da taxonomia de Lineu para classificação de organismos
  - Domínio, reino, filo, classe, ordem, ...
  - Geralmente preferido por Biólogos
    - Aplicações na biologia geralmente não se preocupam com o número ótimo de clusters
    - O interesse está na estrutura da árvore completa (dendograma)

Algoritmos hierárquicos

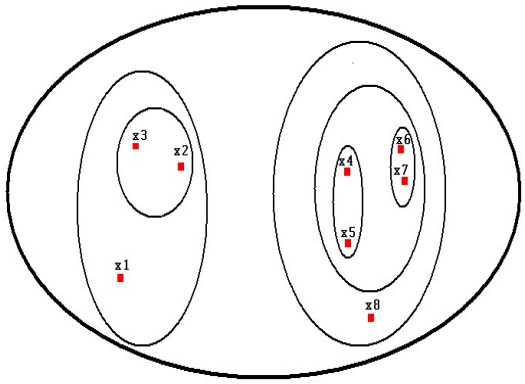
- Tipos:
  - Aglomerativos: combinam, repetidamente, dois clusters em um
    - A cada passo, combina os dois clusters mais próximos
  - Divisivos: Dividem, repetidamente, um cluster em dois
    - A cada passo, divide o cluster menos homogêneo em dois novos clusters

Exemplo



Exemplo

- Não precisa ser apenas por meio de um dendogrtama
  - Diagrama de Venn



Algoritmos hierárquicos

- Definições:
  - Seja $P_t = \{C_1, C_2, ..., C_{m_t}\}$ uma partição no nível t de $X = \{x_1, x_2, ..., x_n\}$
    - P_t é um agrupamento crisp
  - Diz-se que P_t é encaixado em P_t' ($P_t \subset P_t'$) se:
    - Cada cluster em P_t é um subconjunto de um cluster em P_t' e
    - Pelo menos um cluster em P_t é um subconjunto próprio de algum cluster em P_t' ($A \subset B$ e $A \neq B$)
      - A é **subconjunto próprio** de B se e somente se cada elemento de A está em B, mas pelo menos um elemento de B não está em A

Exemplo

- Sejam:
  - $P_A = \{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5\}\}$
  - $P_B = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\}$
  - $P_C = \{\{x_1, x_4\}, \{x_3\}, \{x_2, x_5\}\}$
  - Pode-se dizer que:
    - $P_A \quad P_B$
    - $P_A \quad P_C$
    - $P_A \quad P_A$

Exemplo

- Sejam:
  - $P_A = \{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5\}\}$
  - $P_B = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\}$
  - $P_C = \{\{x_1, x_4\}, \{x_3\}, \{x_2, x_5\}\}$
  - Pode-se dizer que:
    - $P_A \subset P_B$
    - $P_A \not\subset P_C$
    - $P_A \subset P_A$ (Mas não é um subconjunto próprio)

Algoritmos hierárquicos

- Algoritmos aglomerativos
  - Começam com $P_0 = \{\{x_1\}, ..., \{x_n\}\}$
  - A cada passo t, combinam dois clusters em um, produzindo:
    - $|P_{t+1}| = |P_t| - 1$ e $P_t \subset P_{t+1}$
  - No passo final (passo n-1) tem-se a hierarquia:
    - $P_0 = \{\{x_1\}, ..., \{x_n\}\} \subset P_1 ... \subset P_{n-1} = \{x_1, ..., x_n\}$
    - *Obs.: O símbolo $\subset$ não se refere a um conjunto estar contido em outro, mas a um agrupamento (partição) estar encaixado em outro na hierarquia de agrupamentos*

Algoritmos hierárquicos

- Algoritmos divisivos
  - Começam com $P_0 = \{x_1, ..., x_n\}$
  - A cada passo t, dividem um cluster em dois, produzindo:
    - $|P_{t+1}| = |P_t| + 1$ e $P_{t+1} \subset P_t$
  - No passo final (passo n-1) tem-se a hierarquia:
    - $P_{n-1} = \{\{x_1\}, ..., \{x_n\}\} \subset ... \subset P_0 = \{x_1, ..., x_n\}$
    - *Obs.: O símbolo $\subset$ não se refere a um conjunto estar contido em outro, mas a um agrupamento (partição) estar encaixado em outro na hierarquia de agrupamentos*

Esquema Aglomerativo Generalizado (EAG)

1 Incializar $P_0 = \{\{x_1\}, \ldots, \{x_n\}\}$, $t = 0$
2 Para $t = 1$ até $n - 1$ faça
Encontrar o par de clusters mais próximos (C_i, C_j)
$P_t = (P_{t-1} - \{C_i, C_j\}) \cup \{\{C_i, \cup C_j\}\}$
/ atualiza centros*
/ Número de chamadas a $d(C_i, C_j)$ é $O(n^3)$*

Esquema Aglomerativo Generalizado (EAG)

- Dois métodos de implementação comuns são baseados em:
  - Matrizes
  - Teoria dos grafos
- Uma matriz de objetos *n x m*, D(x), contém os *n* objetos com *m* atributos cada
- Uma matriz de proximidade *(n-t) x (n–t)*, $Prox_t$, fornece a proximidade entre todos os pares de clusters em um nível *t*
  - Utiliza medida de distância (ex. Euclidiana)

Exemplo

- Sejam os dados
  - $x_1 = [1, 1]^t$,
  - $x_2 = [2, 1]^t$,
  - $x_3 = [5, 4]^t$,
  - $x_4 = [6, 5]^t$,
  - $x_5 = [6.5, 6]^t$

$$prox_0^{SM} = \begin{bmatrix} 1 & 0.75 & 0.26 & 0.21 & 0.18 \\ 0.75 & 1 & 0.44 & 0.35 & 0.20 \\ 0.26 & 0.44 & 1 & 0.96 & 0.90 \\ 0.21 & 0.35 & 0.96 & 1 & 0.98 \\ 0.18 & 0.20 & 0.90 & 0.98 & 1 \end{bmatrix}$$

SM: medida de similaridade
Medida de Tanimoto

Exemplo

- Sejam os dados
  - $x_1 = [1, 1]^t$,
  - $x_2 = [2, 1]^t$,
  - $x_3 = [5, 4]^t$,
  - $x_4 = [6, 5]^t$,
  - $x_5 = [6.5, 6]^t$

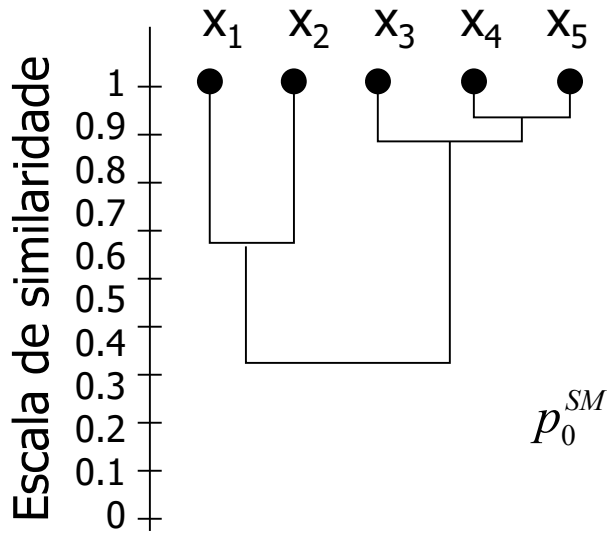
$$prox_0^{DM} = \begin{bmatrix} 0 & 1 & 5 & 6,4 & 7,4 \\ 1 & 0 & 4,2 & 5,7 & 6,7 \\ 5 & 4,2 & 0 & 1,4 & 2,5 \\ 6,4 & 5,7 & 1,4 & 0 & 1,1 \\ 7,4 & 6,7 & 2,5 & 1,1 & 0 \end{bmatrix}$$

DM: medida de dissimilaridade
Distância Euclidiana

Algoritmos hierárquicos

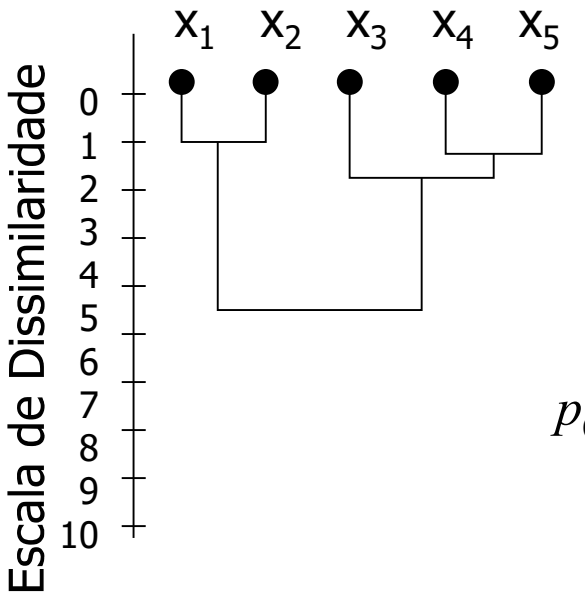
- Dendograma de proximidade
  - Árvore que indica hierarquia de partições
    - Incluindo a proximidade entre dois clusters e quando eles são combinados
    - O corte de um dendograma em qualquer nível produz uma simples partição

Exemplo



$$p_0^{SM} = \begin{bmatrix} 1 & 0.75 & 0.26 & 0.21 & 0.18 \\ 0.75 & 1 & 0.44 & 0.35 & 0.20 \\ 0.26 & 0.44 & 1 & 0.96 & 0.90 \\ 0.21 & 0.35 & 0.96 & 1 & 0.98 \\ 0.18 & 0.20 & 0.90 & 0.98 & 1 \end{bmatrix}$$

Exemplo

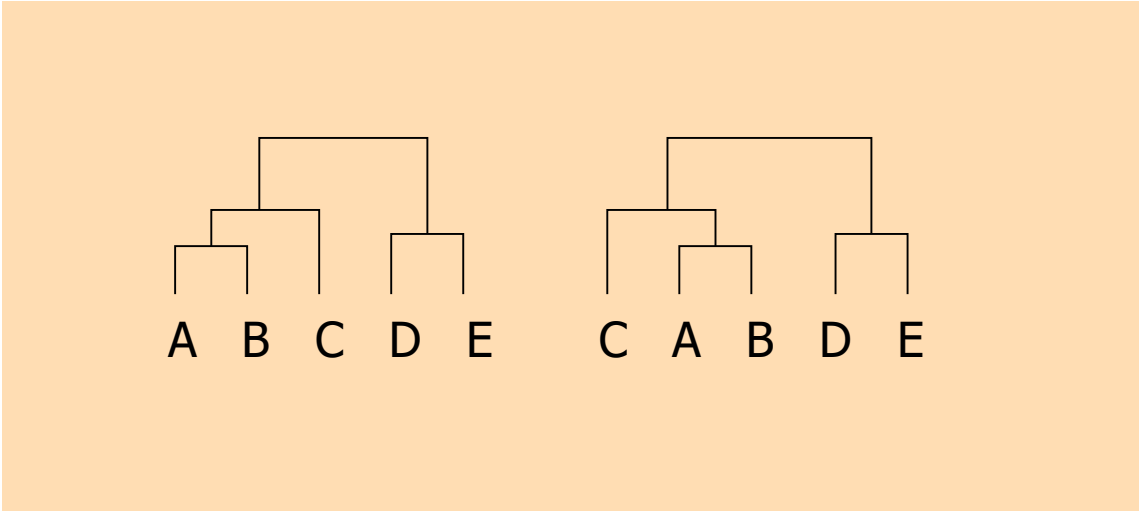


$$p_0^{DM} = \begin{bmatrix} 0 & 1 & 5 & 6,4 & 7,4 \\ 1 & 0 & 4,2 & 5,7 & 6,7 \\ 5 & 4,2 & 0 & 1,4 & 2,5 \\ 6,4 & 5,7 & 1,4 & 0 & 1,1 \\ 7,4 & 6,7 & 2,5 & 1,1 & 0 \end{bmatrix}$$

Algoritmos hierárquicos

- Deve ser observado que o desenho do dendograma é arbitrário
  - Clusters podem ser rotacionados no ponto de bifurcação
    - Afeta a proximidade aparente entre fronteiras de clusters adjacentes
      - Mas a informação importante está contida no conteúdo do cluster e na sua similaridade

Algoritmos hierárquicos



Algoritmos hierárquicos

- E para calcular a distância?
  - Existem várias métricas
    - Distância Euclidiana
    - Distância Manhattan (bloco-cidade)
    - Distância quadrática
    - Distância de Mahalanobis
    - ...

Algoritmos hierárquicos

- Como escolher uma partição?
  - Partição com n clusters
    - Selecionando partição com *n* clusters na sequência de agrupamentos da hierarquia
  - Partição que melhor se encaixa nos dados
    - Procurar no dendograma grandes mudanças em níveis adjacentes
      - Nesse caso, uma mudança de *j* para *j-1* grupos pode indicar que *j* é o melhor número de grupos
      - Existem outros procedimentos, alguns mais objetivos

Algoritmos hierárquicos

- Existe uma grande variedade de algoritmos hierárquicos
  - Geralmente diferem na forma de calcular distância inter-clusters (entre grupos)

$$d_{AB} = \min_{\substack{i \in A \\ j \in B}} (d_{ij})$$ Por ligação simples (single-link)

$$d_{AB} = \max_{\substack{i \in A \\ j \in B}} (d_{ij})$$ Por ligação completa (complete-link)

$$d_{AB} = \frac{1}{n_A n_B} \sum_{i \in A} \sum_{j \in B} d_{ij}$$ Pela média do grupo (average-link)

Validação de agrupamentos

- Como avaliar os clusters gerados por um algoritmo de agrupamento?
  - Especialista no domínio dos dados
    - Demorado para grandes conjuntos de dados
    - Subjetivo
  - Existem várias medidas de validação para agrupamento de dados
    - Julgam aspectos diferentes

Validação de agrupamentos

- Por que avaliar agrupamentos?
  - Para evitar encontrar padrões em ruídos
  - Para comparar algoritmos de agrupamento
  - Para comparar duas partições
  - Para comparar dois grupos

Medidas de validação

- Podem ser divididas em três grupos
  - Índices ou critérios internos
    - Medem a qualidade da partição obtida sem considerar informações externas
  - Índices ou critérios relativos
    - Usados para comparar duas partições ou grupos
  - Índices ou critérios externos
    - Medem o quanto os rótulos dos grupos coincidem com a classe verdadeira

Medidas internas

- Coesão de clusters
  - Mede o quão relacionados estão os objetos dentro de um cluster
- Separação de clusters
  - Mede quão distinto ou separado um cluster é dos demais clusters

Medidas internas

- Silhueta
  - Combina coesão com separação
  - Calculada para cada objeto que faz parte de um agrupamento
    - Baseada em:
      - Distância entre os objetos de um mesmo cluster e
      - Distância dos objetos de um cluster ao cluster mais próximo

Medidas internas

- Silhueta
  - Para cada objeto *i*
    - *a(i)* = distância média de *i* aos outros objetos de seu cluster
    - *b(i)* = min (distância média de *i* aos objetos dos outros clusters)

$$s(i) = \begin{cases} 1 - a(i)/b(i), & \text{se } a(i) < b(i) \\ 0, & \text{se } a(i) = b(i) \\ b(i)/a(i) - 1, & \text{se } a(i) > b(i) \end{cases}$$

  - Largura média da silhueta
    - Média sobre todos os objetos do conjunto de dados
    - Valor entre -1 e 1 (quanto mais próximo de 1, melhor)

Medidas externas

- Medidas orientadas a similaridade
  - Comparam duas partições
    - Índice Rand

$$Rand = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

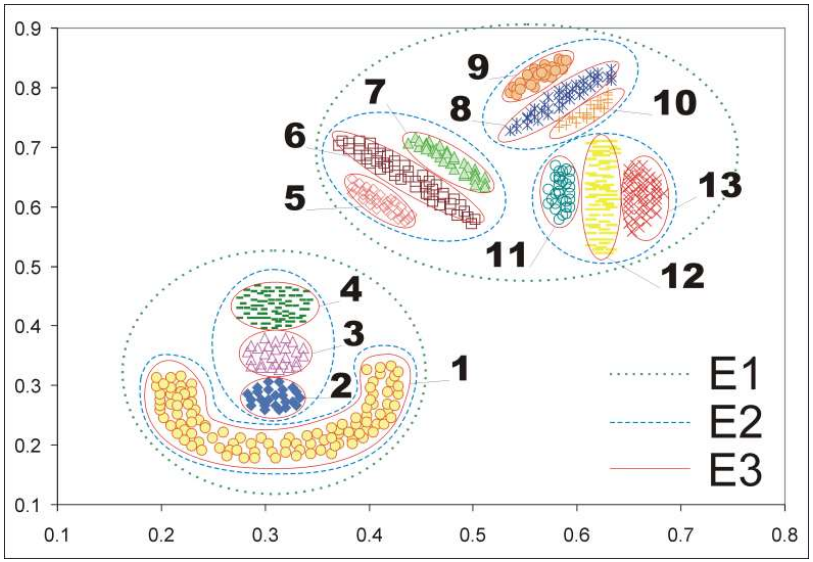
    - Jackard

$$Jac = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

  - Onde:
    - f_{00} = número de pares de objetos com classes e clusters diferentes
    - f_{01} = número de pares de objetos com classes diferentes e mesmo cluster
    - f_{10} = número de pares de objetos com mesma classe e clusters diferentes
    - f_{11} = número de pares de objetos com mesmas classes e clusters

Dificuldades

- Um mesmo conjunto de dados pode ter mais de uma estrutura relevante
  - Cada estrutura obedece uma definição de cluster ou nível de refinamento diferente
  - Análise de agrupamento tradicional busca por uma única estrutura dos dados
    - Limita a quantidade de conhecimento que pode ser obtido



Combinação de agrupamentos

- Objetivo:
  - Obter partições de melhor qualidade
- Vantagens:
  - Robustez frente a diferentes conformações dos dados
  - Novidade
    - Partições novas que não poderiam ser obtida com nenhum algoritmo, individualmente
  - Estabilidade
    - Obtém partições com menor sensibilidade a ruídos, outliers, variações de amostragem ou variabilidade dos algoritmos

Abordagens existentes

- *Ensemble* de agrupamentos
  - Geração de um conjunto de partições base
    - O mais diverso possível
  - Combinação das partições base em uma partição consenso
    - Utilizando uma função consenso
- Agrupamento multi-objetivo
  - Otimização simultânea de dois ou mais critérios de agrupamento complementares

Abordagens existentes

- *Ensemble* multi-objetivo
  - Combina as duas abordagens:
    - Gera um conjunto de partições base
    - Combina iterativamente por meio de uma técnica de ensemble
      - Ao mesmo em que seleciona as partições mais significativas com uma técnica multi-objetivo
    - Satisfação de mais de uma medida de validação

Comparação das abordagens

- Ensemble:
  - Obtém uma única partição
  - Precisa de ajustes finos de parâmetros
  - Influenciado por partições iniciais de baixa qualidade
- Agrupamento multi-objetivo:
  - Não requer muitos ajustes de parâmetros
  - Resulta em um número elevado partições
  - Tem indicações das melhores para o algoritmo de otimização
    - Não necessariamente as melhores que a técnica pode obter

Aplicações

- Compressão (redução) de dados
    - Representa cada cluster como um único dado
- Formulação de hipóteses sobre a natureza dos dados
- Verificar hipóteses sobre os dados
    - Que atributos são correlacionados
    - Que atributos são independentes
- Predição baseada em grupos

Considerações finais

- Abordagens tradicionais de agrupamento são muito utilizadas em AM
    - Várias definições de agrupamento
    - Diversos algoritmos
- Dificuldade de validar agrupamentos encontrados
- Semi-supervisionado

Fim da apresentação