Introdução à Clusterização de Dados

Guilherme de Alencar Barreto

gbarreto@ufc.br

Grupo de Aprendizado de Máquinas – GRAMA
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará – UFC
http://lattes.cnpq.br/8902002461422112

Pré-Requisitos

1. Noções de Álgebra Linear
2. Noções de Geometria Analítica
3. Noções de Funções
4. Noções de Limites

Objetivo Geral

Apresentar os fundamentos do algoritmo K-médias, de metodologias de validação de agrupamentos e de formas (qualitativa e quantitativa) de reportar os resultados por agrupamento.

Conteúdo dos Slides

1. Definições Preliminares
2. Algoritmos K-Médias (*batch* e sequencial)
3. Algoritmos K-Medianas, K-Medóides e Fuzzy K-Médias
4. Técnicas de Validação de Agrupamentos
5. Análise Quantitativa dos Resultados
6. Análise Qualitativa dos Resultados
7. Exemplos Variados

- Clusterização de dados é uma tarefa **não-supervisionada**, uma vez que não se tem informação prévia sobre classes às quais os dados pertencem.
- Para tanto, um algoritmo de clusterização deve usar apenas **informações extraídas dos próprios dados**, buscando agrupá-los por similaridade.
- Algoritmos de clusterização do tipo **particional** dividem o espaço de atributos em células, regiões ou simplesmente partições, não-superpostas, em geral com o auxílio de vetores-protótipos.
- Cada vetor de atributos é então associado a um dos protótipos existentes por critérios de similaridade, por exemplo, menor distância.

Definições Preliminares

Considere um conjunto de dados $\mathcal{X}$ formado por N vetores de atributos sem seus respectivos rótulos

$$\boxed{\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^{N}, \text{ tal que } \mathbf{x}_n \in \mathbb{R}^p} \quad (1)$$

em que

- p é a dimensão do vetor de atributos.
- N é a cardinalidade de $\mathcal{X}$:

$$\boxed{card(\mathcal{X}) = \sharp\mathcal{X} = N} \quad (2)$$

Definição de Partição

- O objetivo da clusterização é dividir os N vetores de dados em K grupos ($K \ll N$) com o auxílio de K protótipos devidamente posicionados no espaço dos dados.
- O conjunto de K protótipos é representado como segue:

$$\boxed{\mathcal{W} = \{\mathbf{w}_i\}_{i=1}^{K}, \text{ tal que } \mathbf{w}_i \in \mathbb{R}^p} \tag{3}$$

- A partição associada ao protótipo $\mathbf{w}_i$ é definida como

$$\boxed{V_i = \{\mathbf{x} \in \mathbb{R}^p \;\mid\; \|\mathbf{x} - \mathbf{w}_i\|^2 < \|\mathbf{x} - \mathbf{w}_j\|^2, \forall j \neq i\}} \tag{4}$$

em que $\|\cdot\|$ denota a distância euclidiana. O número de elementos de V_i (i.e., sua cardinalidade) é $\sharp V_i = N_i$.

Algoritmo K-Médias - Conceituação

- Usa-se o algoritmo K-médias[a] para posicionar os protótipos no espaço dos dados. Feito isso, os protótipos passam a ser os representantes dos objetos mais próximos deles.
- Para avaliar o posicionamento dos protótipos usa-se a *soma das distâncias quadráticas* (SSD, sigla em Inglês) de um objeto ao protótipo mais próximo:

$$SSD(K) = \sum_{i=1}^{K} \sum_{\forall \mathbf{x} \in V_i} \|\mathbf{x} - \mathbf{w}_i\|^2 \tag{5}$$

em que V_i é a partição de dados associada ao protótipo $\mathbf{w}_i$.

[a] MacQueen, J. B. (1967). *Some Methods for classification and Analysis of Multivariate Observations*. Proc. 5th Berkeley Symp. on Mathematical Statistics and Probability, pp. 281–297.

Algoritmo K-Médias (versão *batch*)

Passo 1 - Definir um valor para K e atribuir valores iniciais aos protótipos: $\mathbf{w}_i(0)$, $i = 1, \ldots, K$. Fazer $k = 1$.

Passo 3 - Determinar a partição $V_i(k)$ associada aos protótipos $\mathbf{w}_i(k-1)$, $i = 1, \ldots, K$.

Passo 4 - Atualizar o i-ésimo protótipo pela média dos $N_i(k)$ vetores em sua partição atual, $V_i(k)$:

$$\boxed{\mathbf{w}_i(k) = \frac{1}{N_i(k)} \sum_{\mathbf{x} \in V_i(k)} \mathbf{x}} \tag{6}$$

Passo 5 - Fazer $k = k + 1$. Repetir os **Passos** 3 e 4 até a convergência do algoritmo.

Algoritmo K-Médias (versão *sequencial*)

Passos 1-2 - Iguais ao da versão *batch*.

Passo 3 - Encontrar o índice do protótipo mais próximo ao vetor de atributos atual $\mathbf{x}(t)$:

$$i^*(t) = \arg\min_{\forall i} \|\mathbf{x}(t) - \mathbf{w}_i(t)\|^2 \quad (7)$$

em que $\|\cdot\|$ é a norma euclidiana.

Passo 4 - Atualizar a posição do protótipo $\mathbf{w}_{i^*}(t)$ usando a seguinte regra recursiva:

$$\begin{aligned} \mathbf{w}_{i^*}(t+1) &= \mathbf{w}_{i^*}(t) + \alpha_{i^*}(t)(\mathbf{x}(t) - \mathbf{w}_{i^*}(t)), && (8) \\ &= (1 - \alpha_{i^*}(t))\mathbf{w}_{i^*}(t) + \alpha_{i^*}(t)\mathbf{x}(t), && (9) \end{aligned}$$

em que $\alpha_{i^*}(t) = 1/C_{i^*}(t)$, com $C_{i^*}(t)$ sendo o número de vezes que o vetor $\mathbf{w}_{i^*}$ foi selecionado segundo a Eq. (7).

Passo 5 - Repetir os **Passos** 3 e 4 até a convergência do algoritmo.

Algoritmo K-Médias - Comentários 1

- Para iniciar os K protótipos selecione aleatoriamente K objetos (i.e. vetores de atributos) do conjunto de dados.
- Considera-se que o algoritmo convergiu se as posições dos protótipos não mudam após algumas iterações.
- Para avaliar quantitativamente a convergência do algoritmo K-médias calcula-se a SSD por iteração do algoritmo e faz-se um gráfico de $SSD(k) \times k$, onde k é a iteração atual do algoritmo.

Algoritmo K-Médias - Comentários 2

- Por depender da escolha dos K protótipos iniciais, a posição final dos mesmos pode variar em função da inicialização.
- Recomenda-se, portanto, repetir a execução do algoritmo K-médias por N_r rodadas independentes. A cada rodada, os protótipos devem ser iniciados com valores diferentes.
- A cada rodada, deve-se calcular a SSD após a convergência do algoritmo para aquela rodada.
- Escolher os protótipos da rodada que produzir o menor valor para SSD.

Algoritmo K-Médias - Comentários 3

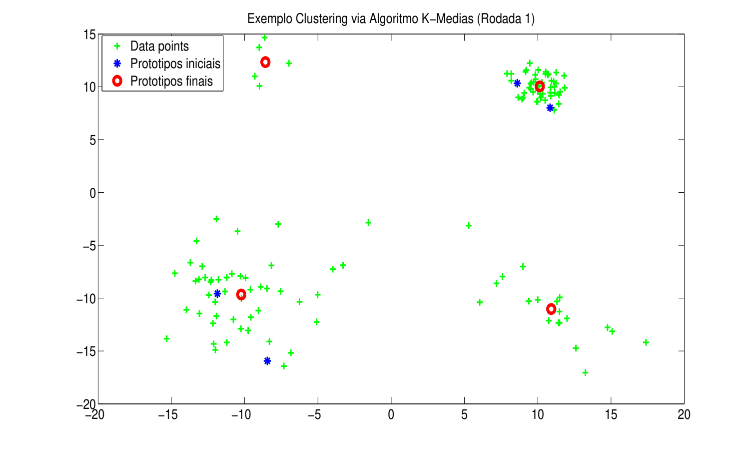
- Como o cálculo da SSD tem elevado custo computacional,principalmente para grandes volumes de dados, pode-seoptar por avaliar a convergência através da evolução da norma quadrática do vetor $\Delta\mathbf{w}_i$, ou seja

$$\|\Delta\mathbf{w}_i(k)\|^2 = \|\mathbf{w}_i(k) - \mathbf{w}_i(k-1)\|^2, \qquad (10)$$

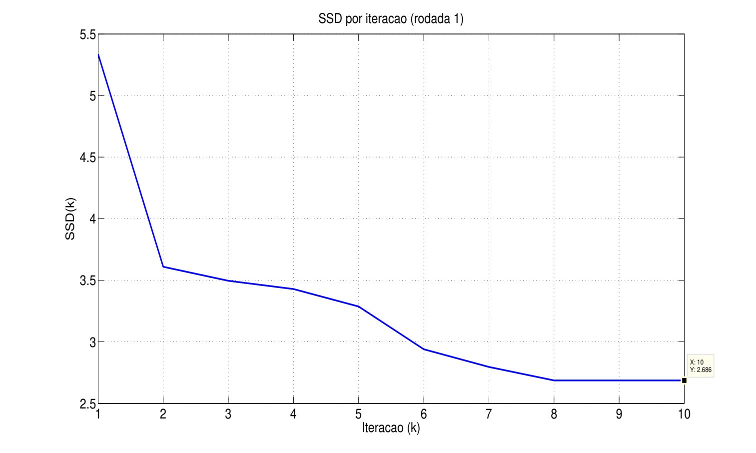
em que k denota a iteração atual do algoritmo K-médias.

- Neste caso, a SSD será calculada apenas uma vez, ao final da convergência do algoritmo.

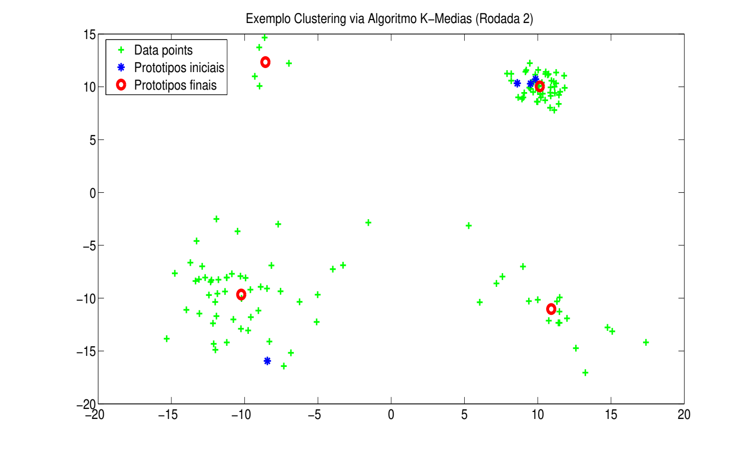
Exemplo 1: Posições Inicial e Final dos Protótipos (Rodada 1)



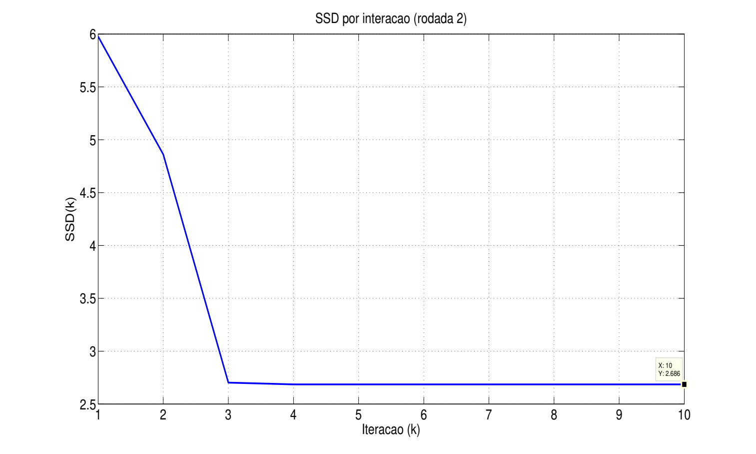
Exemplo 1: Evolução da SSD por iteração (Rodada 1)



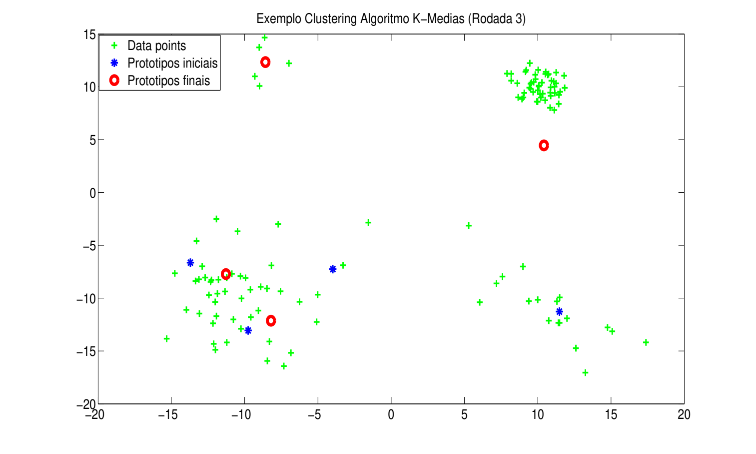
Exemplo 1: Posições Inicial e Final dos Protótipos (Rodada 2)



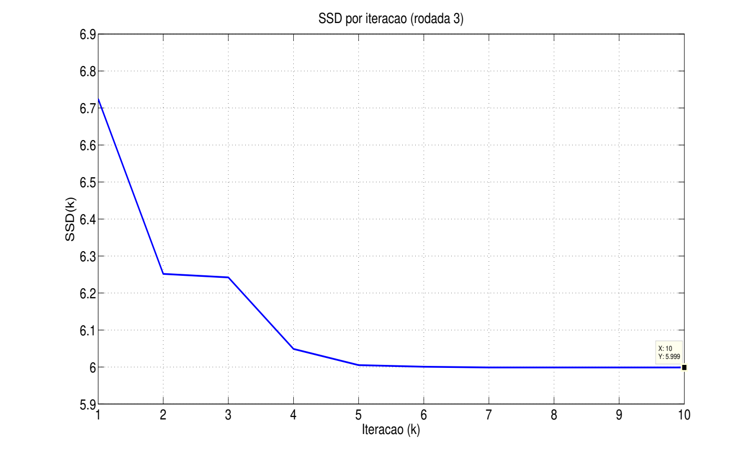
Exemplo 1: Evolução da SSD por iteração (Rodada 2)



Exemplo 1: Posições Inicial e Final dos Protótipos (Rodada 3)



Exemplo 1: Evolução da SSD por iteração (Rodada 3)



Exemplo 1: Discussão dos Resultados

- No exemplo, foram executadas $N_r = 3$ rodadas de treinamento dos $K = 4$ protótipos.
- Note que as posições iniciais dos 4 protótipos são diferentes em cada rodada.
- Das rodadas mostradas, as rodadas 1 e 2 levam ao que parece ser a configuração final ótima para os protótipos.
- Na rodada 3, os protótipos convergiram para posições finais muito ruins, o que resulta em um valor muito alto para a SSD em comparação com as duas rodadas anteriores.
- Recomenda-se repetir a execução para um número alto de rodadas. Eu recomendo $N_r = 10$, pelo menos.

Algoritmo K-Medianas

Passos 1 e 2 - Iguais ao do K-médias.

Passo 3 - Determinar a partição V_i do protótipo $\mathbf{w}_i$, $i = 1, \ldots, K$, usando a distância quarteirão em vez da distância euclidiana:

$$V_i = \{\mathbf{x} \in \mathbb{R}^p \ | \ |\mathbf{x} - \mathbf{w}_i| < |\mathbf{x} - \mathbf{w}_j|, \forall j \neq i\}, \qquad (11)$$

em que $|\mathbf{x} - \mathbf{w}_i| = \sum_{l=1}^{p} |x_l - w_{il}|$.

Passo 4 - Calcular a nova posição do protótipo $\mathbf{w}_i$ como o vetor-mediana dos N_i objetos da partição V_i:

$$\mathbf{w}_i = \text{mediana}\{\forall \mathbf{x} \in V_i\}. \qquad (12)$$

Passo 5 - Repetir os **Passos** 3 e 4 até a convergência do algoritmo.

Algoritmo K-Medóides - Conceituação

- O algoritmo K-medóides[a] é um algoritmo de clusterização relacionado ao algoritmo K-medianas. Ambos são algoritmos particionais que tentam minimizar a distância entre os pontos de um dado grupo e um ponto designado como o centro desse grupo.
- Diferentemente do K-medianas, o K-medóides escolhe pontos do próprio conjunto de dados como centros dos grupos. E em vez da distância quarteirão convencional, usa uma versão generalizada dela.
- Um medóide pode ser definido como um objeto de um grupo cuja distância média para todos os objetos em um grupo é mínima. Ou seja, é aquele ponto mais centralmente localizado no grupo.
- Os algoritmos K-medóides e K-medianas tendem a ser mais robustos a ruído e outliers que o K-médias por usarem métricas de distância menos sensíveis a valores extremos que a distância euclidiana.

[a] Kaufman, L. and Rousseeuw, P. J. (1987), *Clustering by means of Medoids*, in Statistical Data Analysis Based on the L_1-Norm and Related Methods. Edited by Y. Dodge, North-Holland, 405-416.

K-Médias $\times$ K-Medianas

- *Outliers* surgem com frequência em tarefas de reconhecimento de padrões, tais como classificação, clusterização e regressão.
- *Outliers* são pontos que destoam dos demais do conjunto de dados, por não satisfazer as suposições feitas, implícita ou explicitamente, durante a construção do modelo matemático/probabilístico/computacional que "explica" os dados disponíveis.
- Acredite, para todo modelo, seja um modelo linear, uma rede neural, ou uma máquina de vetores-suporte, existem suposições subjacentes ao seu desenvolvimento e uso, tais como
  - O modelo que explica os dados é linear ou não?
  - Os dados são estacionários ou não?
  - Os dados são suficientes para a contrução do modelo?
  - Os resíduos, ou seja, a parte dos dados que o modelo não consegue explicar, podem ser considerados como uma amostra de ruído branco gaussiano?

K-Médias $\times$ K-Medianas

- Em dados uni-, bi- ou tridimensionais, é possível detectar *outliers* por meio de inspeção visual. Porém, em dados de dimensão elevada ($p > 3$), a detecção de outliers deve ser feita de outra maneira.
- Uma possibilidade consiste em usar o procedimento de classificação de classe única, já discutido no contexto de detecção de anomalias.
- Em clusterização, uma pergunta interessante que surge neste contexto da presença de *outliers* nos dados é a seguinte:

  Sabendo que a ocorrência de outliers é comum em Ciência de Dados, deve-se dar preferência ao algoritmo K-médias ou ao K-medianas?

- Um procedimento empírico bem simples é testar os dois algoritmos. Se o conjunto de dados for "bem comportado"; ou seja, livre de *outliers* e gaussiano, as posições finais dos protótipos serão as mesmas. Se estas posições diferirem muito entre si, provavelmente há *outliers* nos dados.

Algoritmo K-Médias com Distância de Mahalanobis

- **Passo 1** - Inicialização:

Número de clusters - Definir um valor para K $(K > 1)$.

Vetores Protótipos - Atribuir valores iniciais aos protótipos: $\mathbf{w}_i(0)$, $i = 1, \ldots, K$.

Matrizes de Covariância - Atribuir valores iniciais às matrizes de covariância: $\mathbf{C}_i(0) = \mathbf{I}_p$, $i = 1, \ldots,$ K, em que $\mathbf{I}_p$ é a matriz identidade $p \times p$.

Fazer $k = 1$.

Algoritmo K-Médias com Distância de Mahalanobis

- **Passo 2** - Determinar a partição $V_i(k)$ associada aos protótipos $\mathbf{w}_i(k-1)$ e matrizes de covariância $\mathbf{C}_i(k-1)$, $i = 1, \ldots, K$, usando distância de Mahalanobis:

$$\boxed{V_i(k) = \{\mathbf{x} \in \mathbb{R}^p \mid g_i^{(k)}(\mathbf{x}) < g_j^{(k)}(\mathbf{x}), \ \forall j \neq i\}}, \tag{13}$$

com as funções discriminantes $g_i^{(k)}(\mathbf{x})$ e $g_j^{(k)}(\mathbf{x})$ dadas por

$$g_i^{(k)}(\mathbf{x}) = (\mathbf{x} - \mathbf{w}_i(k))^T \mathbf{C}_i^{-1}(k)(\mathbf{x} - \mathbf{w}_i(k)) \tag{14}$$

e

$$g_j^{(k)}(\mathbf{x}) = (\mathbf{x} - \mathbf{w}_j(k))^T \mathbf{C}_j^{-1}(k)(\mathbf{x} - \mathbf{w}_j(k)). \tag{15}$$

OBS: Caso a matriz de covariância apresente problemas de inversão, pode-se utilizar alguma estratégia de regularização da mesma.

Algoritmo K-Médias com Distância de Mahalanobis

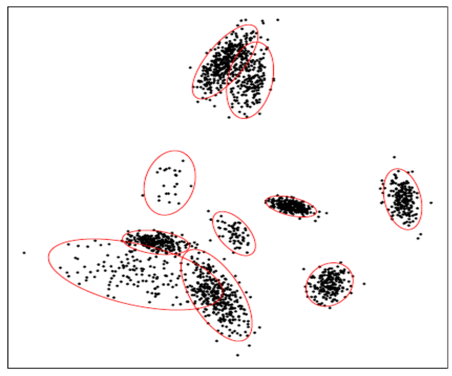
- **Passo 3** - Atualizar o i-ésimo protótipo e a correspondente matriz de covariância usando os $N_i(k)$ vetores de sua partição atual, $V_i(k)$:

$$\boxed{\mathbf{w}_i(k) = \frac{1}{N_i(k)} \sum_{\forall \mathbf{x} \in V_i(k)} \mathbf{x}} \tag{16}$$

$$\boxed{\mathbf{C}_i(k) = \frac{1}{N_i(k)} \sum_{\forall \mathbf{x} \in V_i(k)} \mathbf{x}\mathbf{x}^T} \tag{17}$$

- **Passo 4** - Fazer $k = k + 1$. Repetir os **Passos** 2 a 4 até a convergência do algoritmo.

Algoritmo K-médias com distância de Mahalanobis



Fonte: I. Melnykov & V. Melnykov (2014). "On K-means algorithm with the use of Mahalanobis distances", *Statistics and Probability Letters*, vol. 84, pp. 88-95.

Algoritmo K-Médias Fuzzy - Conceituação

- A pertinência[a] de um objeto $\mathbf{x}$ ao i-ésimo cluster é denotada $\mu_i(\mathbf{x})$, tal que $\mu_i(\mathbf{x}) \in [0,1]$ e $\sum_{i=1}^{K} \mu_i(\mathbf{x}) = 1$.
- A versão fuzzy da soma das distâncias quadráticas (FSSD, sigla em Inglês) é dada por

$$FSSD(K) = \sum_{i=1}^{K} \sum_{\forall \mathbf{x} \in V_i} \|\mathbf{x} - \mathbf{w}_i\|^2 (\mu_i(\mathbf{x}))^z, \qquad (18)$$

em que $z > 1$ representa o grau de nebulosidade da função. Quanto maior z, mais nebuloso é o agrupamento. Se $z = 0$, tem-se o algoritmo K-médias clássico.

[a] J. C. Bezdek (1981): "Pattern Recognition with Fuzzy Objective Function Algorithms". Plenum Press, New York.

Algoritmo K-Médias Fuzzy (versão *batch*)

Passo 1 - Definir K e inicializar os K protótipos.

Passo 2 - Determinar o grau de pertinência dos vetores de atributos aos K protótipos:

$$\mu_i(\mathbf{x}) = \left(\sum_{j=1}^{K} \left(\frac{\|\mathbf{x} - \mathbf{w}_i\|}{\|\mathbf{x} - \mathbf{w}_j\|} \right)^{\frac{2}{z-1}} \right)^{-1} \tag{19}$$

Passo 3 - Calcular a nova posição do protótipo $\mathbf{w}_i$ como a média dos N_i objetos da partição V_i ponderada pelas pertinências:

$$\mathbf{w}_i = \frac{\sum_{\mathbf{x} \in V_i} \mu_i(\mathbf{x})\mathbf{x}}{\sum_{\mathbf{x} \in V_i} \mu_i(\mathbf{x})} \tag{20}$$

Passo 5 - Repetir os **Passos** 2 e 4 até a convergência dos protótipos.

Índices de Validação de Agrupamentos

- Nos algoritmos particionais descritos anteriormente, é preciso especificar de antemão o valor de K. Em outras palavras, K é um hiperparâmetro do algoritmo.
- Contudo, em um problema de clusterização, com freqüência não sabemos qual é o valor ideal ou mais dequado ao conjunto de dados de interesse.
- Nestes casos, faz-se necessário uma investigação sistemática com o objetivo de encontrar um ou mais valores de K que sejam úteis ao processo de análise dos agrupamentos encontrados.
- Para isso, costuma-se fazer uso de índices de validação de agrupamentos.

Índices de Validação de Agrupamentos

- Em geral, os vários índices de validação de agrupamentos existentes procuram avaliar os dois seguintes aspectos do particionamento:
  1. **Coesão Interna** (ou Intragrupo): Os objetos em um agrupamento deveriam ser tão similares entre si quanto possível. Medidas de distância entre os objetos de um agrupamento ou dos elementos deste agrupamento ao seu protótipo fornecem uma indicação de sua coesão ou do seu grau de compactação.
  2. **Separação Externa** (ou Entregrupos): Agrupamentos deveriam estar, em princípio, bem separados. As distâncias (e.g. euclidianas) entre os protótipos dão uma indicação do grau de separação entre os diversos agrupamentos.

Índice Dunn

- O índice Dunn[a], para um dado valor de K, é calculado como

$$DI(K) = \frac{\min_{i \neq j}\{\delta(V_i, V_j)\}}{\max_{1 \leq l \leq K}\{\Delta(V_l)\}} \tag{21}$$

  em que

  1. $\delta(V_i, V_j)$ denota uma medida de dissimilaridade (e.g. distância euclidiana) entre as partições V_i e V_j.
  2. $\Delta(V_l)$ é uma medida da dispersão dos dados da partição V_l.

- Para $K = 2, \ldots, K_{max}$, o maior valor de $DI(K)$ indica uma partição válida ótima.

[a] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters", *Journal of Cybernetics*, 3(3):32–57, 1973.

Índice Dunn (cont.)

- A separação entre clusters $\delta(V_i, V_j)$ é definida como

$$\delta(V_i, V_j) = \min_{\mathbf{x} \in V_i, \mathbf{y} \in V_j} \{d(\mathbf{x}, \mathbf{y})\} \tag{22}$$

Ou seja, é a menor distância entre um elemento da partição V_i e um da partição V_j, $i, j = 1, \ldots, K$.

- E a coesão interna do cluster $\Delta(V_l)$ é dada por

$$\Delta(V_l) = \max_{\mathbf{x}, \mathbf{y} \in V_l} \{d(\mathbf{x}, \mathbf{y})\} \tag{23}$$

Ou seja, é a maior distância entre os elementos da partição V_l, $l = 1, \ldots, K$.

Índice Davies-Bouldin (DB)

- O índice DB[a] é calculado pela seguinte expressão:

$$DB(K) = \frac{1}{K} \sum_{i=1}^{K} R_{i,qt} \tag{24}$$

com $R_{i,qt}$ sendo a razão entre medidas de dispersão intra- e entregrupos:

$$R_{i,qt} = \max_{\forall j, j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right\}. \tag{25}$$

- Para $K = 2, \ldots, K_{max}$, o menor valor de $DB(K)$ indica uma partição válida ótima.

[a] D. L. Davies and D. W. Bouldin, "A cluster separation measure", IEEE Trans. on Pattern Analysis and Machine Intelligence, 1(2):95–104, 1979.

Índice DB (cont.)

- A dispersão interna ao i-ésimo grupo é dada por

$$S_{i,q} = \left[\frac{1}{N_i} \sum_{\mathbf{x} \in V_i} \|\mathbf{x} - \mathbf{w}_i\|^q \right]^{1/q}, \quad q \text{ é inteiro} \geq 1. \tag{26}$$

- A separação entre V_i e V_j é dada pela distância de Minkowski de ordem t entre os protótipos $\mathbf{w}_i$ e $\mathbf{w}_j$:

$$d_{ij,t} = \left\{ \sum_{l=1}^{p} |w_{i,l} - w_{j,l}|^t \right\}^{1/t} = \|\mathbf{w}_i - \mathbf{w}_j\|_t \tag{27}$$

onde p é a dimensão do vetor de atributos e $|\cdot|$ é o valor absoluto.

Índice Calinski-Harabasz (CH)

- O índice CH[a] é calculado como

$$CH(K) = \frac{\mathrm{tr}(\mathbf{S}_B)/(K-1)}{\mathrm{tr}(\mathbf{S}_W)/(N-K)} \tag{28}$$

  em que $\mathbf{S}_B$ é a matriz de dispersão entregrupos e $\mathbf{S}_W$ é a matriz de dispersão intragrupo, para N dados particionados em K grupos.
- O operador $\mathrm{tr}(\cdot)$ denota o traço de uma matriz, ou seja, a soma dos elementos de sua diagonal principal.
- Para $K = 2, \ldots, K_{max}$, o maior valor de $CH(K)$ indica uma partição válida ótima.

[a] R. B. Calinski and J. Harabasz, "A dendrite method for cluster analysis", *Communications in Statistics*, 3(1):1–27, 1974.

Índice CH (cont.)

- A matriz de dispersão entregrupos é calculada como

$$\mathbf{S}_B = \sum_{i=1}^{K} N_i(\mathbf{w}_i - \mathbf{m})(\mathbf{w}_i - \mathbf{m})^T, \tag{29}$$

em que N_i é o número de elementos da partição V_i, $\mathbf{w}_i$ é o protótipo da partição V_i e $\mathbf{m}$ é o vetor médio de todo o conjunto dos dados.

- A matriz de dispersão intragrupo é calculada como

$$\mathbf{S}_W = N_1\mathbf{C}_1 + \cdots + N_K\mathbf{C}_K = \sum_{i=1}^{K} N_i\mathbf{C}_i, \tag{30}$$

em que $\mathbf{C}_i$ é a matriz de covariância do i-ésimo grupo.

Passo 1 - Normalizar os dados.

Passo 2 - Para cada valor de $K = 2, \ldots, K_{max}$, fazer

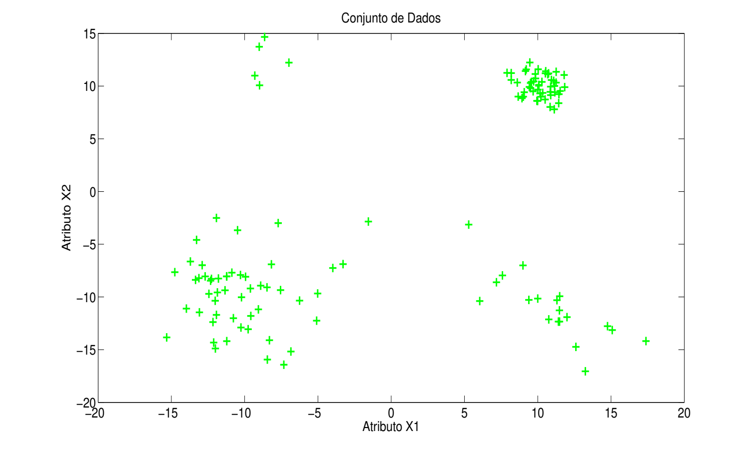
1. Aplicar o algoritmo K-médias por N_r rodadas.
2. Escolher os protótipos da rodada que produzir menor SSD.
3. Calcular o valor do índice de validação escolhido.

Passo 3 - Escolher o valor ótimo K_{opt} como aquele que otimiza o índice de validação de agrupamentos escolhido.

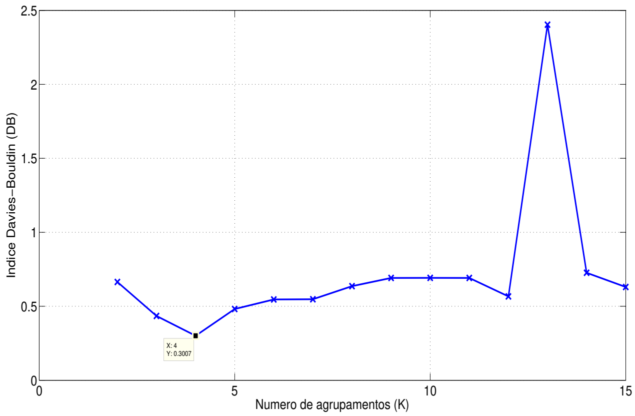
Passo 4 - Particionar os dados entre os K_{opt} agrupamentos usando o critério da distância euclidiana ao protótipo mais próximo.

Passo 5 - Reportar estatísticas descritivas dos atributos por agrupamento (e.g. valores médio, mínimo e máximo, mediana e número de exemplos por agrupamento).

Exemplo 2: Visualização do Conjunto de Dados



Exemplo 2: Índice DB para diferentes valores de K



Exemplo 2: Tabela c/ valores do índice DB ($K = 2, \ldots, 10$).

K	2	3	**4**	5	6	7	8	9	10
DB	0.664	0.435	**0.301**	0.481	0.546	0.547	0.636	0.692	0.692

Exemplo 2: Tabela com $K_{opt} = 4$ protótipos finais.

$\mathbf{w}_1$	10.9217	-11.0335
$\mathbf{w}_2$	-8.5836	12.3378
$\mathbf{w}_3$	-10.2313	-9.6603
$\mathbf{w}_4$	10.1427	10.0429

Exemplo 2: Tabelas-Sumário da Clusterização

Cluster 1 ($N_1 = 19$ objetos)					
Estatísticas	Mín.	Máx.	Médio	Mediana	Desv.Pad.
Atributo X_1	5.2896	17.3873	10.9217	11.4100	3.0923
Atributo X_2	-17.0521	-3.1384	-11.0335	-11.2713	3.0740

Cluster 2 ($N_2 = 5$ objetos)					
Estatísticas	Mín.	Máx.	Médio	Mediana	Desv.Pad.
Atributo X_1	-9.3055	-6.9810	-8.5836	-8.9862	0.9265
Atributo X_2	10.0764	14.6591	12.3378	12.2255	1.8888

Cluster 3 ($N_3 = 51$ objetos)					
Estatísticas	Mín.	Máx.	Médio	Mediana	Desv.Pad.
Atributo X_1	-15.3200	-1.5591	-10.2313	-10.8711	2.9927
Atributo X_2	-16.4247	-2.5098	-9.6603	-9.3495	3.3285

Cluster 4 ($N_4 = 50$ objetos)					
Estatísticas	Mín.	Máx.	Médio	Mediana	Desv.Pad.
Atributo X_1	7.9052	11.8384	10.1427	10.1140	0.9775
Atributo X_2	7.7985	12.2370	10.0429	10.0229	1.0397