DATA MINING & MACHINE LEARNING (I)

Thiago Marzagão

clusterização

- Temos x_1, x_2, ..., x_p e queremos dividir as amostras em clusters.

clusterização

- Temos x_1, x_2, ..., x_p e queremos dividir as amostras em clusters.
- Exemplo: agrupar usuários do Netflix conforme os gêneros preferidos (romance, comédia, terror, ficção científica, etc).

clusterização

- Temos x_1, x_2, ..., x_p e queremos dividir as amostras em clusters.
- Exemplo: agrupar usuários do Netflix conforme os gêneros preferidos (romance, comédia, terror, ficção científica, etc).
- Seja:

clusterização

- Temos x_1, x_2, ..., x_p e queremos dividir as amostras em clusters.
- Exemplo: agrupar usuários do Netflix conforme os gêneros preferidos (romance, comédia, terror, ficção científica, etc).
- Seja:
- $x_1 =$ quantos filmes/seriados de romance o usuário já assistiu no Netflix

clusterização

- Temos x_1, x_2, ..., x_p e queremos dividir as amostras em clusters.
- Exemplo: agrupar usuários do Netflix conforme os gêneros preferidos (romance, comédia, terror, ficção científica, etc).
- Seja:
- x_1 = quantos filmes/seriados de romance o usuário já assistiu no Netflix
- x_2 = quantos filmes/seriados de comédia o usuário já assistiu no Netflix

clusterização

- Temos x_1, x_2, ..., x_p e queremos dividir as amostras em clusters.
- Exemplo: agrupar usuários do Netflix conforme os gêneros preferidos (romance, comédia, terror, ficção científica, etc).
- Seja:
- x_1 = quantos filmes/seriados de romance o usuário já assistiu no Netflix
- x_2 = quantos filmes/seriados de comédia o usuário já assistiu no Netflix
- x_3 = quantos filmes/seriados de terror o usuário já assistiu no Netflix

clusterização

- Temos x_1, x_2, ..., x_p e queremos dividir as amostras em clusters.
- Exemplo: agrupar usuários do Netflix conforme os gêneros preferidos (romance, comédia, terror, ficção científica, etc).
- Seja:
- x_1 = quantos filmes/seriados de romance o usuário já assistiu no Netflix
- x_2 = quantos filmes/seriados de comédia o usuário já assistiu no Netflix
- x_3 = quantos filmes/seriados de terror o usuário já assistiu no Netflix
- x_4 = quantos filmes/seriados de ficção científica o usuário já assistiu no Netflix

clusterização

- Temos x_1, x_2, ..., x_p e queremos dividir as amostras em clusters.
- Exemplo: agrupar usuários do Netflix conforme os gêneros preferidos (romance, comédia, terror, ficção científica, etc).
- Seja:
- x_1 = quantos filmes/seriados de romance o usuário já assistiu no Netflix
- x_2 = quantos filmes/seriados de comédia o usuário já assistiu no Netflix
- x_3 = quantos filmes/seriados de terror o usuário já assistiu no Netflix
- x_4 = quantos filmes/seriados de ficção científica o usuário já assistiu no Netflix
- Queremos dividir os usuários do Netflix em clusters, de acordo com os gêneros preferidos.

k-means

- Vários algoritmos de clusterização são possíveis. O mais conhecido é o k-means.

k-means

- Vários algoritmos de clusterização são possíveis. O mais conhecido é o k-means.
- Objetivo: encontrar a divisão das amostras que minimiza a distância euclidiana quadrada média entre os pares de um mesmo cluster.

k-means

- Vários algoritmos de clusterização são possíveis. O mais conhecido é o k-means.
- Objetivo: encontrar a divisão das amostras que minimiza a distância euclidiana quadrada média entre os pares de um mesmo cluster.
- Matematicamente:

k-means

- Vários algoritmos de clusterização são possíveis. O mais conhecido é o k-means.
- Objetivo: encontrar a divisão das amostras que minimiza a distância euclidiana quadrada média entre os pares de um mesmo cluster.
- Matematicamente:
- $$\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

k-means

- Vários algoritmos de clusterização são possíveis. O mais conhecido é o k-means.
- Objetivo: encontrar a divisão das amostras que minimiza a distância euclidiana quadrada média entre os pares de um mesmo cluster.
- Matematicamente:
- $$\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$
- K é o número de clusters (você escolhe K)

k-means

- Vários algoritmos de clusterização são possíveis. O mais conhecido é o k-means.
- Objetivo: encontrar a divisão das amostras que minimiza a distância euclidiana quadrada média entre os pares de um mesmo cluster.
- Matematicamente:
- $$\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$
- K é o número de clusters (você escolhe K)
- $|C_k|$ é o número de amostras no cluster k

k-means

- Vários algoritmos de clusterização são possíveis. O mais conhecido é o k-means.
- Objetivo: encontrar a divisão das amostras que minimiza a distância euclidiana quadrada média entre os pares de um mesmo cluster.
- Matematicamente:
- $$\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$
- K é o número de clusters (você escolhe K)
- $|C_k|$ é o número de amostras no cluster k
- $j..p$ são as variáveis

k-means

- Vários algoritmos de clusterização são possíveis. O mais conhecido é o k-means.
- Objetivo: encontrar a divisão das amostras que minimiza a distância euclidiana quadrada média entre os pares de um mesmo cluster.
- Matematicamente:
- $$\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$
- K é o número de clusters (você escolhe K)
- $|C_k|$ é o número de amostras no cluster k
- $j..p$ são as variáveis
- i, i' é um par de amostras

k-means

- Resolver $\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$ é difícil: existem quase K^n de particionar n amostras em K clusters.

k-means

- Resolver $\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$ é difícil: existem quase K^n de particionar n amostras em K clusters.
- (NP-difícil: pelo menos tão difícil quanto os problemas mais difíceis em tempo polinomial não-determinístico.)

k-means

- Resolver $\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$ é difícil: existem quase K^n de particionar n amostras em K clusters.
- (NP-difícil: pelo menos tão difícil quanto os problemas mais difíceis em tempo polinomial não-determinístico.)
- Em vez de resolver o problema exatamente nós usamos o algoritmo de Lloyd p/ encontrar uma solução aproximada:

k-means

- Resolver $\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$ é difícil: existem quase K^n de particionar n amostras em K clusters.
- (NP-difícil: pelo menos tão difícil quanto os problemas mais difíceis em tempo polinomial não-determinístico.)
- Em vez de resolver o problema exatamente nós usamos o algoritmo de Lloyd p/ encontrar uma solução aproximada:
- 1) Designe cada amostra, aleatoriamente, a um dos K clusters.

k-means

- Resolver $\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$ é difícil: existem quase K^n de particionar n amostras em K clusters.
- (NP-difícil: pelo menos tão difícil quanto os problemas mais difíceis em tempo polinomial não-determinístico.)
- Em vez de resolver o problema exatamente nós usamos o algoritmo de Lloyd p/ encontrar uma solução aproximada:
- 1) Designe cada amostra, aleatoriamente, a um dos K clusters.
- 2) Compute o centróide de cada cluster.

k-means

- Resolver $\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$ é difícil: existem quase K^n de particionar n amostras em K clusters.
- (NP-difícil: pelo menos tão difícil quanto os problemas mais difíceis em tempo polinomial não-determinístico.)
- Em vez de resolver o problema exatamente nós usamos o algoritmo de Lloyd p/ encontrar uma solução aproximada:
- 1) Designe cada amostra, aleatoriamente, a um dos K clusters.
- 2) Compute o centróide de cada cluster.
- 3) Designe cada amostra ao cluster cujo centróide esteja mais próximo.

k-means

- Resolver $\underset{C_1,\ldots,C_k}{argmin} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$ é difícil: existem quase K^n de particionar n amostras em K clusters.
- (NP-difícil: pelo menos tão difícil quanto os problemas mais difíceis em tempo polinomial não-determinístico.)
- Em vez de resolver o problema exatamente nós usamos o algoritmo de Lloyd p/ encontrar uma solução aproximada:
- 1) Designe cada amostra, aleatoriamente, a um dos K clusters.
- 2) Compute o centróide de cada cluster.
- 3) Designe cada amostra ao cluster cujo centróide esteja mais próximo.
- 4) Repita 2 e 3 até que as amostras permaneçam nos mesmos clusters.

k-means

http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

k-means

- K-means tende a convergir p/ um ótimo local em vez de global.

k-means

- K-means tende a convergir p/ um ótimo local em vez de global.
- Duas soluções (não-excludentes):

k-means

- K-means tende a convergir p/ um ótimo local em vez de global.
- Duas soluções (não-excludentes):
- 1) Executamos k-means várias vezes e escolhemos o resultado que minimiza a soma das distâncias euclidianas quadradas médias.

k-means

- K-means tende a convergir p/ um ótimo local em vez de global.
- Duas soluções (não-excludentes):
- 1) Executamos k-means várias vezes e escolhemos o resultado que minimiza a soma das distâncias euclidianas quadradas médias.
- 2) k-means++:

k-means

- K-means tende a convergir p/ um ótimo local em vez de global.
- Duas soluções (não-excludentes):
- 1) Executamos k-means várias vezes e escolhemos o resultado que minimiza a soma das distâncias euclidianas quadradas médias.
- 2) k-means++:
  - a) escolhemos uma amostra aleatoriamente, c/ igual probabilidade p/ cada amostra ($p_i = 1/n$); chamemos essa amostra de c_1; ela será nosso primeiro centróide

k-means

- K-means tende a convergir p/ um ótimo local em vez de global.
- Duas soluções (não-excludentes):
- 1) Executamos k-means várias vezes e escolhemos o resultado que minimiza a soma das distâncias euclidianas quadradas médias.
- 2) k-means++:
  - a) escolhemos uma amostra aleatoriamente, c/ igual probabilidade p/ cada amostra ($p_i = 1/n$); chamemos essa amostra de c_1; ela será nosso primeiro centróide
  - b) calculamos a distância euclidiana quadrada entre cada amostra e c_1: $D_i^2 = (x_i - c_1)^2$

k-means

- K-means tende a convergir p/ um ótimo local em vez de global.
- Duas soluções (não-excludentes):
- 1) Executamos k-means várias vezes e escolhemos o resultado que minimiza a soma das distâncias euclidianas quadradas médias.
- 2) k-means++:
  - a) escolhemos uma amostra aleatoriamente, c/ igual probabilidade p/ cada amostra ($p_i = 1/n$); chamemos essa amostra de c_1; ela será nosso primeiro centróide
  - b) calculamos a distância euclidiana quadrada entre cada amostra e c_1: $D_i^2 = (x_i - c_1)^2$
  - c) escolhemos uma amostra aleatoriamente, c/ probabilidade $p_i = D_i^2 / \sum_{i=1}^{n} D_i^2$; chamemos essa amostra de c_2; ela será nosso segundo centróide

k-means

- K-means tende a convergir p/ um ótimo local em vez de global.
- Duas soluções (não-excludentes):
- 1) Executamos k-means várias vezes e escolhemos o resultado que minimiza a soma das distâncias euclidianas quadradas médias.
- 2) k-means++:
  - a) escolhemos uma amostra aleatoriamente, c/ igual probabilidade p/ cada amostra ($p_i = 1/n$); chamemos essa amostra de c_1; ela será nosso primeiro centróide
  - b) calculamos a distância euclidiana quadrada entre cada amostra e c_1: $D_i^2 = (x_i - c_1)^2$
  - c) escolhemos uma amostra aleatoriamente, c/ probabilidade $p_i = D_i^2 / \sum_{i=1}^{n} D_i^2$; chamemos essa amostra de c_2; ela será nosso segundo centróide
  - d) recalculamos D_i^2: $D_i^2 = \text{argmin} \left\{ (x_i - c_1)^2, (x_i - c_2)^2 \right\}$

k-means

- K-means tende a convergir p/ um ótimo local em vez de global.
- Duas soluções (não-excludentes):
- 1) Executamos k-means várias vezes e escolhemos o resultado que minimiza a soma das distâncias euclidianas quadradas médias.
- 2) k-means++:
  - a) escolhemos uma amostra aleatoriamente, c/ igual probabilidade p/ cada amostra ($p_i = 1/n$); chamemos essa amostra de c_1; ela será nosso primeiro centróide
  - b) calculamos a distância euclidiana quadrada entre cada amostra e c_1: $D_i^2 = (x_i - c_1)^2$
  - c) escolhemos uma amostra aleatoriamente, c/ probabilidade $p_i = D_i^2 / \sum_{i=1}^{n} D_i^2$; chamemos essa amostra de c_2; ela será nosso segundo centróide
  - d) recalculamos D_i^2: $D_i^2 = \text{argmin} \left\{ (x_i - c_1)^2, (x_i - c_2)^2 \right\}$
  - e) repetimos b) e c) até obtermos k centróides iniciais

k-means

- K-means tende a convergir p/ um ótimo local em vez de global.
- Duas soluções (não-excludentes):
- 1) Executamos k-means várias vezes e escolhemos o resultado que minimiza a soma das distâncias euclidianas quadradas médias.
- 2) k-means++:
  - a) escolhemos uma amostra aleatoriamente, c/ igual probabilidade p/ cada amostra ($p_i = 1/n$); chamemos essa amostra de c_1; ela será nosso primeiro centróide
  - b) calculamos a distância euclidiana quadrada entre cada amostra e c_1: $D_i^2 = (x_i - c_1)^2$
  - c) escolhemos uma amostra aleatoriamente, c/ probabilidade $p_i = D_i^2 / \sum_{i=1}^{n} D_i^2$; chamemos essa amostra de c_2; ela será nosso segundo centróide
  - d) recalculamos D_i^2: $D_i^2 = \text{argmin} \left\{ (x_i - c_1)^2, (x_i - c_2)^2 \right\}$
  - e) repetimos b) e c) até obtermos k centróides iniciais
  - f) executamos k-means

k-means

- Como escolher K?

k-means

- Como escolher K?
- Às vezes o K é óbvio dada a natureza do problema.

k-means

- Como escolher K?
- Às vezes o K é óbvio dada a natureza do problema.
- Por exemplo, se sabemos de antemão que há 10 autores e queremos clusterizar 1000 obras por autor.

k-means

- Como escolher K?
- Às vezes o K é óbvio dada a natureza do problema.
- Por exemplo, se sabemos de antemão que há 10 autores e queremos clusterizar 1000 obras por autor.
- Outras vezes não: existe um cluster "comédia" e outro "romance"? Ou apenas um cluster "comédia romântica"?

k-means

- Como escolher K?
- Às vezes o K é óbvio dada a natureza do problema.
- Por exemplo, se sabemos de antemão que há 10 autores e queremos clusterizar 1000 obras por autor.
- Outras vezes não: existe um cluster "comédia" e outro "romance"? Ou apenas um cluster "comédia romântica"?
- E como saber se o K escolhido é o "correto"?

k-means

- Como escolher K?
- Às vezes o K é óbvio dada a natureza do problema.
- Por exemplo, se sabemos de antemão que há 10 autores e queremos clusterizar 1000 obras por autor.
- Outras vezes não: existe um cluster "comédia" e outro "romance"? Ou apenas um cluster "comédia romântica"?
- E como saber se o K escolhido é o "correto"?
- Não há uma resposta simples. Só mesmo inspecionando, "no olho", a consistência interna de cada cluster. I.e., o cluster k "faz sentido"?

k-means

- Como escolher K?
- Às vezes o K é óbvio dada a natureza do problema.
- Por exemplo, se sabemos de antemão que há 10 autores e queremos clusterizar 1000 obras por autor.
- Outras vezes não: existe um cluster "comédia" e outro "romance"? Ou apenas um cluster "comédia romântica"?
- E como saber se o K escolhido é o "correto"?
- Não há uma resposta simples. Só mesmo inspecionando, "no olho", a consistência interna de cada cluster. I.e., o cluster k "faz sentido"?
- Às vezes o problema é a seleção de variáveis. A inclusão de variáveis irrelevantes pode levar a uma clusterização subótima.

k-means

- E se K for grande?

k-means

- E se K for grande?
- Aí não dá p/ examinar cada cluster "no olho".

k-means

- E se K for grande?
- Aí não dá p/ examinar cada cluster "no olho".
- Solução: silhuetas.

k-means

- E se K for grande?
- Aí não dá p/ examinar cada cluster "no olho".
- Solução: silhuetas.
- Seja a_i a distância média entre a amostra i e as demais amostras do mesmo cluster.

k-means

- E se K for grande?
- Aí não dá p/ examinar cada cluster "no olho".
- Solução: silhuetas.
- Seja a_i a distância média entre a amostra i e as demais amostras do mesmo cluster.
- Seja b_i a distância média entre a amostra i e as amostras do segundo cluster mais próximo a i.

k-means

- E se K for grande?
- Aí não dá p/ examinar cada cluster "no olho".
- Solução: silhuetas.
- Seja a_i a distância média entre a amostra i e as demais amostras do mesmo cluster.
- Seja b_i a distância média entre a amostra i e as amostras do segundo cluster mais próximo a i.
- Silhueta $= \dfrac{b_i - a_i}{max(a_i, b_i)}$

k-means

- E se K for grande?
- Aí não dá p/ examinar cada cluster "no olho".
- Solução: silhuetas.
- Seja a_i a distância média entre a amostra i e as demais amostras do mesmo cluster.
- Seja b_i a distância média entre a amostra i e as amostras do segundo cluster mais próximo a i.
- Silhueta $= \frac{b_i - a_i}{max(a_i, b_i)}$
- A silhueta da amostra i mede o quão bom é o "encaixe" de i no seu cluster. Pode variar de -1 (pior encaixe possível) a $+1$ (melhor encaixe possível).

k-means

- E se K for grande?
- Aí não dá p/ examinar cada cluster "no olho".
- Solução: silhuetas.
- Seja a_i a distância média entre a amostra i e as demais amostras do mesmo cluster.
- Seja b_i a distância média entre a amostra i e as amostras do segundo cluster mais próximo a i.
- Silhueta $= \dfrac{b_i - a_i}{max(a_i, b_i)}$
- A silhueta da amostra i mede o quão bom é o "encaixe" de i no seu cluster. Pode variar de -1 (pior encaixe possível) a $+1$ (melhor encaixe possível).
- A silhueta média de todas as amostras de um cluster mede o quanto esse cluster "faz sentido".

k-means

- E se K for grande?
- Aí não dá p/ examinar cada cluster "no olho".
- Solução: silhuetas.
- Seja a_i a distância média entre a amostra i e as demais amostras do mesmo cluster.
- Seja b_i a distância média entre a amostra i e as amostras do segundo cluster mais próximo a i.
- Silhueta $= \frac{b_i - a_i}{max(a_i, b_i)}$
- A silhueta da amostra i mede o quão bom é o "encaixe" de i no seu cluster. Pode variar de -1 (pior encaixe possível) a $+1$ (melhor encaixe possível).
- A silhueta média de todas as amostras de um cluster mede o quanto esse cluster "faz sentido".
- A silhueta média de todas as amostras de todos os clusters mede o quão boa ou ruim é a nossa clusterização.

exercícios!

- 1) Municípios brasileiros. (Vamos fazer juntos.)

exercícios!

- 1) Municípios brasileiros. (Vamos fazer juntos.)
- 2) Filmes. (Vamos fazer juntos.)

exercícios!

- 1) Municípios brasileiros. (Vamos fazer juntos.)
- 2) Filmes. (Vamos fazer juntos.)
- 3) Olimpíadas. (Vocês vão fazer sozinhos.)