Agrupamento de dados

Marcelo Keese Albertini

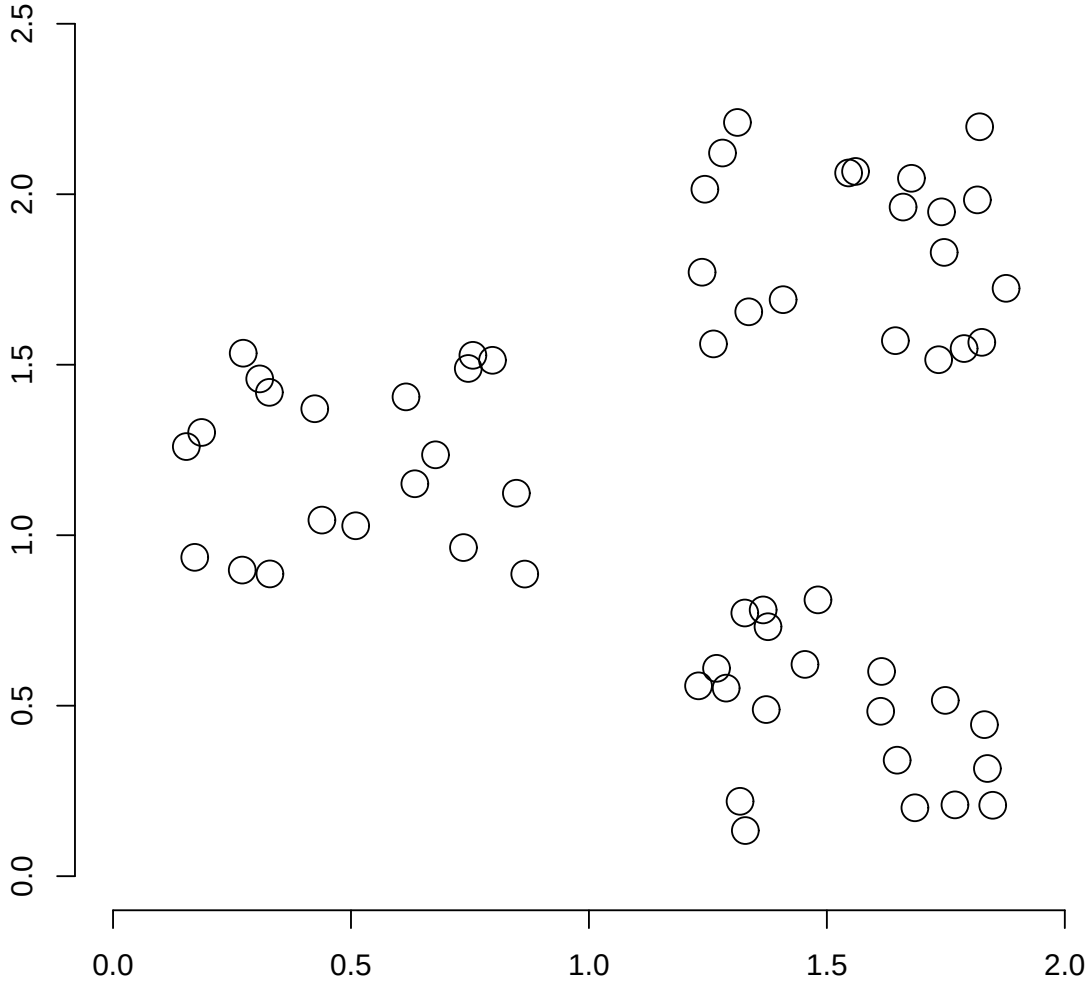
Faculdade de Computação - UFU

Conteúdo

Agrupamento: definição

- Agrupamento de documentos é o processo de organização de elementos em grupos
- Elementos em um mesmo grupo devem ser similares
- Elementos de grupos distintos devem ser dissimilares
- Agrupamento é a forma mais comum de aprendizado não supervisionado
- Não supervisionado = sem rótulos ou informação auxiliar

Algoritmo para agrupar dados



Propor algoritmo para encontrar a estrutura de agrupa-mento nessa figura

Classificação vs. agrupamento

- Classificação: aprendizado supervisionado

Classificação vs. agrupamento

- Classificação: aprendizado supervisionado
- Agrupamento: aprendizado não supervisionado

Classificação vs. agrupamento

- Classificação: aprendizado supervisionado
- Agrupamento: aprendizado não supervisionado
- Classificação: classes são parte da entrada para o algoritmo de aprendizado

Classificação vs. agrupamento

- Classificação: aprendizado supervisionado
- Agrupamento: aprendizado não supervisionado
- Classificação: classes são parte da entrada para o algoritmo de aprendizado
- Agrupamento: grupos são inferidos a partir da proximidade de dados, sem indicação humana/externa

Classificação vs. agrupamento

- Classificação: aprendizado supervisionado
- Agrupamento: aprendizado não supervisionado
- Classificação: classes são parte da entrada para o algoritmo de aprendizado
- Agrupamento: grupos são inferidos a partir da proximidade de dados, sem indicação humana/externa
  - Maneira de influenciar a criação de agrupamento: número de grupos, medida de similaridade, tipo de representação, ...

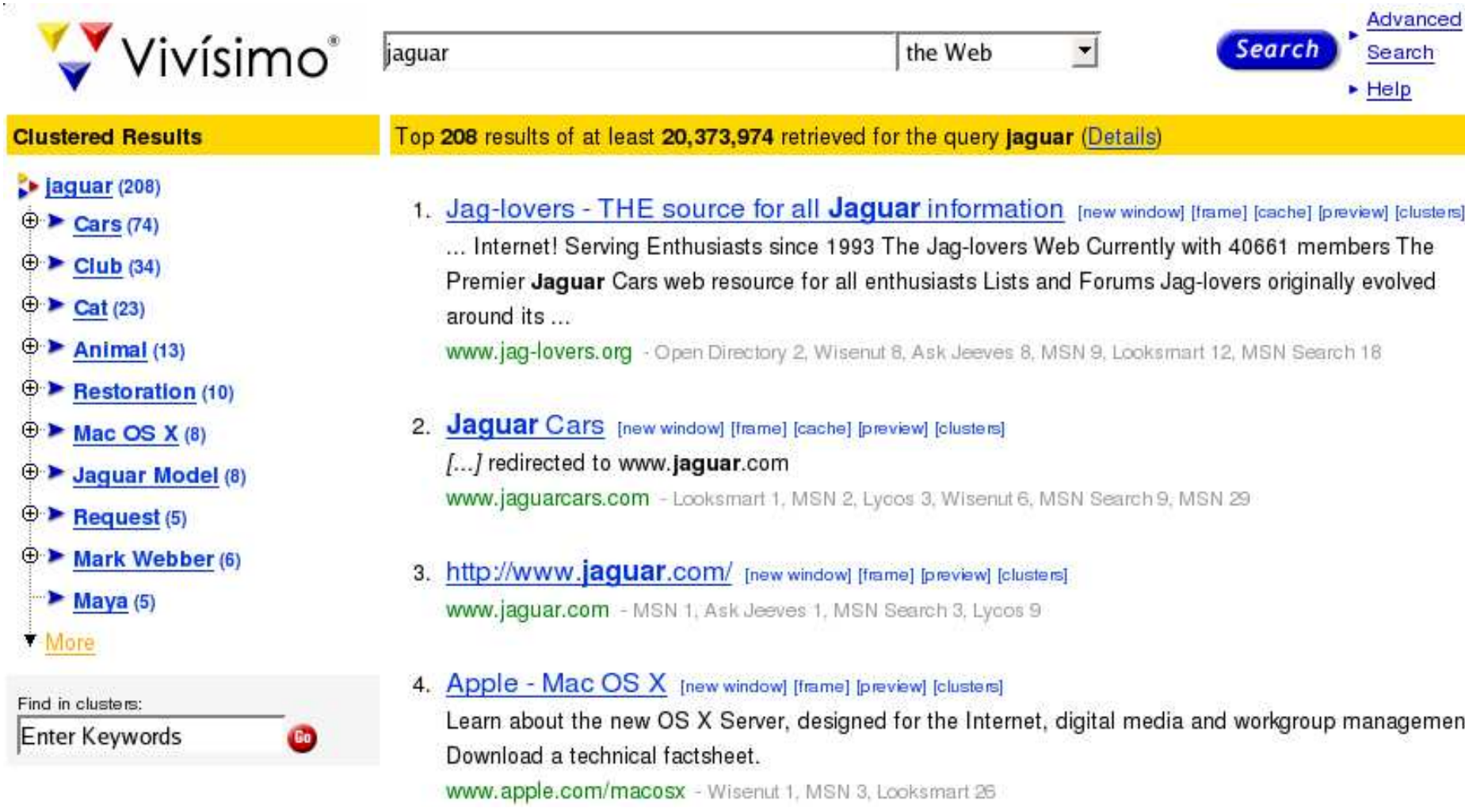
Hipótese de agrupamento

Elementos no mesmo grupo se comportam de maneira similar à relevância em relação à necessidade de informação

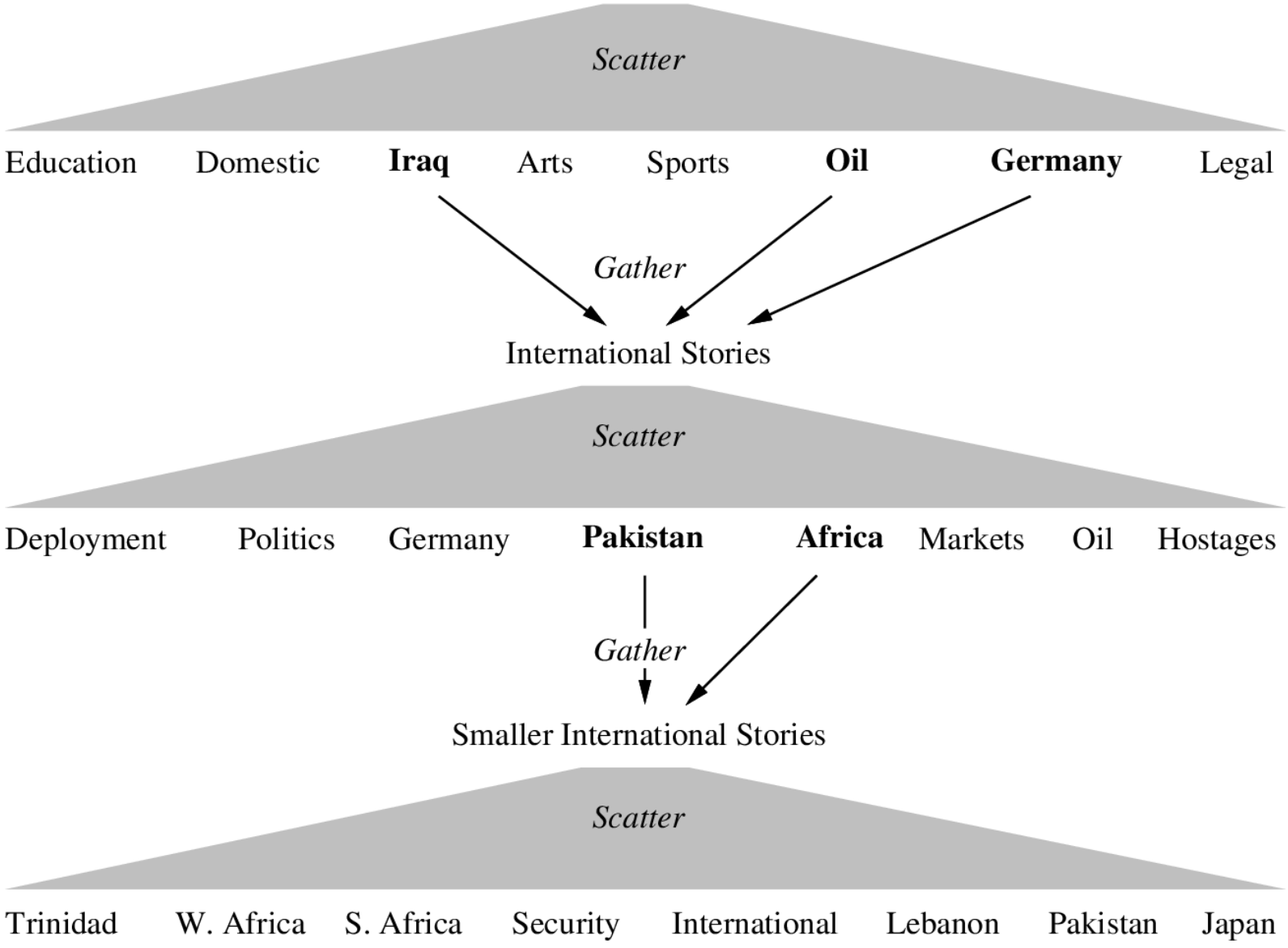
Aplicações de agrupamento

Aplicação	o que é agrupado?	benefício
agrupamento de resultados de busca	resultados de busca	informação mais efetiva apresentação para usuário
Scatter-Gather	(subconjuntos da) coleção	interface alternativa: "busca sem digitar "
Agrupamento de coleções	coleção	navegação exploratória
Recuperação baseada em agrupamento	coleção	melhor eficiência: busca mais rápida

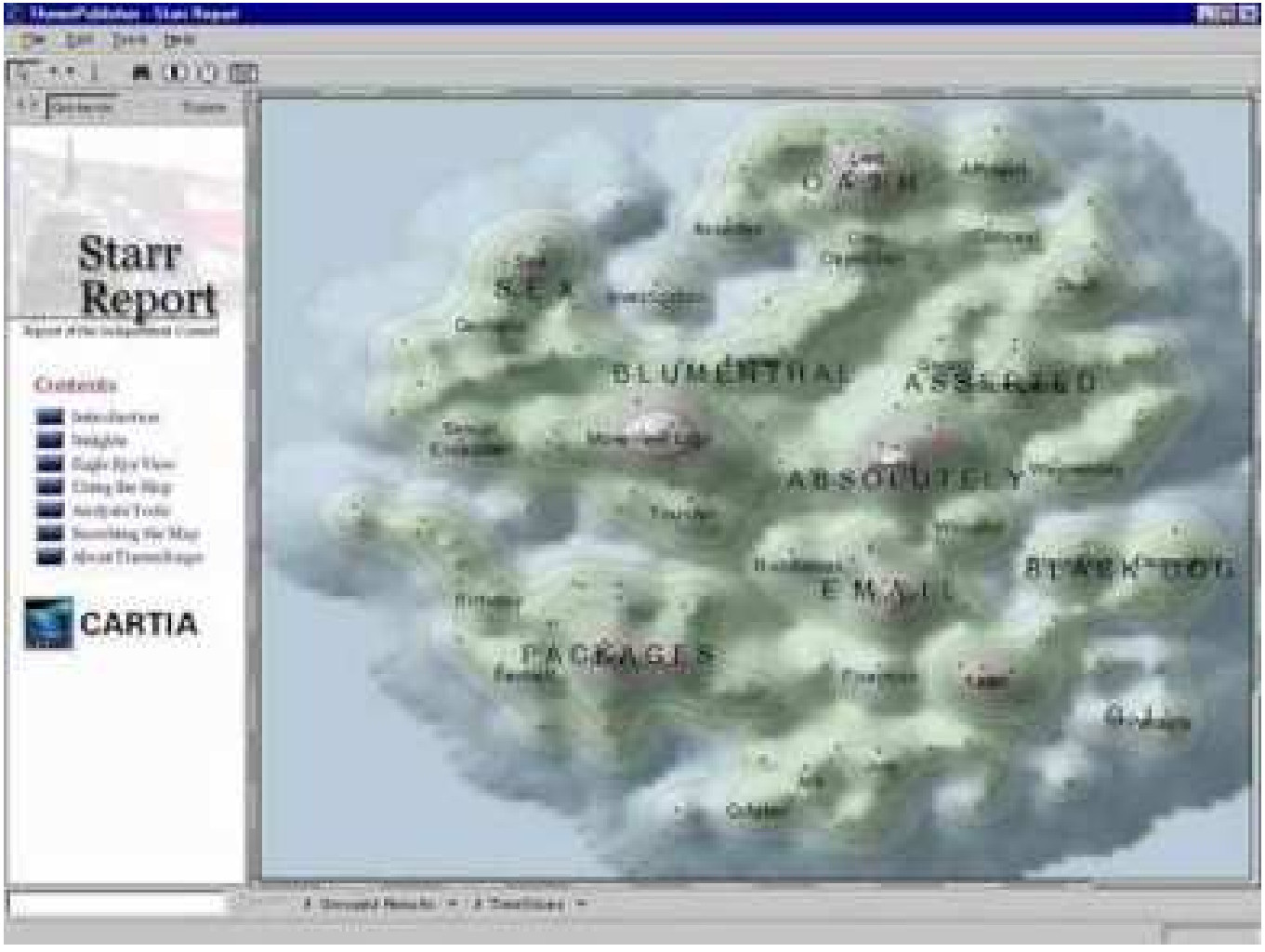
Agrupamento de resultados de busca para melhor navegação



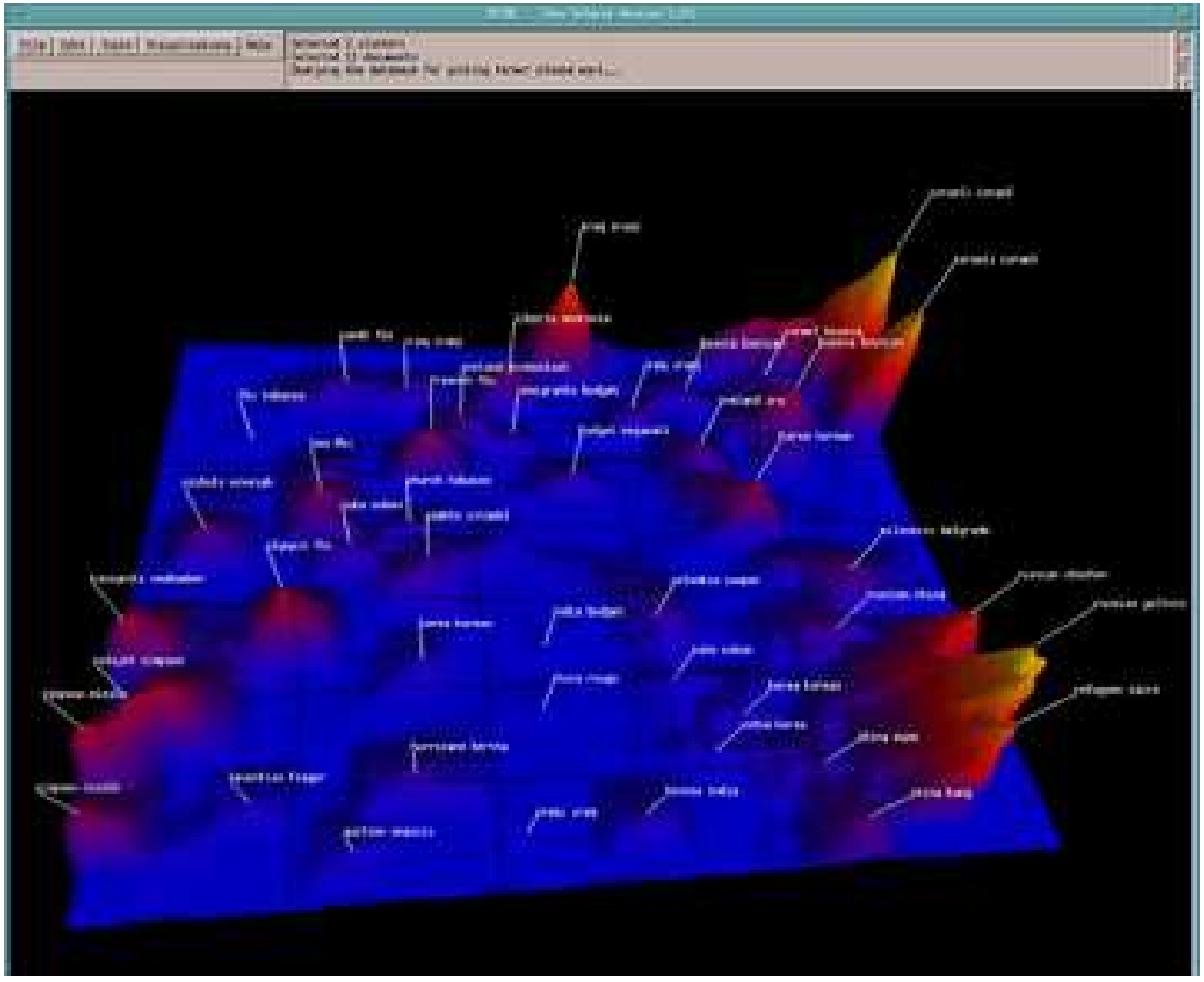
Scatter-Gather



Navegação global combinada com visualização (1)



Navegação global combinada com visualização (2)



Exemplo de aplicação: agrupamento para melhor taxa de recuperação de um buscador

- Agrupar documentos na coleção

Exemplo de aplicação: agrupamento para melhor taxa de recuperação de um buscador

- Agrupar documentos na coleção
- Quando uma consulta obtém um doc d, também retornar documentos no grupo com d

Exemplo de aplicação: agrupamento para melhor taxa de recuperação de um buscador

- Agrupar documentos na coleção
- Quando uma consulta obtém um doc d, também retornar documentos no grupo com d
  - Esperança que se fizermos isso, a consulta "carro" também apresentará "automóvel"

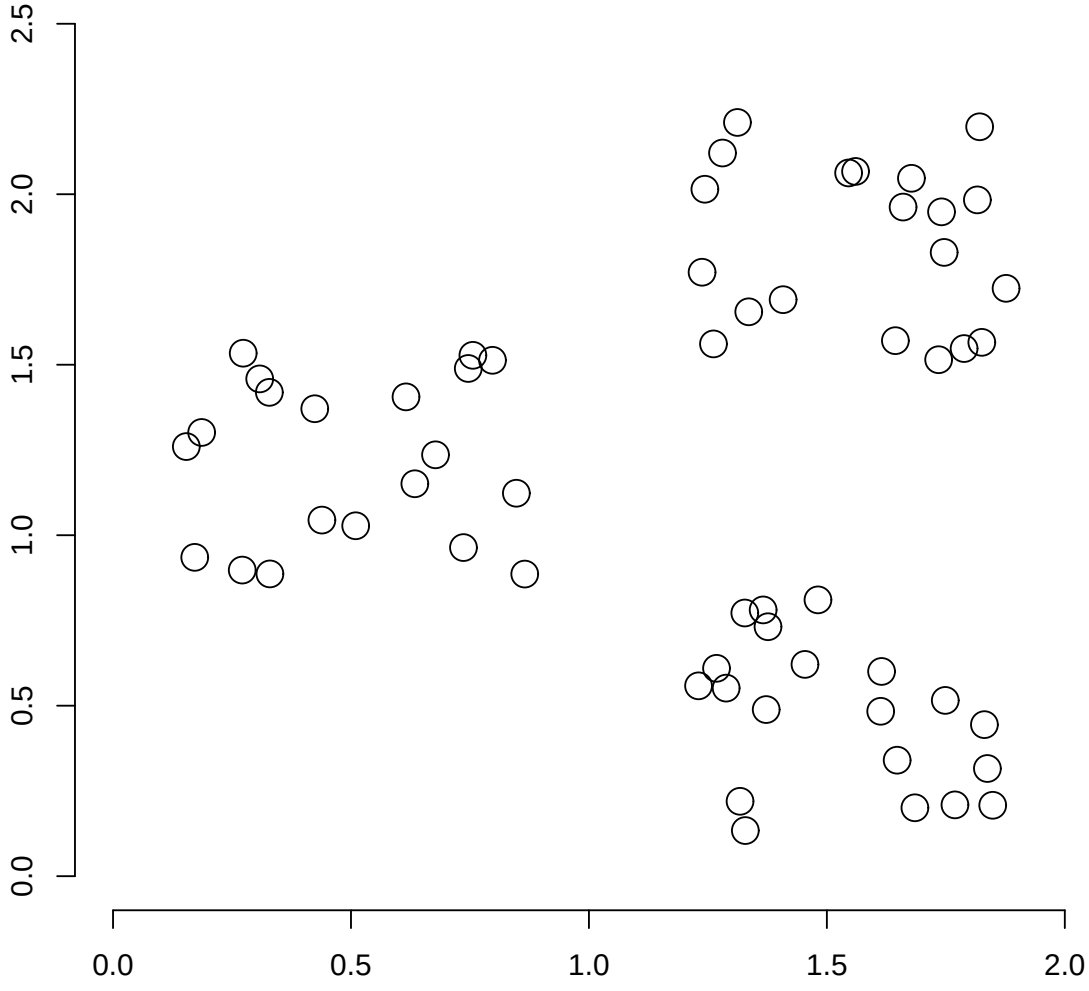
Exemplo de aplicação: agrupamento para melhor taxa de recuperação de um buscador

- Agrupar documentos na coleção
- Quando uma consulta obtém um doc d, também retornar documentos no grupo com d
  - Esperança que se fizermos isso, a consulta "carro" também apresentará "automóvel"
  - Porquê o algoritmo de agrupamento junta documentos contendo "carro" com aqueles contendo "automóvel"

Exemplo de aplicação: agrupamento para melhor taxa de recuperação de um buscador

- Agrupar documentos na coleção
- Quando uma consulta obtém um doc d, também retornar documentos no grupo com d
  - Esperança que se fizermos isso, a consulta "carro" também apresentará "automóvel"
  - Porquê o algoritmo de agrupamento junta documentos contendo "carro" com aqueles contendo "automóvel"
  - Ambos tipos de documentos contém palavras como "partes", "concessionária" e "mecânico"

Exercício: Algoritmo para agrupar dados



Propor algoritmo para encontrar a estrutura de agrupa-mento nessa figura

Metas de agrupamento

- Meta: colocar elementos relacionados no mesmo grupo, colocar elementos não relacionados em grupos diferentes.

Metas de agrupamento

- Meta: colocar elementos relacionados no mesmo grupo, colocar elementos não relacionados em grupos diferentes.
  - Diferentes formas de formalizar essa meta:

Metas de agrupamento

- Meta: colocar elementos relacionados no mesmo grupo, colocar elementos não relacionados em grupos diferentes.
  - Diferentes formas de formalizar essa meta:
- O número de grupos deve ser apropriado para o conjunto de dados que estamos agrupando

Metas de agrupamento

- Meta: colocar elementos relacionados no mesmo grupo, colocar elementos não relacionados em grupos diferentes.
  - Diferentes formas de formalizar essa meta:
- O número de grupos deve ser apropriado para o conjunto de dados que estamos agrupando
  - Inicialmente, assumiremos que o número de grupos k é conhecido

Metas de agrupamento

- Meta: colocar elementos relacionados no mesmo grupo, colocar elementos não relacionados em grupos diferentes.
  - Diferentes formas de formalizar essa meta:
- O número de grupos deve ser apropriado para o conjunto de dados que estamos agrupando
  - Inicialmente, assumiremos que o número de grupos k é conhecido
  - Depois, métodos semi-automáticos para determinar k

Metas de agrupamento

- Meta: colocar elementos relacionados no mesmo grupo, colocar elementos não relacionados em grupos diferentes.
  - Diferentes formas de formalizar essa meta:
- O número de grupos deve ser apropriado para o conjunto de dados que estamos agrupando
  - Inicialmente, assumiremos que o número de grupos k é conhecido
  - Depois, métodos semi-automáticos para determinar k
- Sub-metas em agrupamento

Metas de agrupamento

- Meta: colocar elementos relacionados no mesmo grupo, colocar elementos não relacionados em grupos diferentes.
  - Diferentes formas de formalizar essa meta:
- O número de grupos deve ser apropriado para o conjunto de dados que estamos agrupando
  - Inicialmente, assumiremos que o número de grupos k é conhecido
  - Depois, métodos semi-automáticos para determinar k
- Sub-metas em agrupamento
  - Evitar grupos muito pequenos e muito grandes

Metas de agrupamento

- Meta: colocar elementos relacionados no mesmo grupo, colocar elementos não relacionados em grupos diferentes.
  - Diferentes formas de formalizar essa meta:
- O número de grupos deve ser apropriado para o conjunto de dados que estamos agrupando
  - Inicialmente, assumiremos que o número de grupos k é conhecido
  - Depois, métodos semi-automáticos para determinar k
- Sub-metas em agrupamento
  - Evitar grupos muito pequenos e muito grandes
  - Definir grupos que são fáceis de explicar para o usuário

Metas de agrupamento

- Meta: colocar elementos relacionados no mesmo grupo, colocar elementos não relacionados em grupos diferentes.
  - Diferentes formas de formalizar essa meta:
- O número de grupos deve ser apropriado para o conjunto de dados que estamos agrupando
  - Inicialmente, assumiremos que o número de grupos k é conhecido
  - Depois, métodos semi-automáticos para determinar k
- Sub-metas em agrupamento
  - Evitar grupos muito pequenos e muito grandes
  - Definir grupos que são fáceis de explicar para o usuário
  - e muitos outros . . .

Metas de agrupamento

- Meta: colocar elementos relacionados no mesmo grupo, colocar elementos não relacionados em grupos diferentes.
  - Diferentes formas de formalizar essa meta:
- O número de grupos deve ser apropriado para o conjunto de dados que estamos agrupando
  - Inicialmente, assumiremos que o número de grupos k é conhecido
  - Depois, métodos semi-automáticos para determinar k
- Sub-metas em agrupamento
  - Evitar grupos muito pequenos e muito grandes
  - Definir grupos que são fáceis de explicar para o usuário
  - e muitos outros . . .
- Agrupamento particional vs. hierárquico

Metas de agrupamento

- Meta: colocar elementos relacionados no mesmo grupo, colocar elementos não relacionados em grupos diferentes.
  - Diferentes formas de formalizar essa meta:
- O número de grupos deve ser apropriado para o conjunto de dados que estamos agrupando
  - Inicialmente, assumiremos que o número de grupos k é conhecido
  - Depois, métodos semi-automáticos para determinar k
- Sub-metas em agrupamento
  - Evitar grupos muito pequenos e muito grandes
  - Definir grupos que são fáceis de explicar para o usuário
  - e muitos outros . . .
- Agrupamento particional vs. hierárquico
- Agrupamento "fechado" (*hard*) vs. "flexível" (*soft*)

Agrupamento particional vs. hierárquico

- Algoritmos particionais

Agrupamento particional vs. hierárquico

- Algoritmos particionais
  - Usualmente inicia com uma partição aleatória dos elementos em grupos

Agrupamento particional vs. hierárquico

- Algoritmos particionais
  - Usualmente inicia com uma partição aleatória dos elementos em grupos
  - Refinar/corrigir grupos iterativamente

Agrupamento particional vs. hierárquico

- Algoritmos particionais
  - Usualmente inicia com uma partição aleatória dos elementos em grupos
  - Refinar/corrigir grupos iterativamente
  - Algoritmo principal: k-médias

Agrupamento particional vs. hierárquico

- Algoritmos particionais
  - Usualmente inicia com uma partição aleatória dos elementos em grupos
  - Refinar/corrigir grupos iterativamente
  - Algoritmo principal: k-médias
- Algoritmos hierárquicos

Agrupamento particional vs. hierárquico

- Algoritmos particionais
  - Usualmente inicia com uma partição aleatória dos elementos em grupos
  - Refinar/corrigir grupos iterativamente
  - Algoritmo principal: k-médias
- Algoritmos hierárquicos
  - Criar uma hierarquia

Agrupamento particional vs. hierárquico

- Algoritmos particionais
  - Usualmente inicia com uma partição aleatória dos elementos em grupos
  - Refinar/corrigir grupos iterativamente
  - Algoritmo principal: k-médias
- Algoritmos hierárquicos
  - Criar uma hierarquia
  - De baixo para cima, aglomerativo (mais comum)

Agrupamento particional vs. hierárquico

- Algoritmos particionais
  - Usualmente inicia com uma partição aleatória dos elementos em grupos
  - Refinar/corrigir grupos iterativamente
  - Algoritmo principal: k-médias
- Algoritmos hierárquicos
  - Criar uma hierarquia
  - De baixo para cima, aglomerativo (mais comum)
  - De cima para baixo, divisivo

Agrupamento fechado (*hard*) vs. flexível (*soft*)

- Agrupamento fechado: cada documento pertence a exatamente um grupo

Agrupamento fechado (*hard*) vs. flexível (*soft*)

- Agrupamento fechado: cada documento pertence a exatamente um grupo
  - Mais comum e fácil de fazer

Agrupamento fechado (*hard*) vs. flexível (*soft*)

- Agrupamento fechado: cada documento pertence a exatamente um grupo
  - Mais comum e fácil de fazer
- Agrupamento flexível: um elemento pode pertencer a mais de um grupo

Agrupamento fechado (*hard*) vs. flexível (*soft*)

- Agrupamento fechado: cada documento pertence a exatamente um grupo
  - Mais comum e fácil de fazer
- Agrupamento flexível: um elemento pode pertencer a mais de um grupo
  - Faz mais sentido para aplicações que necessitam de hierarquias navegáveis

Agrupamento fechado (*hard*) vs. flexível (*soft*)

- Agrupamento fechado: cada documento pertence a exatamente um grupo
  - Mais comum e fácil de fazer
- Agrupamento flexível: um elemento pode pertencer a mais de um grupo
  - Faz mais sentido para aplicações que necessitam de hierarquias navegáveis
  - Exemplo: documentos sobre *tênis* podem ser de dois grupos:

Agrupamento fechado (*hard*) vs. flexível (*soft*)

- Agrupamento fechado: cada documento pertence a exatamente um grupo
  - Mais comum e fácil de fazer
- Agrupamento flexível: um elemento pode pertencer a mais de um grupo
  - Faz mais sentido para aplicações que necessitam de hierarquias navegáveis
  - Exemplo: documentos sobre *tênis* podem ser de dois grupos:
    - esportes

Agrupamento fechado (*hard*) vs. flexível (*soft*)

- Agrupamento fechado: cada documento pertence a exatamente um grupo
  - Mais comum e fácil de fazer
- Agrupamento flexível: um elemento pode pertencer a mais de um grupo
  - Faz mais sentido para aplicações que necessitam de hierarquias navegáveis
  - Exemplo: documentos sobre *tênis* podem ser de dois grupos:
    - esportes
    - sapatos

Agrupamento fechado (*hard*) vs. flexível (*soft*)

- Agrupamento fechado: cada documento pertence a exatamente um grupo
  - Mais comum e fácil de fazer
- Agrupamento flexível: um elemento pode pertencer a mais de um grupo
  - Faz mais sentido para aplicações que necessitam de hierarquias navegáveis
  - Exemplo: documentos sobre *tênis* podem ser de dois grupos:
    - esportes
    - sapatos
  - Só é possível com agrupamento flexível

Algoritmos particionais

- Algoritmos particionais obtém uma partição de N elementos em um conjunto de k grupos.

Algoritmos particionais

- Algoritmos particionais obtém uma partição de N elementos em um conjunto de k grupos.
  - Dados: um conjunto de elementos, uma maneira de compará-los e o número k:

Algoritmos particionais

- Algoritmos particionais obtém uma partição de N elementos em um conjunto de k grupos.
  - Dados: um conjunto de elementos, uma maneira de compará-los e o número k:
  - Encontrar: uma partição em k grupos que otimiza um dado critério de particionamento

Algoritmos particionais

- Algoritmos particionais obtém uma partição de N elementos em um conjunto de k grupos.
  - Dados: um conjunto de elementos, uma maneira de compará-los e o número k:
  - Encontrar: uma partição em k grupos que otimiza um dado critério de particionamento
- Força-bruta: enumerar exaustivamente partições, escolher a melhor de todas segundo o critério

Algoritmos particionais

- Algoritmos particionais obtém uma partição de N elementos em um conjunto de k grupos.
  - Dados: um conjunto de elementos, uma maneira de compará-los e o número k:
  - Encontrar: uma partição em k grupos que otimiza um dado critério de particionamento
- Força-bruta: enumerar exaustivamente partições, escolher a melhor de todas segundo o critério
- Métodos heurísticos: problema das k-médias

O problema das k-médias

- Agrupamento particional
- Entrada: um conjunto X e uma distância $d : X \times X \to R_+$
- Saída: um conjunto de centros $C = \{c_1, c_2, \ldots, c_k\}$ onde

$$\phi_C(x) = \arg\min_{c \in C} d(x, c)$$

- Então o objetivo do problema é encontrar C tal que

$$\min \sum_{x \in X} d(\phi_C(x), x)^2$$

Ideia de um algoritmo para o problema das k-médias

- Cada grupo é definido por um centroide:

Ideia de um algoritmo para o problema das k-médias

- Cada grupo é definido por um centroide:
- Um centroide:

$$\mu(c) = \frac{1}{|\omega(c)|} \sum_{x \in \omega(c)} x$$

onde usamos $\omega(c)$ para denotar um grupo de elementos que estão mais próximos de c

Ideia de um algoritmo para o problema das k-médias

- Cada grupo é definido por um centroide:
- Um centroide:

$$\mu(c) = \frac{1}{|\omega(c)|} \sum_{x \in \omega(c)} x$$

  onde usamos $\omega(c)$ para denotar um grupo de elementos que estão mais próximos de c
- Buscar pela diferença quadrática média mínima iterando:

Ideia de um algoritmo para o problema das k-médias

- Cada grupo é definido por um centroide:
- Um centroide:

$$\mu(c) = \frac{1}{|\omega(c)|} \sum_{x \in \omega(c)} x$$

  onde usamos $\omega(c)$ para denotar um grupo de elementos que estão mais próximos de c
- Buscar pela diferença quadrática média mínima iterando:
  - reatribuição: atribuir cada vetor ao seu centroide mais próximo

Ideia de um algoritmo para o problema das k-médias

- Cada grupo é definido por um centroide:
- Um centroide:

$$\mu(c) = \frac{1}{|\omega(c)|} \sum_{x \in \omega(c)} x$$

  onde usamos $\omega(c)$ para denotar um grupo de elementos que estão mais próximos de c
- Buscar pela diferença quadrática média mínima iterando:
  - reatribuição: atribuir cada vetor ao seu centroide mais próximo
  - recomputação: recomputar cada centroide como a média de vetores que foram escolhidos na reatribuição

Pseudo-código: algoritmo de Lloyd para agrupamento k-médias [10]

1. Escolher k elementos $C \subset X$
2. Repetir até C não mudar mais
   - 2.1 Para todo $x \in X$, achar $\phi_C(x)$ (centro $c \in C$ mais perto de x)
   - 2.2 Para todo $i \in \{1 \ldots k\}$, fazer $c_i = \text{média}\{x \in C | \phi_C(x) = c_i\}$

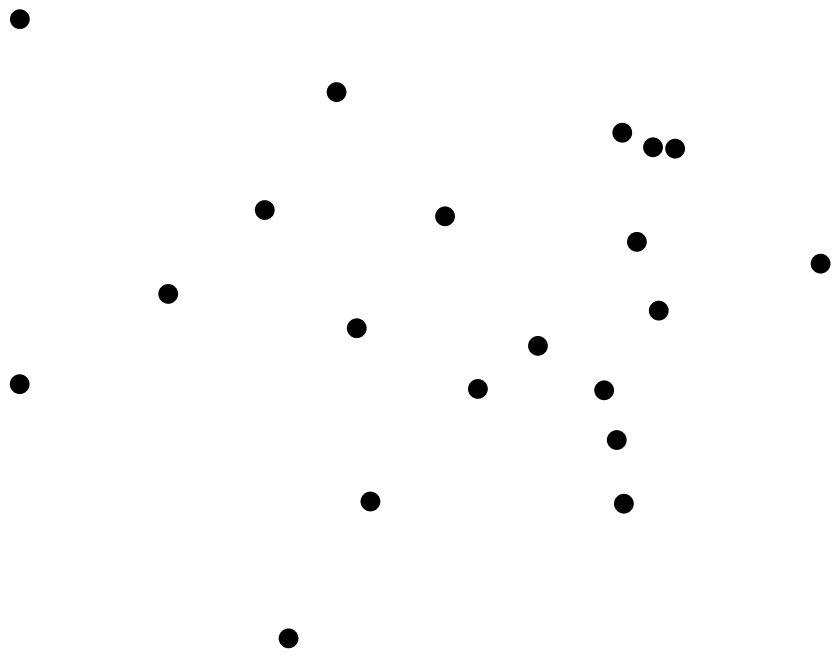
Pseudo-código: algoritmo de Lloyd para agrupamento k-médias [10]

1. Escolher k elementos $C \subset X$
2. Repetir até C não mudar mais
   - 2.1 Para todo $x \in X$, achar $\phi_C(x)$ (centro $c \in C$ mais perto de x)
   - 2.2 Para todo $i \in \{1 \ldots k\}$, fazer $c_i = \text{média}\{x \in C | \phi_C(x) = c_i\}$

- Custo naïve do algoritmo para R iterações, n elementos, d dimensões em x
  - Passo de atribuição: $\approx k \times n \times d$ computações entre elementos e centros
  - Passo de recomputação: $\approx n \times d$
- Total: $\approx R \times n \times d \times (k + 1)$

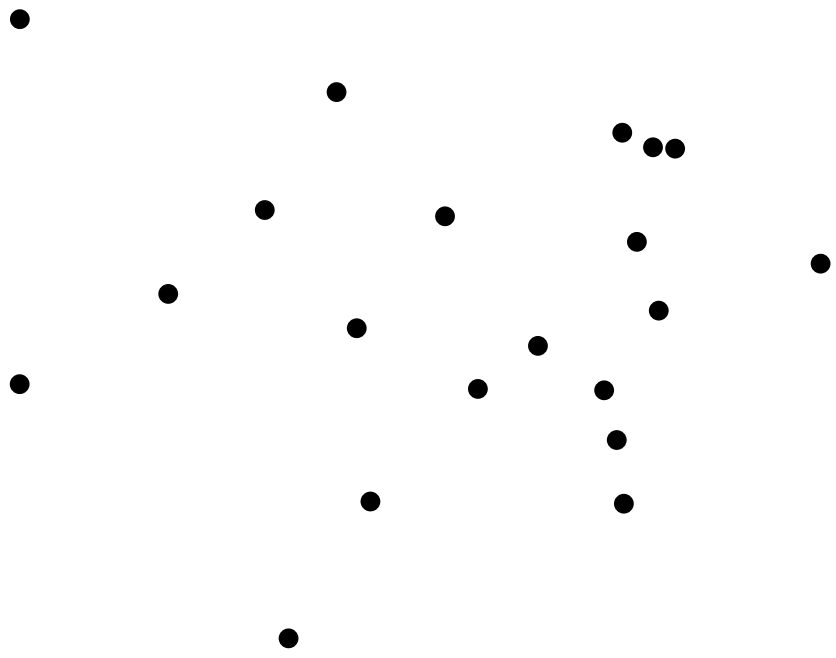
Exemplo: pontos para agrupamento

Cada ponto representa um documento.



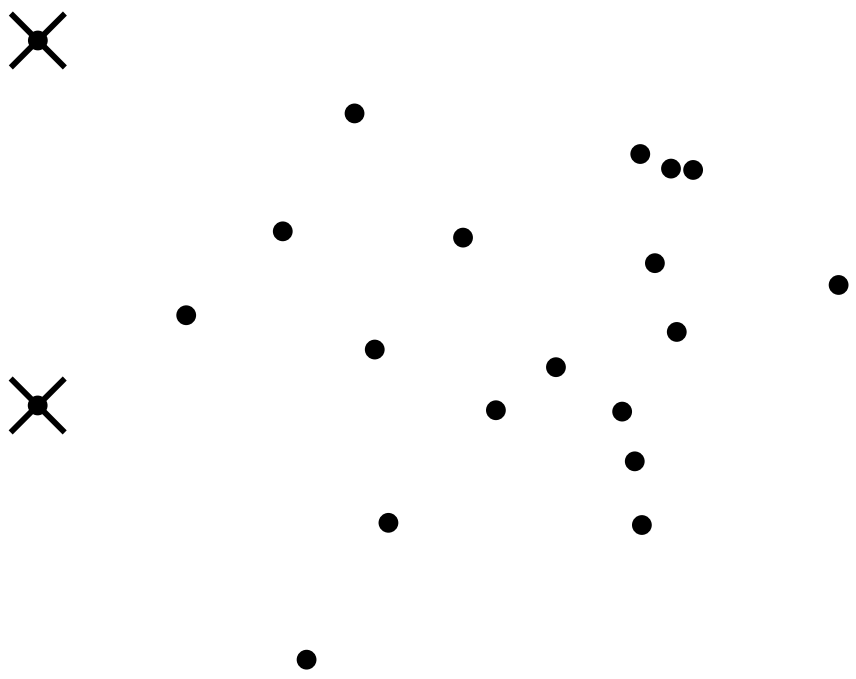
Exemplo

Cada ponto representa um documento.

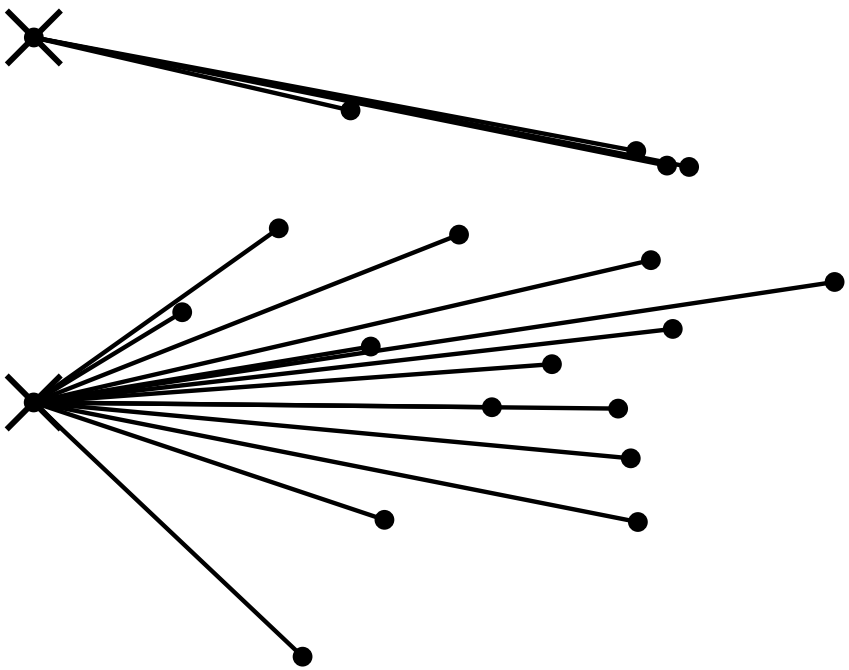


Exercício: (i) Adivinhar qual é o agrupamento ideal para dois grupos nesse caso ; (ii) computar os centroides dos grupos

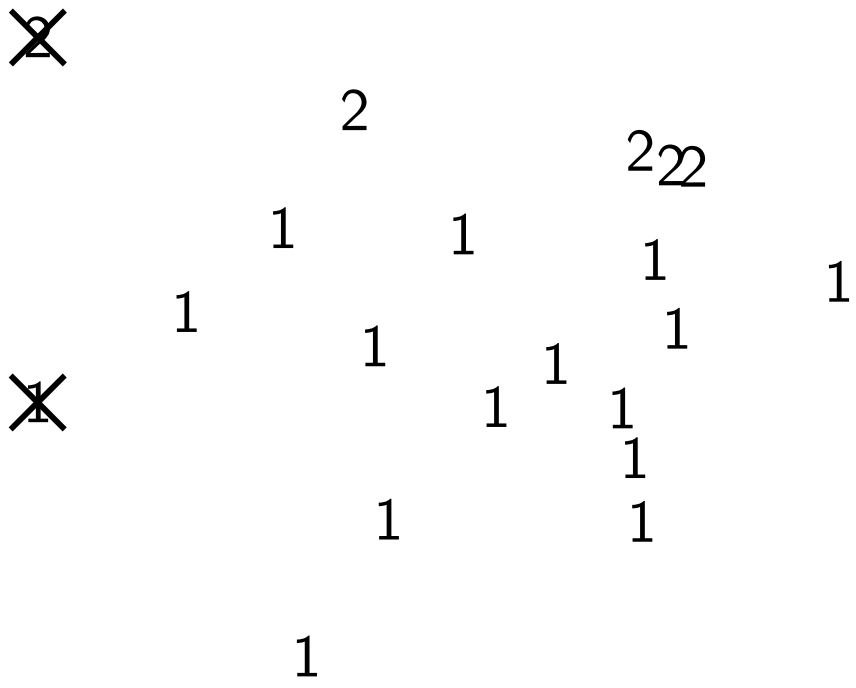
Exemplo: encontrar centroides iniciais aleatoriamente



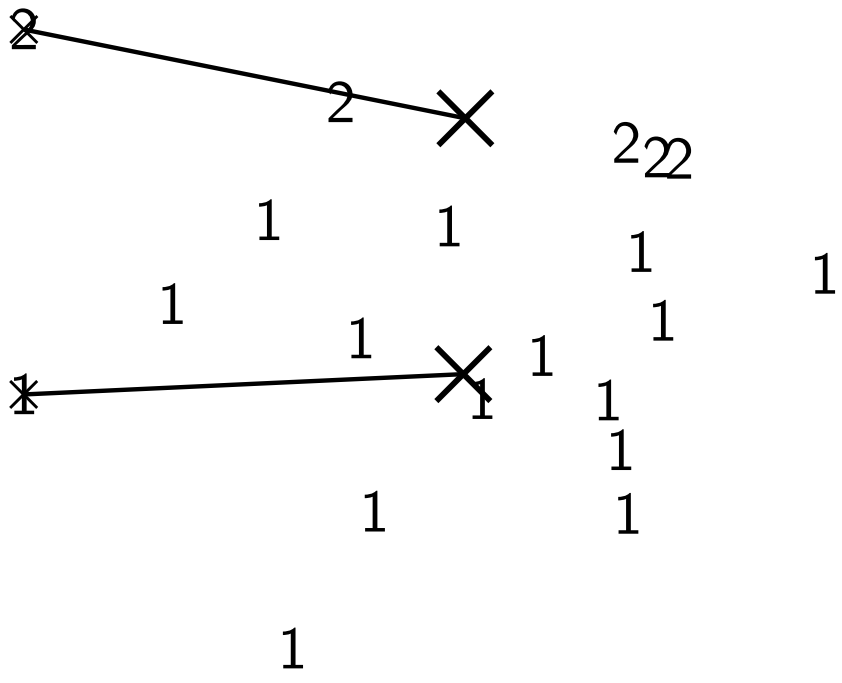
Exemplo: atribuir pontos para o centroide mais próximo



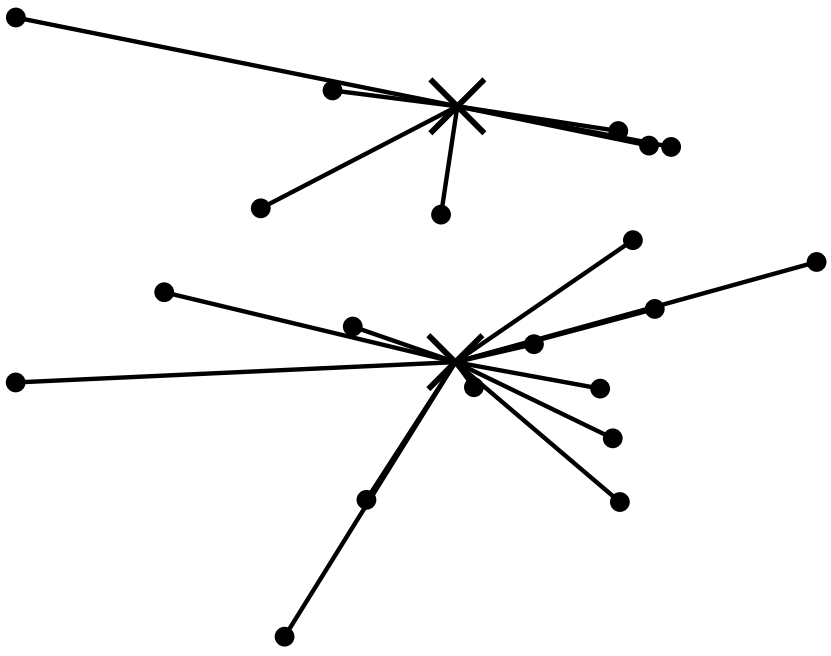
Exemplo: reatribuição



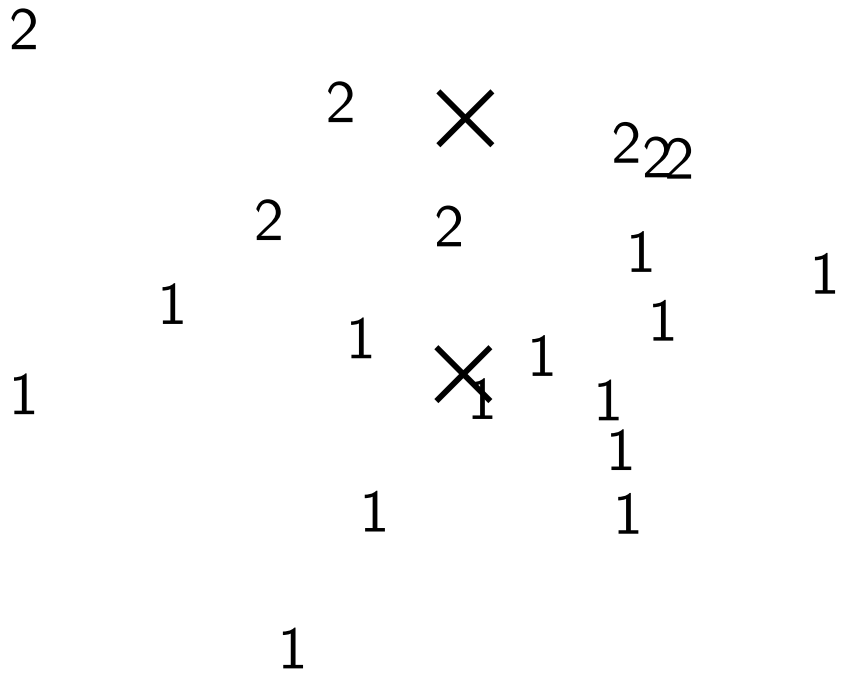
Exemplo: recalcular os centroides dos grupos



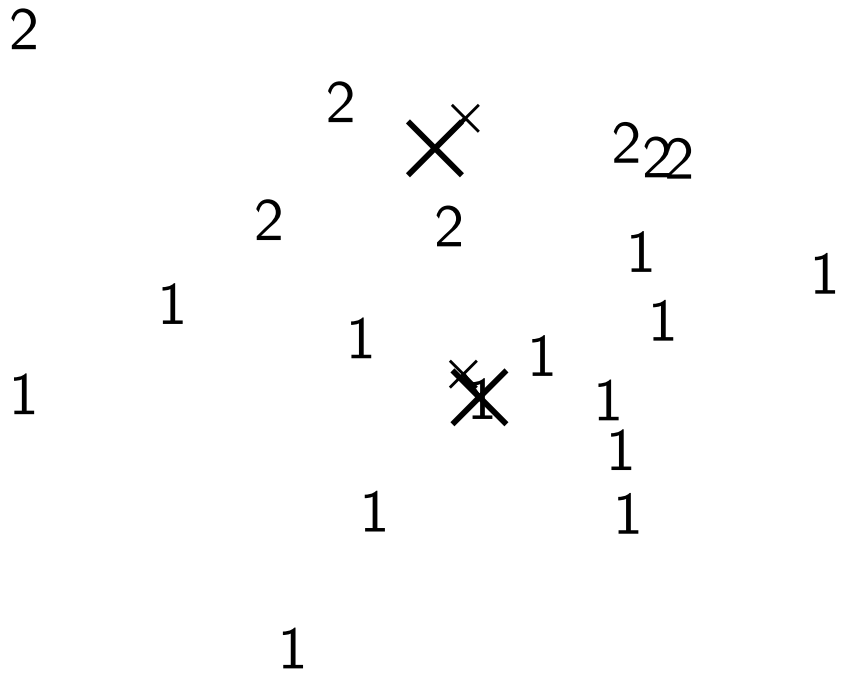
Exemplo: atribuir pontos para o centroide mais próximo



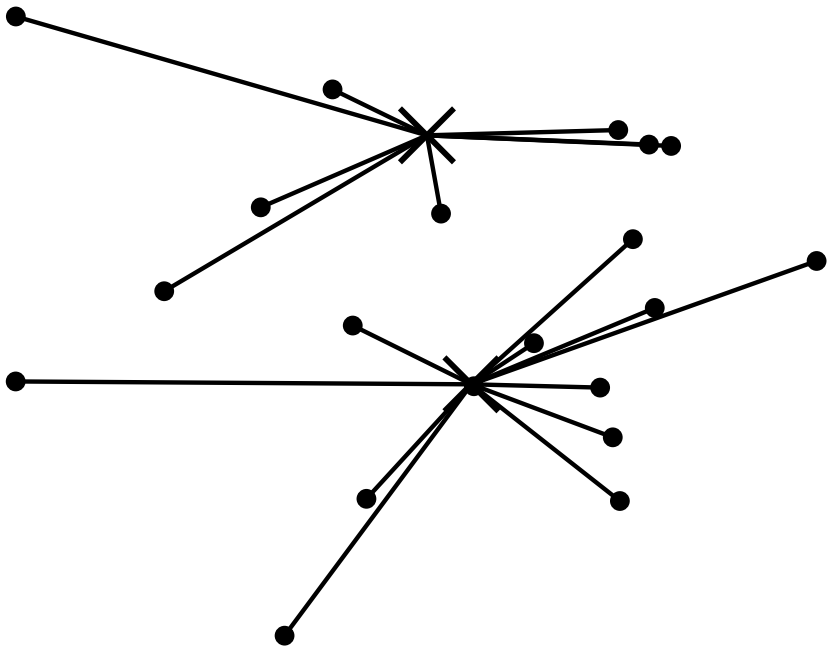
Exemplo: reatribuição



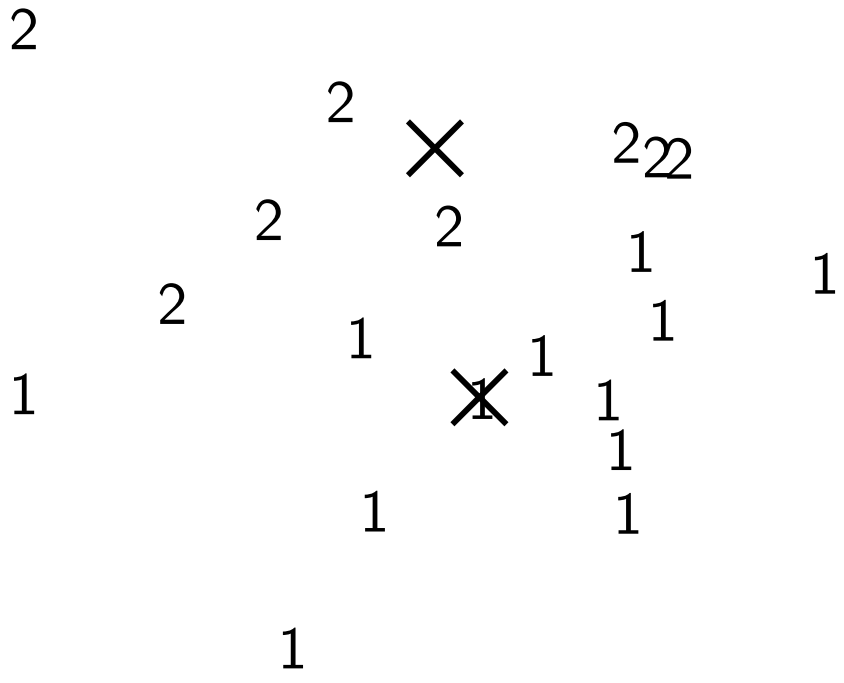
Exemplo: recalcular os centroides dos grupos



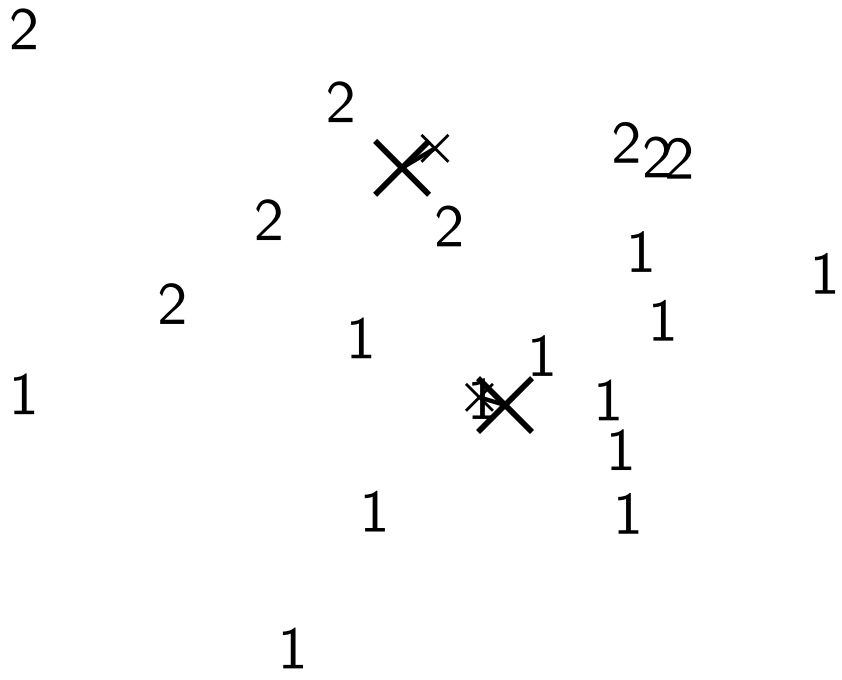
Exemplo: atribuir pontos para o centroide mais próximo



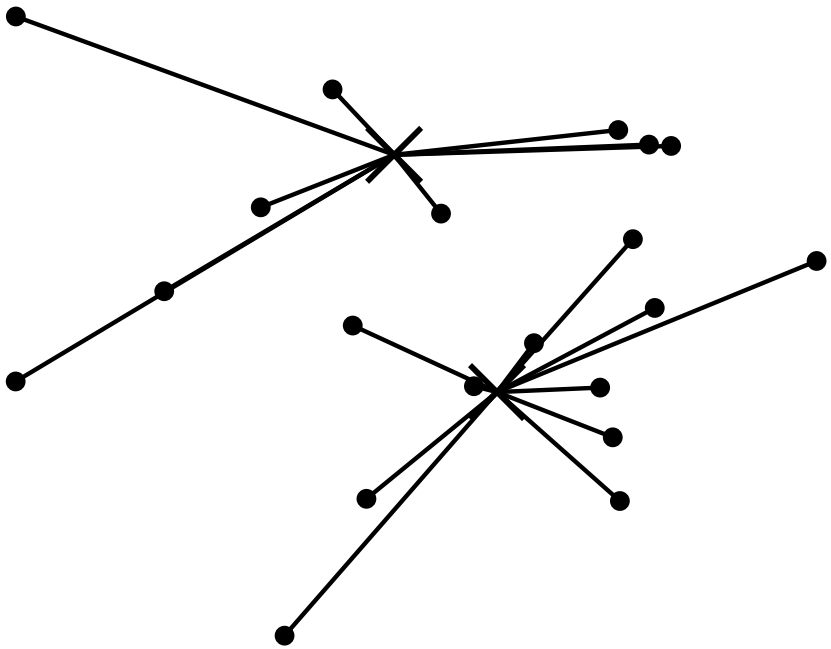
Exemplo: reatribuição



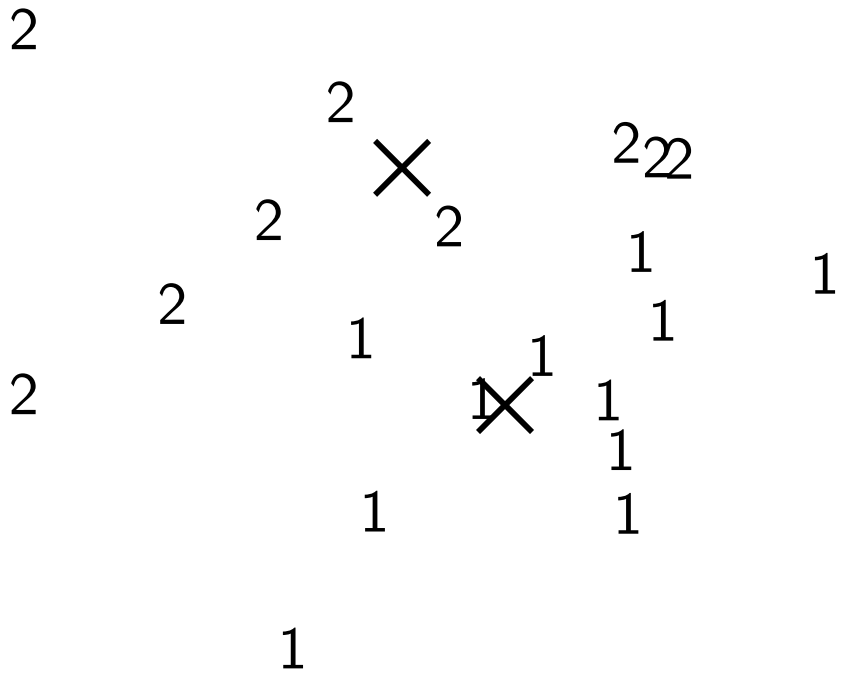
Exemplo: recalcular os centroides dos grupos



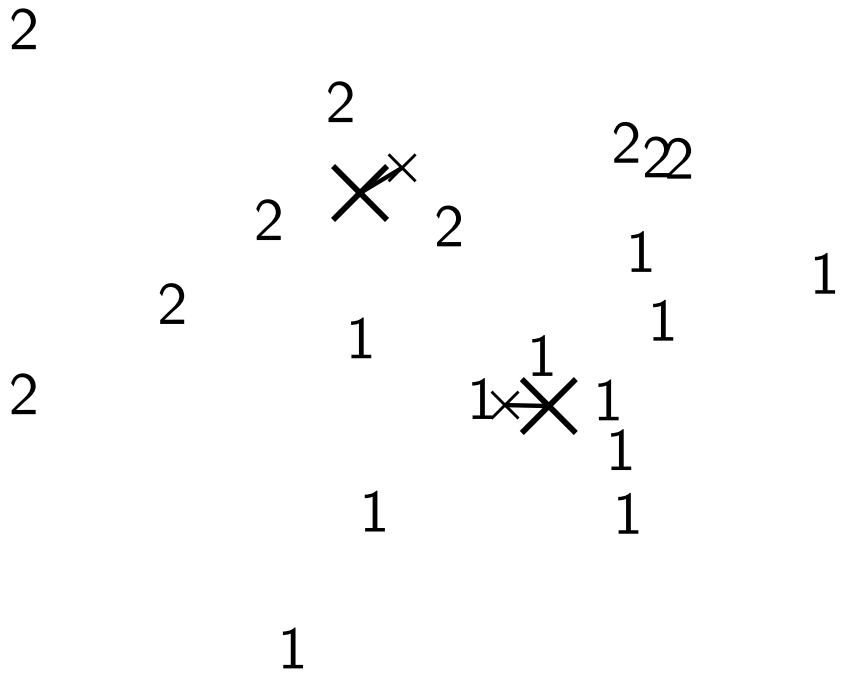
Exemplo: atribuir pontos para o centroide mais próximo



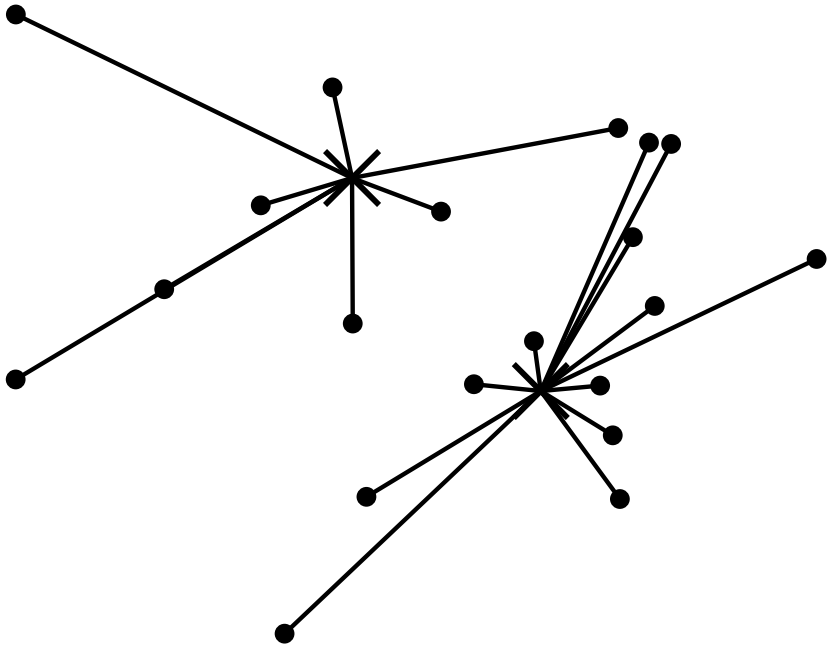
Exemplo: reatribuição



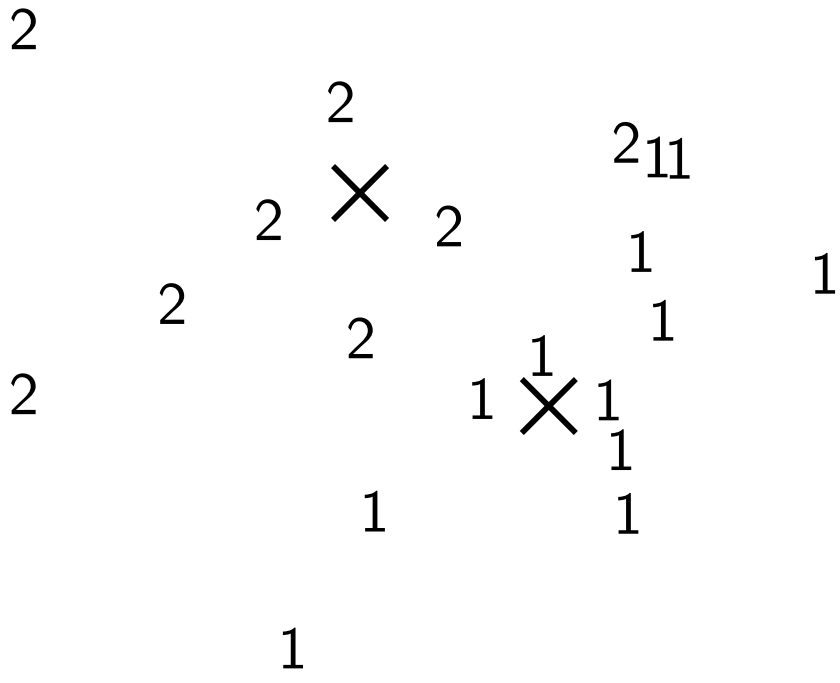
Exemplo: recalcular os centroides dos grupos



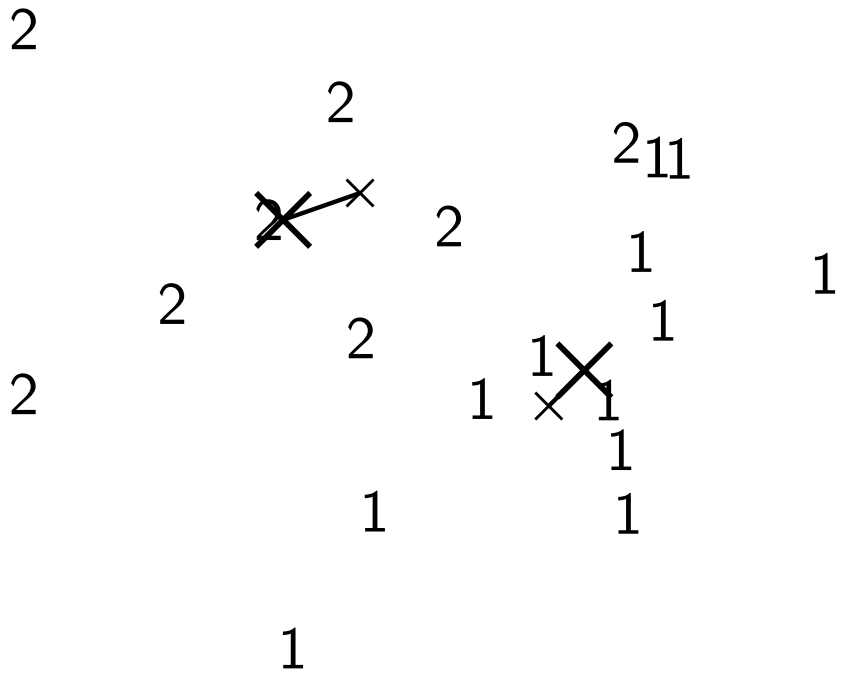
Exemplo: atribuir pontos para o centroide mais próximo



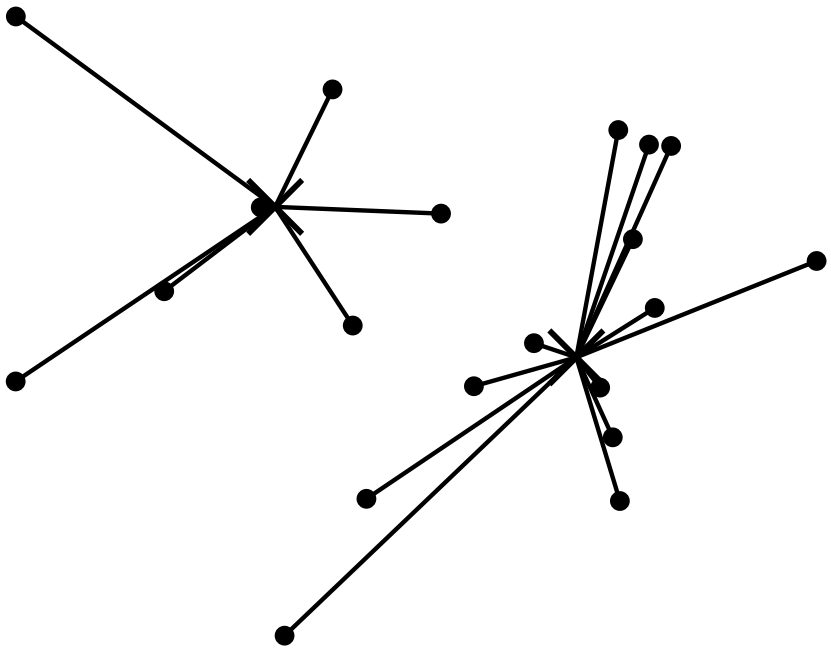
Exemplo: reatribuição



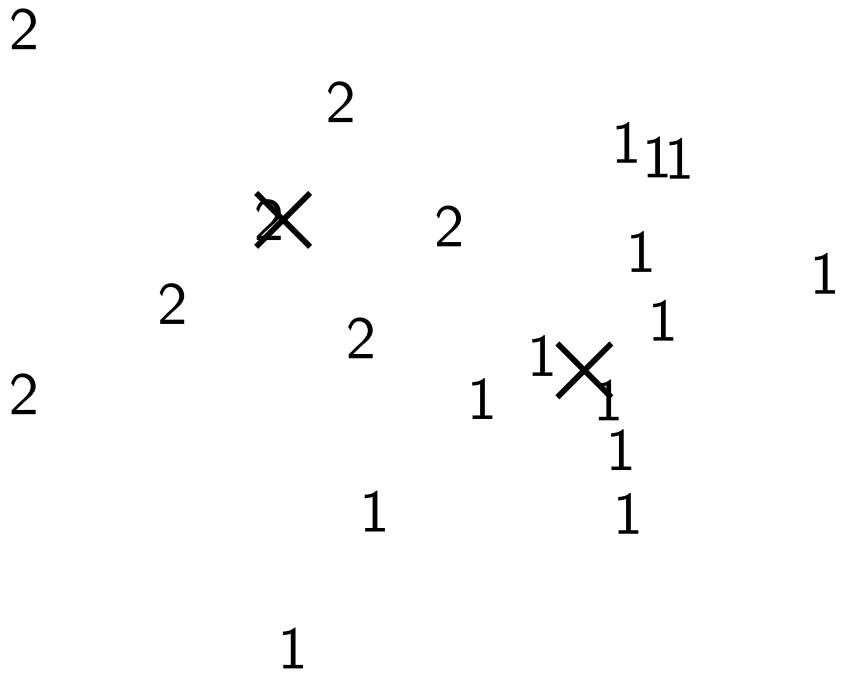
Exemplo: recalcular os centroides dos grupos



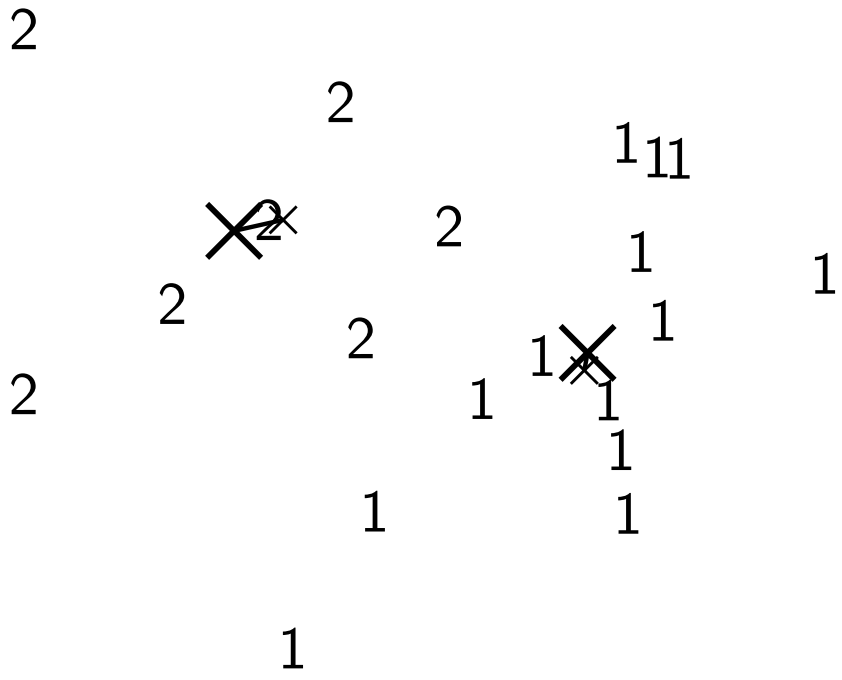
Exemplo: atribuir pontos para o centroide mais próximo



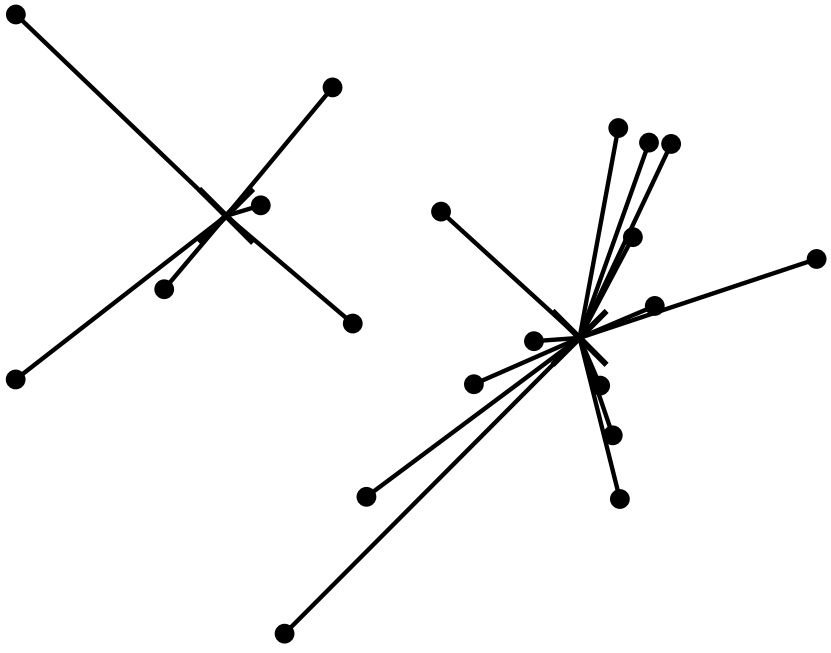
Exemplo: reatribuição



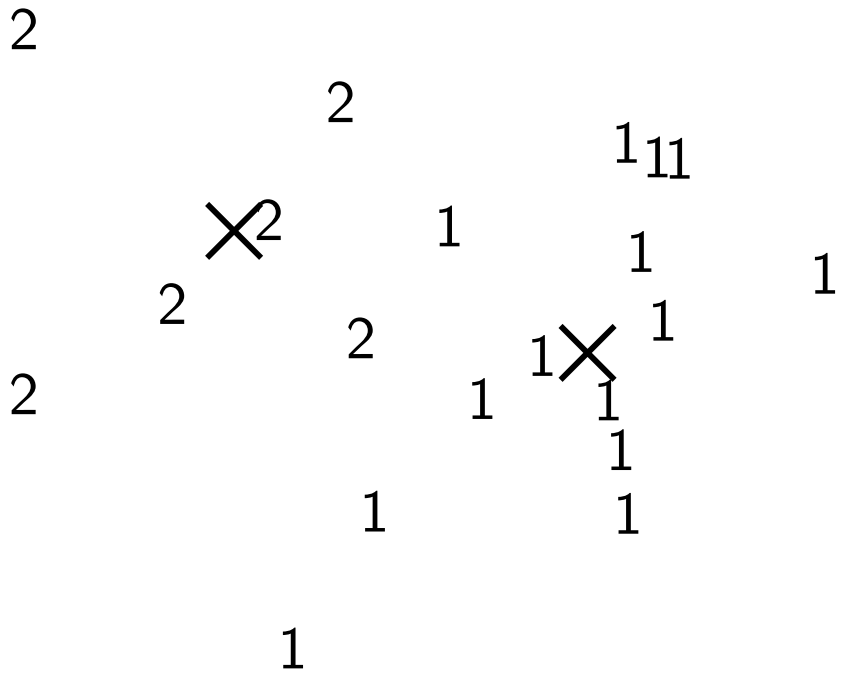
Exemplo: recalcular os centroides dos grupos



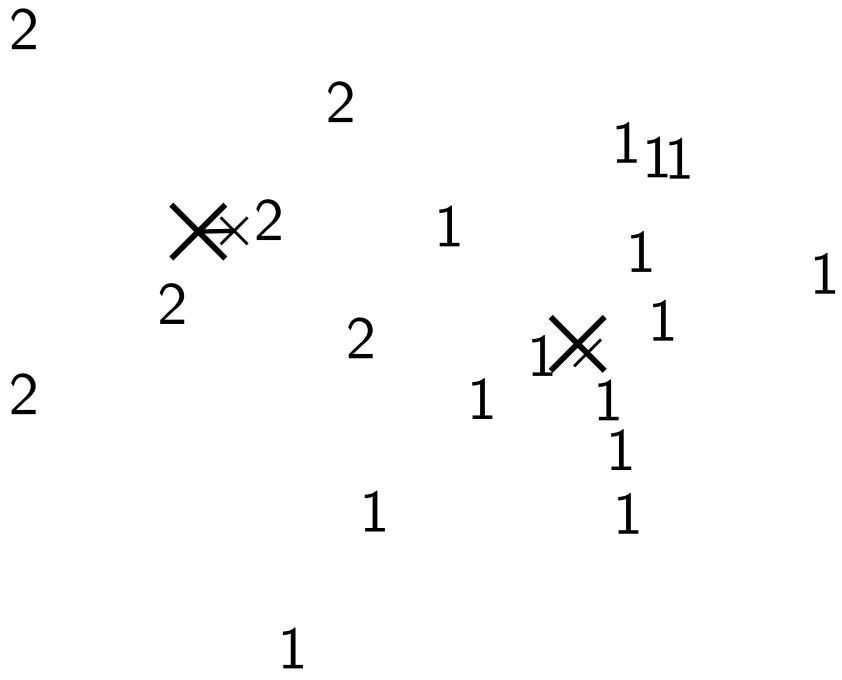
Exemplo: atribuir pontos para o centroide mais próximo



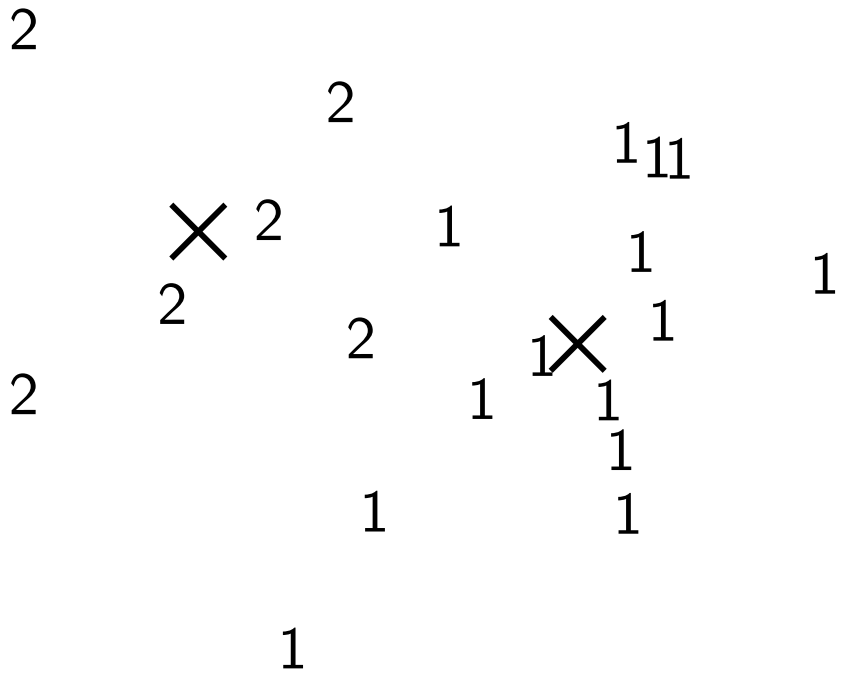
Exemplo: reatribuição



Exemplo: recalcular os centroides dos grupos



Exemplo: centroides e atribuições após convergência



Convergência do k-médias é garantida: ideia da prova

- RSS = *Residual Sum of Squares*

Convergência do k-médias é garantida: ideia da prova

- RSS = *Residual Sum of Squares*
- RSS = soma das distâncias quadráticas entre documentos e seus centroides

Convergência do k-médias é garantida: ideia da prova

- RSS = *Residual Sum of Squares*
- RSS = soma das distâncias quadráticas entre documentos e seus centroides
- RSS monotonicamente reduz em cada reatribuição e recálculo

Convergência do k-médias é garantida: ideia da prova

- RSS = *Residual Sum of Squares*
- RSS = soma das distâncias quadráticas entre documentos e seus centroides
- RSS monotonicamente reduz em cada reatribuição e recálculo
  - porque cada vetor é atribuído para o centroide mais perto

Convergência do k-médias é garantida: ideia da prova

- RSS = *Residual Sum of Squares*
- RSS = soma das distâncias quadráticas entre documentos e seus centroides
- RSS monotonicamente reduz em cada reatribuição e recálculo
  - porque cada vetor é atribuído para o centroide mais perto
- Chega-se a um ponto fixo onde centroides não mudam mais

Convergência do k-médias é garantida: ideia da prova

- RSS = *Residual Sum of Squares*
- RSS = soma das distâncias quadráticas entre documentos e seus centroides
- RSS monotonicamente reduz em cada reatribuição e recálculo
  - porque cada vetor é atribuído para o centroide mais perto
- Chega-se a um ponto fixo onde centroides não mudam mais
- Conjunto finito & RSS decrescente $\Rightarrow$ convergência

Recomputação decresce a distância média

$\mathrm{RSS} = \sum_{k=1}^{K} \mathrm{RSS}_k$ – soma residual de distâncias quadradas (medida de adequação)

$$\begin{aligned} \mathrm{RSS}_k(\vec{v}) &= \sum_{\vec{x} \in \omega_k} \|\vec{v} - \vec{x}\|^2 = \sum_{\vec{x} \in \omega_k} \sum_{m=1}^{M} (v_m - x_m)^2 \\ \frac{\partial \mathrm{RSS}_k(\vec{v})}{\partial v_m} &= \sum_{\vec{x} \in \omega_k} 2(v_m - x_m) = 0 \end{aligned}$$

$$v_m = \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} x_m$$

Recomputação decresce a distância média

$\text{RSS} = \sum_{k=1}^{K} \text{RSS}_k$ – soma residual de distâncias quadradas (medida de adequação)

$$\begin{aligned} \text{RSS}_k(\vec{v}) &= \sum_{\vec{x} \in \omega_k} \|\vec{v} - \vec{x}\|^2 = \sum_{\vec{x} \in \omega_k} \sum_{m=1}^{M} (v_m - x_m)^2 \\ \frac{\partial \text{RSS}_k(\vec{v})}{\partial v_m} &= \sum_{\vec{x} \in \omega_k} 2(v_m - x_m) = 0 \end{aligned}$$

$$v_m = \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} x_m$$

A última linha é a definição do centroide por componente.

Recomputação decresce a distância média

$\text{RSS} = \sum_{k=1}^{K} \text{RSS}_k$ – soma residual de distâncias quadradas (medida de adequação)

$$\begin{aligned} \text{RSS}_k(\vec{v}) &= \sum_{\vec{x} \in \omega_k} \|\vec{v} - \vec{x}\|^2 = \sum_{\vec{x} \in \omega_k} \sum_{m=1}^{M} (v_m - x_m)^2 \\ \frac{\partial \text{RSS}_k(\vec{v})}{\partial v_m} &= \sum_{\vec{x} \in \omega_k} 2(v_m - x_m) = 0 \end{aligned}$$

$$v_m = \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} x_m$$

A última linha é a definição do centroide por componente. Minimizamos RSS_k quando o centroide anterior é substituído com um novo centroide.

Recomputação decresce a distância média

$\text{RSS} = \sum_{k=1}^{K} \text{RSS}_k$ – soma residual de distâncias quadradas (medida de adequação)

$$\begin{aligned}
\text{RSS}_k(\vec{v}) &= \sum_{\vec{x} \in \omega_k} \|\vec{v} - \vec{x}\|^2 = \sum_{\vec{x} \in \omega_k} \sum_{m=1}^{M} (v_m - x_m)^2 \\
\frac{\partial \text{RSS}_k(\vec{v})}{\partial v_m} &= \sum_{\vec{x} \in \omega_k} 2(v_m - x_m) = 0
\end{aligned}$$

$$v_m = \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} x_m$$

A última linha é a definição do centroide por componente. Minimizamos RSS_k quando o centroide anterior é substituído com um novo centroide. RSS, a soma de RSS_k, precisa então também reduzir durante os recálculos. M é o número de características que compõem os exemplos.

k-médias converge

- Mas não sabemos quanto tempo levará

k-médias converge

- Mas não sabemos quanto tempo levará
- Se não nos importamos com alguns elementos trocando de grupo repetidamente então, convergência será rápida ($<$ 10-20 iterações).

k-médias converge

- Mas não sabemos quanto tempo levará
- Se não nos importamos com alguns elementos trocando de grupo repetidamente então, convergência será rápida ($<$ 10-20 iterações).
- Porém, convergência completa pode levar mais iterações

Otimalidade do k-médias

- Convergência $\neq$ otimalidade (fraqueza do k-médias)

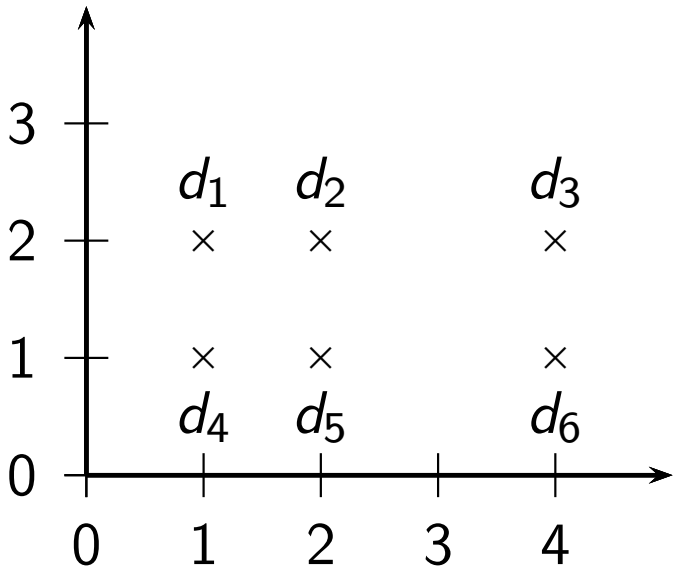
Otimalidade do k-médias

- Convergência $\neq$ otimalidade (fraqueza do k-médias)
- Convergência não significa encontrar o agrupamento ótimo

Otimalidade do k-médias

- Convergência $\neq$ otimalidade (fraqueza do k-médias)
- Convergência não significa encontrar o agrupamento ótimo
- Início com centroides ruins $\Rightarrow$ resultado ruim

Exercício: agrupamento subótimo



- Qual é o agrupamento ótimo para $K = 2$?
- Convergiremos nesse agrupamento para quaisquer centroides iniciais aleatórios d_i, d_j?

Inicialização de k-médias

- Comum: inicialização aleatória $\Rightarrow$ agrupamentos subótimos

Inicialização de k-médias

- Comum: inicialização aleatória $\Rightarrow$ agrupamentos subótimos
- Outras:

Inicialização de k-médias

- Comum: inicialização aleatória $\Rightarrow$ agrupamentos subótimos
- Outras:
  1. Não escolher elementos espúrios

Inicialização de k-médias

- Comum: inicialização aleatória $\Rightarrow$ agrupamentos subótimos
- Outras:
  1. Não escolher elementos espúrios
  2. Encontrar centroides bem espalhados no espaço

Inicialização de k-médias

- Comum: inicialização aleatória $\Rightarrow$ agrupamentos subótimos
- Outras:
  1. Não escolher elementos espúrios
  2. Encontrar centroides bem espalhados no espaço
  3. Usar vários execuções com conjuntos iniciais distintos, ficar com agrupamento de menor RSS

Complexidade de tempo do algoritmo naïve para k-médias

- Calcular uma distância entre dois vetores é $O(d)$, d sendo número de dimensões.

Complexidade de tempo do algoritmo naïve para k-médias

- Calcular uma distância entre dois vetores é $O(d)$, d sendo número de dimensões.
- Passo de reatribuição: $O(KNd)$ no calculo de KN distâncias centroide-elementos de K grupos

Complexidade de tempo do algoritmo naïve para k-médias

- Calcular uma distância entre dois vetores é $O(d)$, d sendo número de dimensões.
- Passo de reatribuição: $O(KNd)$ no calculo de KN distâncias centroide-elementos de K grupos
- Passo de recomputação: $O(Nd)$ ao somar cada valor das d dimensões do elemento ao seu centroide

Complexidade de tempo do algoritmo naïve para k-médias

- Calcular uma distância entre dois vetores é $O(d)$, d sendo número de dimensões.
- Passo de reatribuição: $O(KNd)$ no calculo de KN distâncias centroide-elementos de K grupos
- Passo de recomputação: $O(Nd)$ ao somar cada valor das d dimensões do elemento ao seu centroide
- Número de iterações é denotado por I desconhecido

Complexidade de tempo do algoritmo naïve para k-médias

- Calcular uma distância entre dois vetores é $O(d)$, d sendo número de dimensões.
- Passo de reatribuição: $O(KNd)$ no calculo de KN distâncias centroide-elementos de K grupos
- Passo de recomputação: $O(Nd)$ ao somar cada valor das d dimensões do elemento ao seu centroide
- Número de iterações é denotado por I desconhecido
- Complexidade geral: $O(IKNd)$

Complexidade de tempo do algoritmo naïve para k-médias

- Calcular uma distância entre dois vetores é $O(d)$, d sendo número de dimensões.
- Passo de reatribuição: $O(KNd)$ no calculo de KN distâncias centroide-elementos de K grupos
- Passo de recomputação: $O(Nd)$ ao somar cada valor das d dimensões do elemento ao seu centroide
- Número de iterações é denotado por I desconhecido
- Complexidade geral: $O(IKNd)$
- Casos patológicos $I \sim f(K)$ e complexidade fica pior que linear

Complexidade de tempo do k-médias

- Número de iterações é super-polinomial $O(N^{35}K^{34}d^8)$ [2]
- Se pontos estão em uma grade de tamanho M, então iterações é $O(dn^4M^2)$ [2]
- Usualmente poucas iterações são necessárias
- Alguns casos podem levar muitas iterações
- Execução de várias instâncias do k-médias ajuda encontrar bom agrupamento

Acelerando k-médias

- Usar propriedades da desigualdade triangular para evitar computações desnecessárias [5]
- Rodar inicialmente com amostra bem menor que os dados totais para obter estimativa inicial boa [4, 8]
- Rodar um algoritmo de uma-passada (fluxo de dados) para $k \log k$ grupos e depois mesclar para k [8, 1]

Yinyang *k*-means

- Explora propriedades da desigualdade triangular
- Otimizações nas fases do *k*-means
  - Etapa de atribuição: detectar quais distâncias são desnecessárias para calcular
    - Filtros de distâncias para evitar calcular distâncias
    - Filtros são usados globalmente ou em conjuntos de grupos
  - Etapa de atualização: somente atualizar os centroides removendo os pontos que sairam e adicionando os pontos que chegaram

Desigualdade triangular

Formulação original:

$$d(a, c) \leq d(a, b) + d(b, c)$$

Desigualdade triangular reversa:

$$d(a, c) \geq |d(a, b) - d(b, c)|$$

Então, sendo b e c dois centros de grupos e x um ponto qualquer:

$$|d(x, b) - d(b, c)| \leq d(x, c) \leq d(x, b) + d(b, c)$$

Prova da desigualdade triangular reversa

Usando que

$$\|y - x\| = \| - 1(x - y)\| = | - 1|\|x - y\| = \|x - y\|$$

Sendo fato (1):

$$\|x\| = \|(x - y) + y\| \leq \|x - y\| + \|y\| \Rightarrow \|x\| - \|y\| \leq \|x - y\|$$

Sendo fato (2):

$$\|y\| = \|(y - x) + x\| \leq \|y - x\| + \|x\| \Rightarrow \|x\| - \|y\| \geq -\|x - y\|$$

Combinando fato (1) e fato (2):

$$-\|x - y\| \leq \|x\| - \|y\| \leq \|x - y\| \Rightarrow |\|x\| - \|y\|| \leq \|x - y\|$$

Condição para filtragem (global)

- C é conjunto de grupos e $b(x)$ é o centro do grupo de x
- $\delta(c) = d(c, c')$ é a modificação do centro c para c'
- Para todo ponto x, $lb(x)$ é um limiar inferior se $lb(x) \leq d(x, c), \forall c \in C - b(x)$
  - Nenhuma distância de x para um centro é menor que $lb(x)$ (deconsiderando $b(x)$)
- $ub(x)$ é um limiar superior se $ub(x) \geq d(x, b(x))$

- Um ponto x no grupo $b(x)$ não muda de grupo depois de uma atualização de centro, se

$$lb(x) - \max_{c \in C} \delta(c) \geq ub(x) + \delta(b(x))$$

Prova da condição para filtragem

Sendo que

$ub(x) + \delta(b(x))$ é um novo limiar superior para $d(x, b')$ e $lb(x) - \max_{c \in C} \delta(c)$ é novo limiar inferior para $d(x, c')$ para outros centros c'.

Como, da desigualdade triangular reversa, obteve-se:

$$|d(x, b) - d(b, c)| \leq d(x, c) \leq d(x, b) + d(b, c)$$

então

$$d(x, c') \geq d(x, c) - d(c, c') = d(x, c) - \delta(c) \geq d(x, c) - \max_{c \in C} \delta(c)$$

E também

$$d(x, b') \leq d(x, b) + \delta(b) \leq ub(x) + \delta(b)$$

E portanto

$$d(x, b') \leq d(x, c')$$

Atualização de limiares

- Necessário armazenamento $O(n)$ váriaveis de limiares
- Atualização de limiar inferior
  - $$lb'(x) = lb(x) - \max_{c \in C} \delta(c)$$
- Atualização limiar superior
  - $$ub'(x) = ub(x) + \delta(b)$$

Algoritmo com filtragem global

- Inicializar centros com pontos aleatórios
- Fazer um passo de atribuição de pontos a grupos representados pelos centros
- Calcular $\delta(x)$, $ub(x) = \min_{c \in C} d(x, c)$, $lb(x) = \min_{b \in C - c} d(x, b)$,
- Fazer iterações de atribuição e atualização até convergir e, durante,
  - Atualizar $lb(x) = lb(x) - \max_{c \in C} (x, c)$ e $ub(x) = ub(x) + \delta(b(x))$
  - Para cada iteração somente é necessário verificar a atribuição dos pontos que $ub(x) + \delta(b(x)) > lb(x) - \max \delta(c)$

Filtros de distâncias locais

- Se houver centros com grandes mudanças, o filtro global pode ficar inefetivo
- Desempenho pode ser melhorado com filtros locais
- Mesma ideia aplicada a conjuntos de grupos, onde cada conjunto tem $t \leq k/10$ grupos
- Em vez de armazenar limiares para cada ponto, é possível armazenar limiares para conjuntos de grupos
- Dessa forma, uma grande mudança em um conjunto de grupos não afeta outros limiares

Atualização de médias acelerada

- Modificar na média somente o que foi modificado

$$c' = \frac{\left(c \cdot |V| - \left(\sum_{y \in V - OV} y\right) + \sum_{y' \in V' - OV} y'\right)}{|V'|}$$

- c é uma média
- V e V' são conjuntos de elementos antes e depois de atualização
- $OV = V \cap V'$ são os elementos que não mudaram de grupo (continuam em V)

k-médias++ [3]

1. Escolher $c_1 \in X$ aleatoriamente. Seja $C_1 = \{c_1\}$.

   Em geral seja $C_i = \{c_1, \ldots, c_i\}$
2. Para $i = 2 \ldots k$ fazer
   2.1 Escolher c_i de X com probabilidade proporcional a $d(x, \phi_{C_{i-1}}(x))^2$

- Lloyd k-médias é $(1 + \epsilon)$-aproximado em tempo $2^{(k/\epsilon)^{O}(1)}nd$ [9]
- k-médias++ é $O(\log n)$-aproximado (ou 8-aproximado se dados são bem espaçados) [2]

Mini-batch k-médias

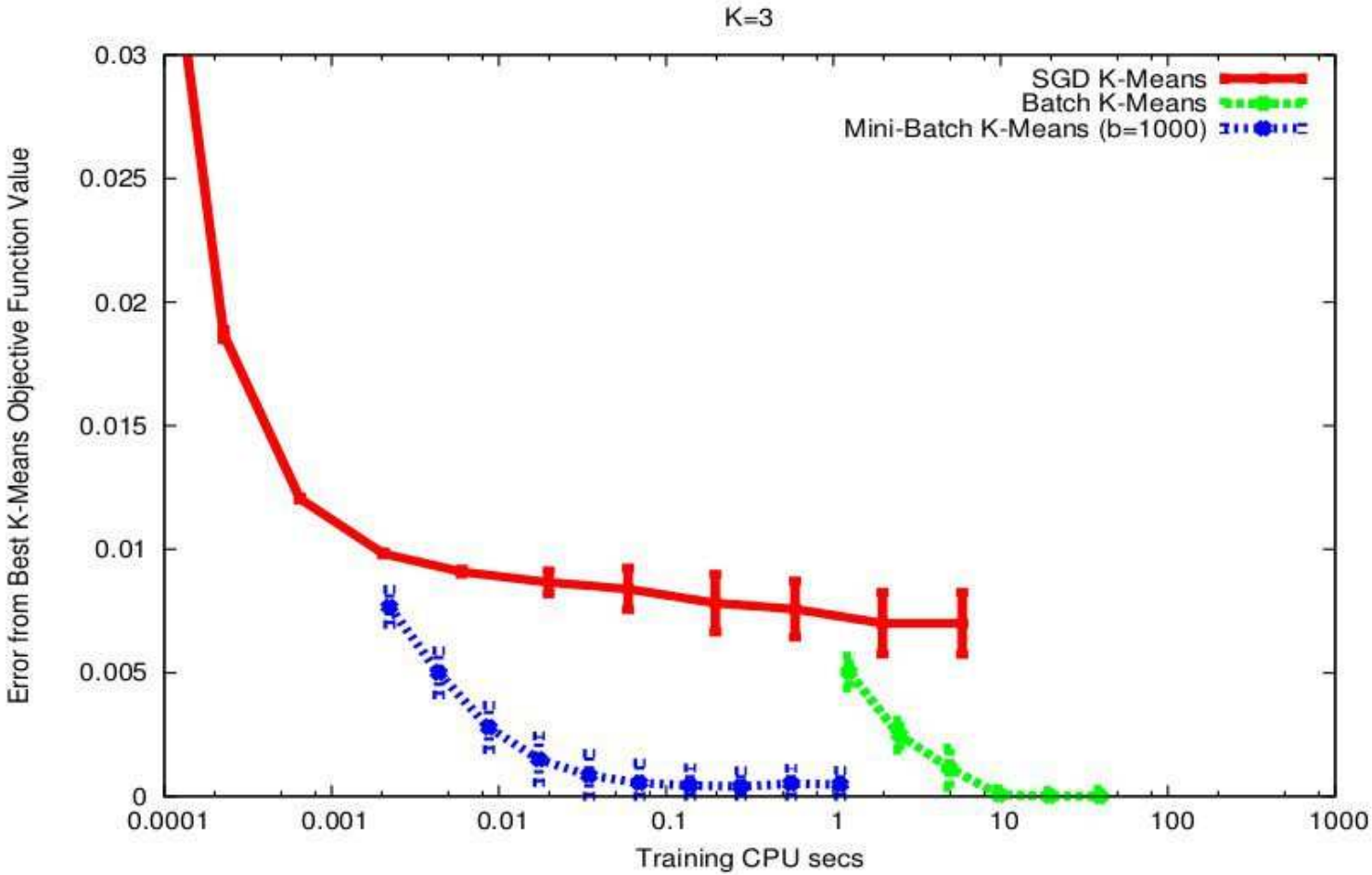
- O algoritmo de Floyd precisa de todos dados em memória (funciona em batch)
- É possível atualizar um centro por vez, mas resultado não fica muito bom
- Alternativa é usar pequenos blocos para atualizar centroides

Mini-batch k-médias

- Entradas: k, tamanho b de blocos, dataset X

Inicializar cada $c \in C$ com x aleatórios de X
$v \leftarrow 0$
for $i \in \{1 \dots t\}$ **do**
  $M \leftarrow b$ exemplos aleatórios de X
  for $x \in M$ **do**
    $d[x] = f(C, x)$ // pega centro mais perto de x
  end for
  for $x \in M$ **do**
    $c \leftarrow d[x]$ // pegar centro de x
    $v[c] \leftarrow v[c] + 1$ // atualizar contagem por centro
    $eta \leftarrow 1/v[c]$ // pega taxa de aprendizado por centro
    $c \leftarrow (1 - eta) * c + eta * x$ //passo do gradiente
  end for
end for

Comparação batch, estocástico e mini-batch



https://algorithmicthoughts.wordpress.com/2013/07/26/machine-learning-mini-batch-k-means/

k médias com maximização da expectativa (Expectation Maximization – EM)

- Problema do k-médias: cada ponto pertence a somente um cluster
- Relaxamento: para cada p, C, $\Pr(p \in C)$
- $$c = \frac{\sum_p pPr(p \in C)}{\sum_p Pr(p \in C)}$$

k-centros

- Objetivo do k-médias:

$$\min_S \sum_{k=1}^{K} \sum_{i:x_i \in C_k} (x_i - \mu_k)^T (x_i - \mu_k)$$

onde é centroide $\mu_k = \frac{1}{N_k} \sum_{i:x_j \in C_k} x_i$

- Objetivo do k-centros: achar centros $c^1, \ldots, c^k$

$$\max_j \max_{p \in C^j} \|p - c^j\|$$

ou

$$\min_S \max_{k=1,\ldots,K} \max_{i:x_i \in C_k} (x_i - \mu_k)^T (x_i - \mu_k)$$

Algoritmo k-centros (Gonzalez [7])

1. Sortear um ponto x_j como centroide inicial, fazer $h_1 = x_j$ e por em $H = \{h_1\}$
2. Inicializar distâncias ao centroide $dist(x_j), \forall j$ e o cluster atual $cluster(x_j), \forall j$
3. Para o resto dos $2, \ldots, K$ centroides, pegar ponto com maior distância x_i tirando o conjunto H
4. Adicionar x_i em $H = H \cup \{x_i\}$
5. Para cada ponto x_i, recalcular $dist(x_i)$ e $cluster(x_i)$

Propriedades do algoritmo k-centros

- Tempo de execução é $O(kn)$
- Sendo $L(\cdot, \cdot)$ uma métrica qualquer
- Seja D_k o tamanho do grupo C_k tal que D_k é o menor dos seguintes valores
  - o menor valor de distância entre pontos no grupo C_k ou
  - $D_k/2$ de algum ponto de C_k e seu centro
- Se ótimo é $D^* = \min_S \max_{k=1,\ldots,K} \max_{i,j:x_i,x_j \in C_k} L(x_i, x_j)$
- Algoritmo garante um fator de aproximação 2
- Ou seja, encontra-se D tal que $D \leq 2 * D^*$

Inicialização das médias iniciais

- Uma boa inicialização faz convergência ser mais rápida
- Bom seria cada média ser inicializada com um ponto de cada grupo
- Para k grupos, necessário avaliar $k \log k$ elementos para ter um elemento para cada grupo
- Opção 1: particionar aleatoriamente e pegar média
- Inicialização para o k-médias

Quantos grupos?

- Número de grupos k é conhecido em muitos casos

Quantos grupos?

- Número de grupos k é conhecido em muitos casos
  - E.g., pode haver restrição externa em k. Exemplo: No caso de Scatter-Gather, difícil mostrar mais que 10–20 grupos ao mesmo tempo

Quantos grupos?

- Número de grupos k é conhecido em muitos casos
  - E.g., pode haver restrição externa em k. Exemplo: No caso de Scatter-Gather, difícil mostrar mais que 10–20 grupos ao mesmo tempo
- E se não há limitação em k? Há um número "correto" de grupos?

Quantos grupos?

- Número de grupos k é conhecido em muitos casos
  - E.g., pode haver restrição externa em k. Exemplo: No caso de Scatter-Gather, difícil mostrar mais que 10–20 grupos ao mesmo tempo
- E se não há limitação em k? Há um número "correto" de grupos?
- Uma forma: definir um critério de otimização

Quantos grupos?

- Número de grupos k é conhecido em muitos casos
  - E.g., pode haver restrição externa em k. Exemplo: No caso de Scatter-Gather, difícil mostrar mais que 10–20 grupos ao mesmo tempo
- E se não há limitação em k? Há um número "correto" de grupos?
- Uma forma: definir um critério de otimização
  - Dados os elementos, encontrar k para os quais o ótimo é encontrado

Quantos grupos?

- Número de grupos k é conhecido em muitos casos
  - E.g., pode haver restrição externa em k. Exemplo: No caso de Scatter-Gather, difícil mostrar mais que 10–20 grupos ao mesmo tempo
- E se não há limitação em k? Há um número "correto" de grupos?
- Uma forma: definir um critério de otimização
  - Dados os elementos, encontrar k para os quais o ótimo é encontrado
  - Qual critério de otimização podemos usar?

Quantos grupos?

- Número de grupos k é conhecido em muitos casos
  - E.g., pode haver restrição externa em k. Exemplo: No caso de Scatter-Gather, difícil mostrar mais que 10–20 grupos ao mesmo tempo
- E se não há limitação em k? Há um número "correto" de grupos?
- Uma forma: definir um critério de otimização
  - Dados os elementos, encontrar k para os quais o ótimo é encontrado
  - Qual critério de otimização podemos usar?
  - Não podemos usar RSS ou distância média quadrática ao centroide como critério: definiria sempre $K = N$.

Função objetivo simples para k: ideia básica

- Iniciar com 1 grupo ($K = 1$)

Função objetivo simples para k: ideia básica

- Iniciar com 1 grupo ($K = 1$)
- Continuar adicionando grupos (= continua aumentando k)

Função objetivo simples para k: ideia básica

- Iniciar com 1 grupo ($K = 1$)
- Continuar adicionando grupos (= continua aumentando k)
- Adicionar uma penalidade para cada novo grupo

Função objetivo simples para k: ideia básica

- Iniciar com 1 grupo ($K = 1$)
- Continuar adicionando grupos (= continua aumentando k)
- Adicionar uma penalidade para cada novo grupo
- Então ponderar a penalidade do grupo em relação à distância quadrática média

Função objetivo simples para k: ideia básica

- Iniciar com 1 grupo ($K = 1$)
- Continuar adicionando grupos (= continua aumentando k)
- Adicionar uma penalidade para cada novo grupo
- Então ponderar a penalidade do grupo em relação à distância quadrática média
- Escolher valores de k com melhor ponderação penalidade/distância quadrática média

Função objetivo simples para k

- Dado um agrupamento, definir o custo para um documento como a distância quadrática para o centroide

Função objetivo simples para k

- Dado um agrupamento, definir o custo para um documento como a distância quadrática para o centroide
- Definir distorção total RSS(K) como a soma de todos os custos individuais de cada documento (corresponde à distância média)

Função objetivo simples para k

- Dado um agrupamento, definir o custo para um documento como a distância quadrática para o centroide
- Definir distorção total RSS(K) como a soma de todos os custos individuais de cada documento (corresponde à distância média)
- Então, penalizar cada grupo com um custo λ

Função objetivo simples para k

- Dado um agrupamento, definir o custo para um documento como a distância quadrática para o centroide
- Definir distorção total RSS(K) como a soma de todos os custos individuais de cada documento (corresponde à distância média)
- Então, penalizar cada grupo com um custo λ
- Então para cada agrupamento com k grupo, penalidade total de grupos é $K\lambda$

Função objetivo simples para k

- Dado um agrupamento, definir o custo para um documento como a distância quadrática para o centroide
- Definir distorção total RSS(K) como a soma de todos os custos individuais de cada documento (corresponde à distância média)
- Então, penalizar cada grupo com um custo λ
- Então para cada agrupamento com k grupo, penalidade total de grupos é $K\lambda$
- Definir o custo total de um agrupamento como a distorção mais a penalidade total de grupo: RSS(K) + $K\lambda$

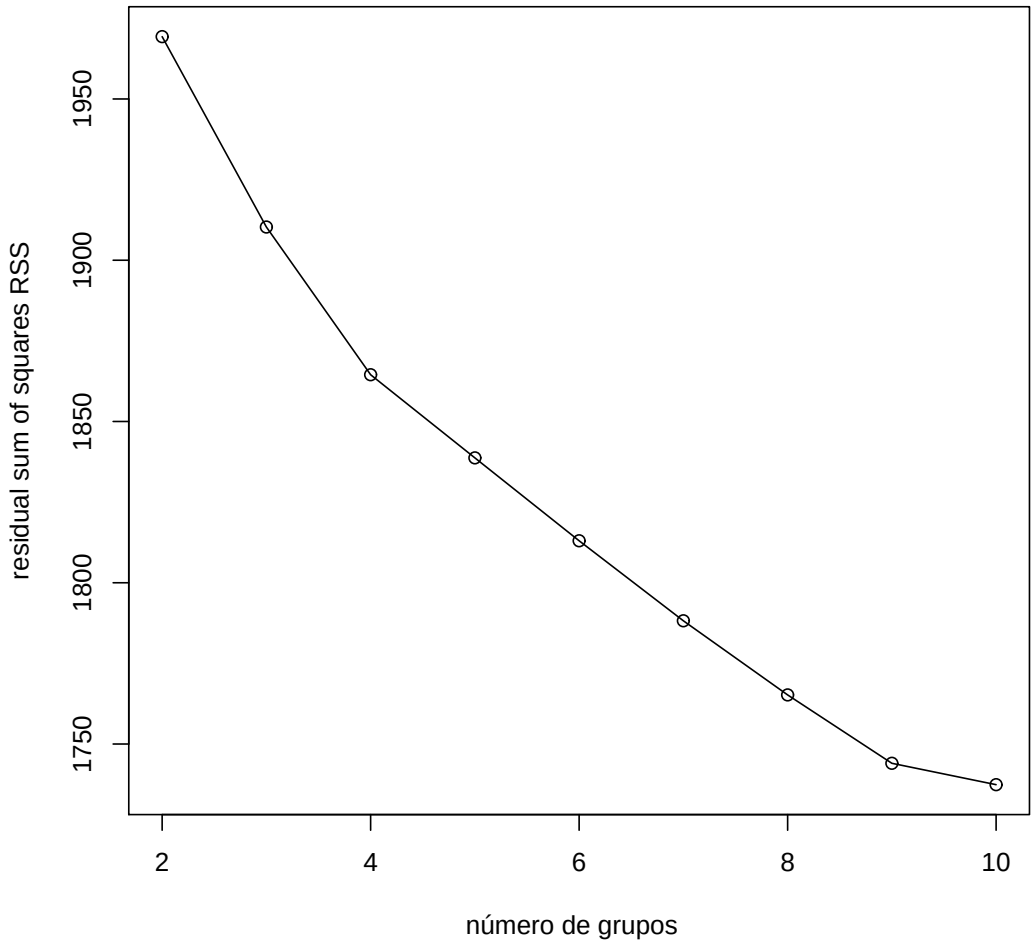
Função objetivo simples para k

- Dado um agrupamento, definir o custo para um documento como a distância quadrática para o centroide
- Definir distorção total RSS(K) como a soma de todos os custos individuais de cada documento (corresponde à distância média)
- Então, penalizar cada grupo com um custo λ
- Então para cada agrupamento com k grupo, penalidade total de grupos é $K\lambda$
- Definir o custo total de um agrupamento como a distorção mais a penalidade total de grupo: RSS(K) + $K\lambda$
- Selecionar k que minimiza (RSS(K) + $K\lambda$)

Função objetivo simples para k

- Dado um agrupamento, definir o custo para um documento como a distância quadrática para o centroide
- Definir distorção total RSS(K) como a soma de todos os custos individuais de cada documento (corresponde à distância média)
- Então, penalizar cada grupo com um custo λ
- Então para cada agrupamento com k grupo, penalidade total de grupos é $K\lambda$
- Definir o custo total de um agrupamento como a distorção mais a penalidade total de grupo: RSS(K) + $K\lambda$
- Selecionar k que minimiza (RSS(K) + $K\lambda$)
- Ainda precisa determinar bom valor para λ ...

Encontrar o "joelho" da curva



Escolher o número de grupos onda a curva "dobra". Aqui: 4 ou 9.

DBSCAN [6]

- Problema k-médias tende a reconhecer grupos com formas esferoides
- DBSCAN reconhece grupos de acordo com sua densidade
- Cada ponto tem um raio epsilon em sua volta que engloba outros pontos
- Se o número de pontos for maior que um mínimo **min_points**, então pontos pertencem ao mesmo grupo
- Isso ocorre recursivamente para outros pontos

O que é um bom agrupamento?

- Critério interno

O que é um bom agrupamento?

- Critério interno
  - Exemplo de um critério interno: RSS no k-médias

O que é um bom agrupamento?

- Critério interno
  - Exemplo de um critério interno: RSS no k-médias
- Mas um critério interno não avalia diretamente a utilidade de um agrupamento na aplicação

O que é um bom agrupamento?

- Critério interno
  - Exemplo de um critério interno: RSS no k-médias
- Mas um critério interno não avalia diretamente a utilidade de um agrupamento na aplicação
- Alternativa: critério externo

O que é um bom agrupamento?

- Critério interno
  - Exemplo de um critério interno: RSS no k-médias
- Mas um critério interno não avalia diretamente a utilidade de um agrupamento na aplicação
- Alternativa: critério externo
  - Avaliar de acordo com uma classificação definida previamente

Critério externo para qualidade de agrupamento

- Usar conjunto de documentos conhecido

Critério externo para qualidade de agrupamento

- Usar conjunto de documentos conhecido
- Meta: agrupamento deve reproduzir classes conhecidas

Critério externo para qualidade de agrupamento

- Usar conjunto de documentos conhecido
- Meta: agrupamento deve reproduzir classes conhecidas
- Mas somente queremos reproduzir como os documento são divididos em grupos, sem usar os rótulos para treinamento

Critério externo para qualidade de agrupamento

- Usar conjunto de documentos conhecido
- Meta: agrupamento deve reproduzir classes conhecidas
- Mas somente queremos reproduzir como os documento são divididos em grupos, sem usar os rótulos para treinamento
- Medida de quão bem conseguimos reproduzir as classes: pureza

Critério externo: pureza

$$\text{pureza}(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ é o conjunto de grupos e $C = \{c_1, c_2, \ldots, c_J\}$ é o conjunto de classes.

Critério externo: pureza

$$\text{pureza}(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

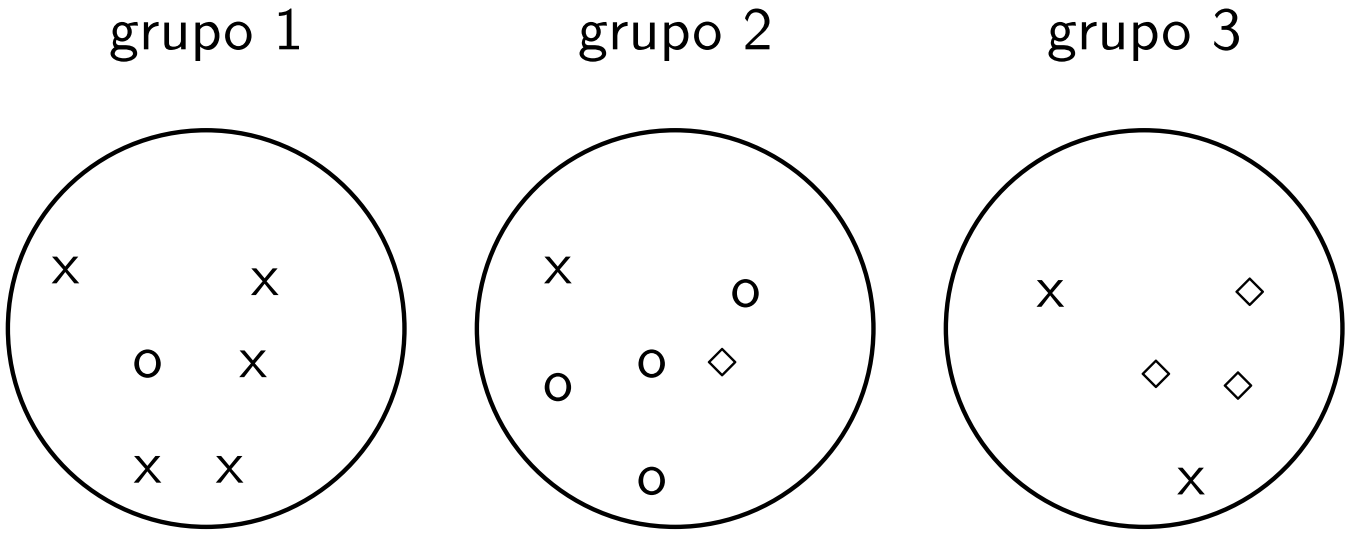
- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ é o conjunto de grupos e $C = \{c_1, c_2, \ldots, c_J\}$ é o conjunto de classes.
- Para cada grupo ω_k: encontrar classe c_j com mais membros n_{kj} em ω_k

Critério externo: pureza

$$\text{pureza}(\Omega, C) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

- $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ é o conjunto de grupos e $C = \{c_1, c_2, \ldots, c_J\}$ é o conjunto de classes.
- Para cada grupo ω_k: encontrar classe c_j com mais membros n_{kj} em ω_k
- Soma todos os n_{kj} e dividir pelo número total de pontos

Exemplo: pureza



Para calcular a pureza: $5 = \max_j |\omega_1 \cap c_j|$ (classe x, grupo 1); $4 = \max_j |\omega_2 \cap c_j|$ (classe o, grupo 2); e $3 = \max_j |\omega_3 \cap c_j|$ (classe $\diamond$, grupo 3). Pureza é $(1/17) \times (5 + 4 + 3) \approx 0.71$.

Outro critério externo: índice Rand

- Pureza pode aumentar com k

Outro critério externo: índice Rand

- Pureza pode aumentar com k
- Índice Rand não tem esse problema

Outro critério externo: índice Rand

- Pureza pode aumentar com k
- Índice Rand não tem esse problema
- Definição: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$

Outro critério externo: índice Rand

- Pureza pode aumentar com k
- Índice Rand não tem esse problema
- Definição: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$
- Baseado em tabela de contingência 2x2 com todos os pares de documentos:

	mesmo grupo	grupos distintos
mesma classe	verd. positivos (TP)	falsos negativos (FN)
classes distintas	falsos pos. (FP)	verdadeiros neg. (TN)

Outro critério externo: índice Rand

- Pureza pode aumentar com k
- Índice Rand não tem esse problema
- Definição: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$
- Baseado em tabela de contingência 2x2 com todos os pares de documentos:

	mesmo grupo	grupos distintos
mesma classe	verd. positivos (TP)	falsos negativos (FN)
classes distintas	falsos pos. (FP)	verdadeiros neg. (TN)

- TP+FN+FP+TN é o número total de número de pares

Outro critério externo: índice Rand

- Pureza pode aumentar com k
- Índice Rand não tem esse problema
- Definição: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$
- Baseado em tabela de contingência 2x2 com todos os pares de documentos:

	mesmo grupo	grupos distintos
mesma classe	verd. positivos (TP)	falsos negativos (FN)
classes distintas	falsos pos. (FP)	verdadeiros neg. (TN)

- TP+FN+FP+TN é o número total de número de pares
- TP+FN+FP+TN = $\binom{N}{2}$ para N documentos.

Outro critério externo: índice Rand

- Pureza pode aumentar com k
- Índice Rand não tem esse problema
- Definição: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$
- Baseado em tabela de contingência 2x2 com todos os pares de documentos:

	mesmo grupo	grupos distintos
mesma classe	verd. positivos (TP)	falsos negativos (FN)
classes distintas	falsos pos. (FP)	verdadeiros neg. (TN)

- TP+FN+FP+TN é o número total de número de pares
- TP+FN+FP+TN $= \binom{N}{2}$ para N documentos.
- Exemplo: $\binom{17}{2} = 136$ no exemplo o/⋄/x

Outro critério externo: índice Rand

- Pureza pode aumentar com k
- Índice Rand não tem esse problema
- Definição: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$
- Baseado em tabela de contingência 2x2 com todos os pares de documentos:

	mesmo grupo	grupos distintos
mesma classe	verd. positivos (TP)	falsos negativos (FN)
classes distintas	falsos pos. (FP)	verdadeiros neg. (TN)

- TP+FN+FP+TN é o número total de número de pares
- TP+FN+FP+TN $= \binom{N}{2}$ para N documentos.
- Exemplo: $\binom{17}{2} = 136$ no exemplo o/⋄/x
- Cada par é ou positivo ou negativo (o agrupamento coloca documentos no mesmo ou em grupos diferentes) . . .

Outro critério externo: índice Rand

- Pureza pode aumentar com k
- Índice Rand não tem esse problema
- Definição: $\mathrm{RI} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}}$
- Baseado em tabela de contingência 2x2 com todos os pares de documentos:

	mesmo grupo	grupos distintos
mesma classe	verd. positivos (TP)	falsos negativos (FN)
classes distintas	falsos pos. (FP)	verdadeiros neg. (TN)

- TP+FN+FP+TN é o número total de número de pares
- TP+FN+FP+TN = $\binom{N}{2}$ para N documentos.
- Exemplo: $\binom{17}{2} = 136$ no exemplo o/⋄/x
- Cada par é ou positivo ou negativo (o agrupamento coloca documentos no mesmo ou em grupos diferentes) . . .
- . . . e é "true" (verdadeiro/correto) ou "false" (falso/incorreto): a decisão de agrupamento está correta ou incorreta

Índice Rand (RI)

Como examplo, calculamos o RI para o exemplo o/$\diamond$/x. Primeiro, calculamos TP + FP. Os três grupos contém 6, 6, e 5 pontos, respectivamente, então o número total de "positivos" ou pares de elementos que estão no mesmo grupo é:

$$\text{TP} + \text{FP} = \binom{6}{2} + \binom{6}{2} + \binom{5}{2} = 40$$

Desse, os pares x no grupo 1, os pares o no grupo 2, so pares $\diamond$ no grupo 3, e os pares x grupo 3 são verdadeiros positivos:

$$\text{TP} = \binom{5}{2} + \binom{4}{2} + \binom{3}{2} + \binom{2}{2} = 20$$

Então, $\text{FP} = 40 - 20 = 20$.
FN e TN são calculamos de maneira similar

Índice Rand (RI) para exemplo o/◇/x

	mesmo grupo	grupos distintos
mesma classe	$\mathrm{TP} = 20$	$\mathrm{FN} = 24$
classes distintas	$\mathrm{FP} = 20$	$\mathrm{TN} = 72$

RI é então $(20 + 72)/(20 + 20 + 24 + 72) \approx 0.68$.

Outras medidas de avaliação externa

- Informação mútua normalizada (IMN)

Outras medidas de avaliação externa

- Informação mútua normalizada (IMN)
  - Quanta informação o agrupamento tem sobre a classificação?

Outras medidas de avaliação externa

- Informação mútua normalizada (IMN)
  - Quanta informação o agrupamento tem sobre a classificação?
  - Grupos unitários (número de grupos = número de elementos) tem informação mútua máxima

Outras medidas de avaliação externa

- Informação mútua normalizada (IMN)
  - Quanta informação o agrupamento tem sobre a classificação?
  - Grupos unitários (número de grupos = número de elementos) tem informação mútua máxima
  - Portanto: normalizar pela entropia dos grupos e classes

Outras medidas de avaliação externa

- Informação mútua normalizada (IMN)
  - Quanta informação o agrupamento tem sobre a classificação?
  - Grupos unitários (número de grupos = número de elementos) tem informação mútua máxima
  - Portanto: normalizar pela entropia dos grupos e classes
- Medida F

Referências

N. Ailon, R. Jaiswal, and C. Monteleoni.
Streaming k-means approximation.
In *Advances in Neural Information Processing Systems*, pages 10–18, 2009.

D. Arthur, B. Manthey, and H. Röglin.
Smoothed analysis of the k-means method.
Journal of the ACM (JACM), 58(5):19, 2011.

D. Arthur and S. Vassilvitskii.
k-means++: The advantages of careful seeding.
In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035.
Society for Industrial and Applied Mathematics, 2007.

B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii.
Scalable k-means++.
Proceedings of the VLDB Endowment, 5(7):622–633, 2012.

Y. Ding, Y. Zhao, X. Shen, M. Musuvathi, and T. Mytkowicz.
Yinyang k-means: A drop-in replacement of the classic k-means with consistent speedup.
In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 579–587, 2015.

M. Ester, H.-P. Kriegel, J. Sander, and X. Xu.
A density-based algorithm for discovering clusters in large spatial databases with noise.
In *Kdd*, volume 96, pages 226–231, 1996.

T. F. Gonzalez.
Clustering to minimize the maximum intercluster distance.
Theoretical Computer Science, 38:293–306, 1985.

S. Guha, A. Meyerson, N. Mishra, R. Motwani, and L. O'Callaghan.
Clustering data streams: Theory and practice.
Knowledge and Data Engineering, IEEE Transactions on, 15(3):515–528, 2003.

Referências (cont.)

A. Kumar, Y. Sabharwal, and S. Sen.
A simple linear time (1+ ε)-approximation algorithm for geometric k-means clustering in any dimensions.
In *Proceedings-Annual Symposium on Foundations of Computer Science*, pages 454–462. IEEE, 2004.

S. P. Lloyd.
Least squares quantization in pcm.
Information Theory, IEEE Transactions on, 28(2):129–137, 1982.