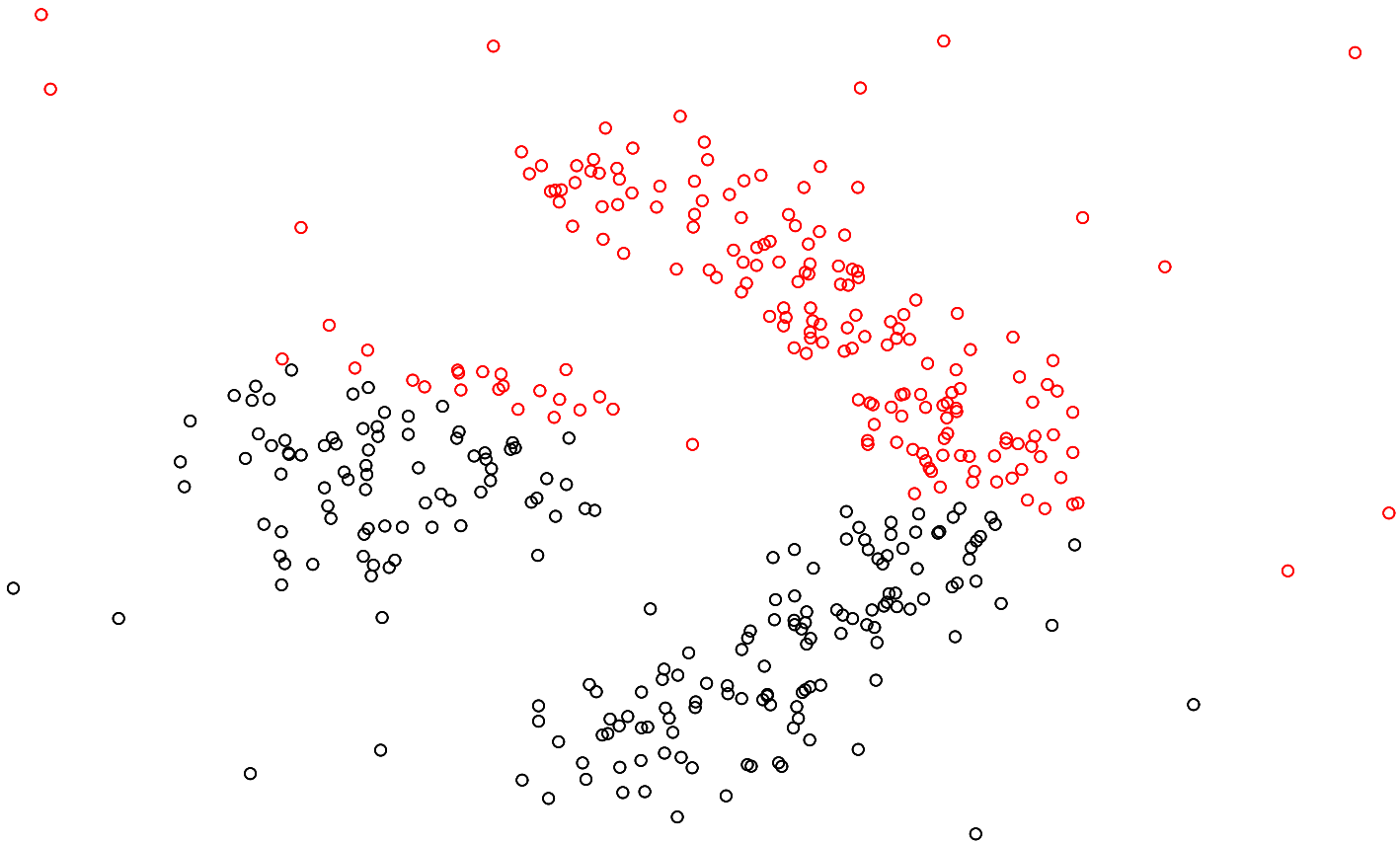
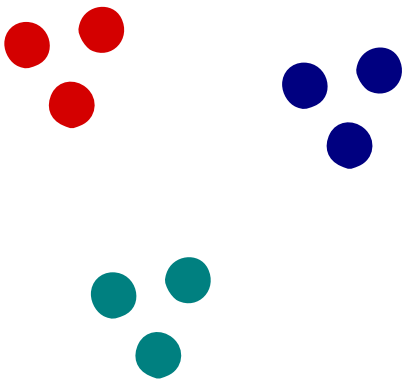
Agrupamento de dados

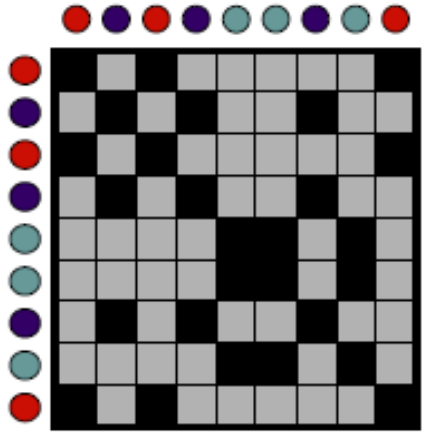
- Critério 1: “grupos são concentrações de dados“ – k-means
- Critério 2: “grupos são conjuntos de elementos próximos entre si“ – espectral

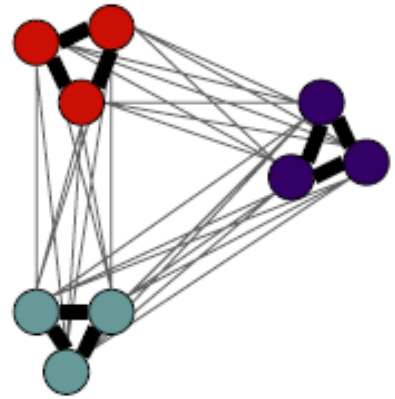


Dados e grafos

- Se temos dados x_i, $i \in 0 \ldots n$, criamos grafo com arestas ponderadas de similaridades







- Objetivo: particionar grafo tal que arestas em um grupo tem pesos altos e arestas entre grupos tem pesos baixos

Criação de grafos de similaridades

$$W_{ij} = \exp^{-\frac{\|x_i - x_j\|^2}{\sigma^2}}$$

i e j indicam vértices, x_i, x_j indica vetores de dados, W indica matriz de similaridades

σ controla tamanho da vizinhança

Cenário simplificado: corte de um grafo em 2 grupos

- Problema do **corte mínimo**: min-cut
- Obter sub-grafos A e B tal que o corte é mínimo:

$$cut(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

- Costuma isolar vértices

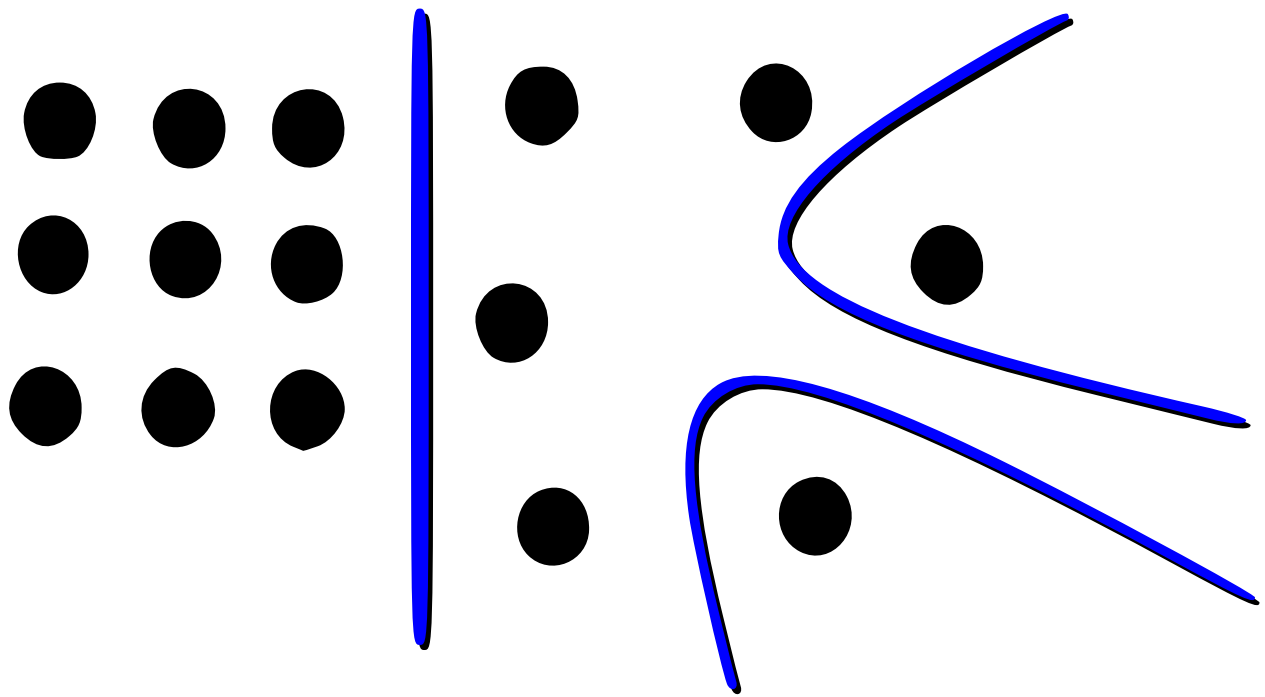


Figura: Fonte: Aarti Singh (2010) e Shi & Malik (2000)

Particionamento em 2 grupos com corte normalizado

- Corte entre A e B de ser mínimo e **tamanho** de A e B devem ser similares

$$Ncut(A, B) = cut(A, B) \left(\frac{1}{vol(A)} + \frac{1}{vol(B)} \right)$$

onde $vol(A) = \sum_{i \in A} d_i$ e o grau de i é

$$d_i = \sum_{j=1}^{n} w_{ij}$$

- Spectral clustering é uma relaxação deste problema

Laplaciano de um grafo

- D é a matriz de grau dos vértices com d_i na diagonal principal
- W é a matriz de adjacências ponderadas
- $L = D - W$ é o Laplaciano do grafo:

$$L_{i,j} = \begin{cases} grau(v_i) & \text{se } i = j \\ -w_{ij} & \text{se } i \neq j \text{ e } v_i \text{ adjacente a } v_j \\ 0 & \text{caso contrário} \end{cases}$$

Corte normalizado e Laplaciano de um grafo

- Corte normalizado: $Ncut(A,B) = cut(A,B)\left(\frac{1}{vol(A)} + \frac{1}{vol(B)}\right)$
- Seja $\mathbf{f} = [f_1 f_2 \dots f_n]^T$ com $f_i = \begin{cases} \frac{1}{vol(A)} & \text{se } i \in A \\ -\frac{1}{vol(B)} & \text{se } i \in B \end{cases}$

Reescrita no formato de álgebra de matrizes

- $2\mathbf{f}^T\mathbf{L}\mathbf{f} = \sum_{ij} w_{ij}(\mathbf{f}_i - \mathbf{f}_j)^2 = \sum_{i\in A, j\in B} w_{ij}\left(\frac{1}{vol(A)} + \frac{1}{vol(B)}\right)^2$
- $\mathbf{f}^T\mathbf{D}\mathbf{f} = \sum_j d_i\mathbf{f}_i^2 = \sum_{i\in A}\frac{d_i}{vol(A)^2} + \sum_{j\in B}\frac{d_j}{vol(B)^2} = \frac{1}{vol(A)} + \frac{1}{vol(B)}$
- $Ncut(A,B) = 2\frac{\mathbf{f}^T\mathbf{L}\mathbf{f}}{\mathbf{f}^T\mathbf{D}\mathbf{f}}$
- Então, o objetivo do corte normalizado é min $\frac{\mathbf{f}^T\mathbf{L}\mathbf{f}}{\mathbf{f}^T\mathbf{D}\mathbf{f}}$

Prova que $\mathbf{f}^T\mathbf{L}\mathbf{f} = \sum_{ij} w_{ij}(\mathbf{f}_i - \mathbf{f}_j)^2$

- Seja $f = [f_1 f_2 \dots f_n]^T$ com $f_i = \begin{cases} \frac{1}{vol(A)} & \text{se } i \in A \\ -\frac{1}{vol(B)} & \text{se } i \in B \end{cases}$
- Da definição de L temos
- $\mathbf{f}^T\mathbf{L}\mathbf{f} = \mathbf{f}^T\mathbf{D}\mathbf{f} - \mathbf{f}^T\mathbf{W}\mathbf{f}$
- $= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$
- $= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} d_j f_j^2\right)$
- $= \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$

Relaxação do corte normalizado

- O problema de corte torna-se encontrar $\mathbf{f}$:

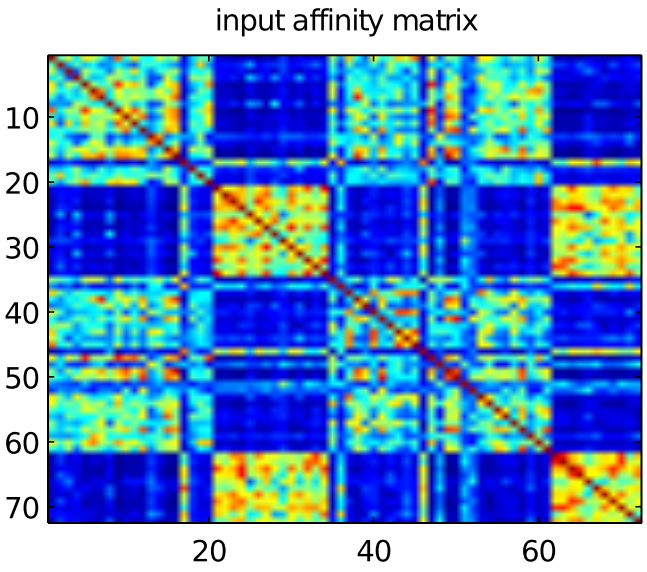
$$\min Ncut(A, B) = \min \frac{\mathbf{f}^T \mathbf{L}\mathbf{f}}{\mathbf{f}^T \mathbf{D}\mathbf{f}} \text{ s.a. } \mathbf{f}^T \mathbf{D}\mathbf{1} = \mathbf{0}$$

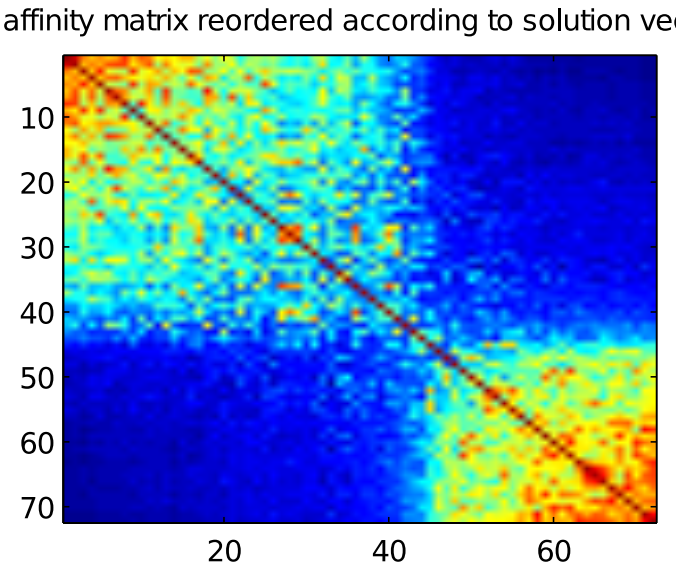
- Usando o teorema do quociente de Rayleigh, e relaxar para $\mathbf{f} \in \mathbb{R}^n$ a solução é o auto-vetor com o segundo menor auto-valor de:

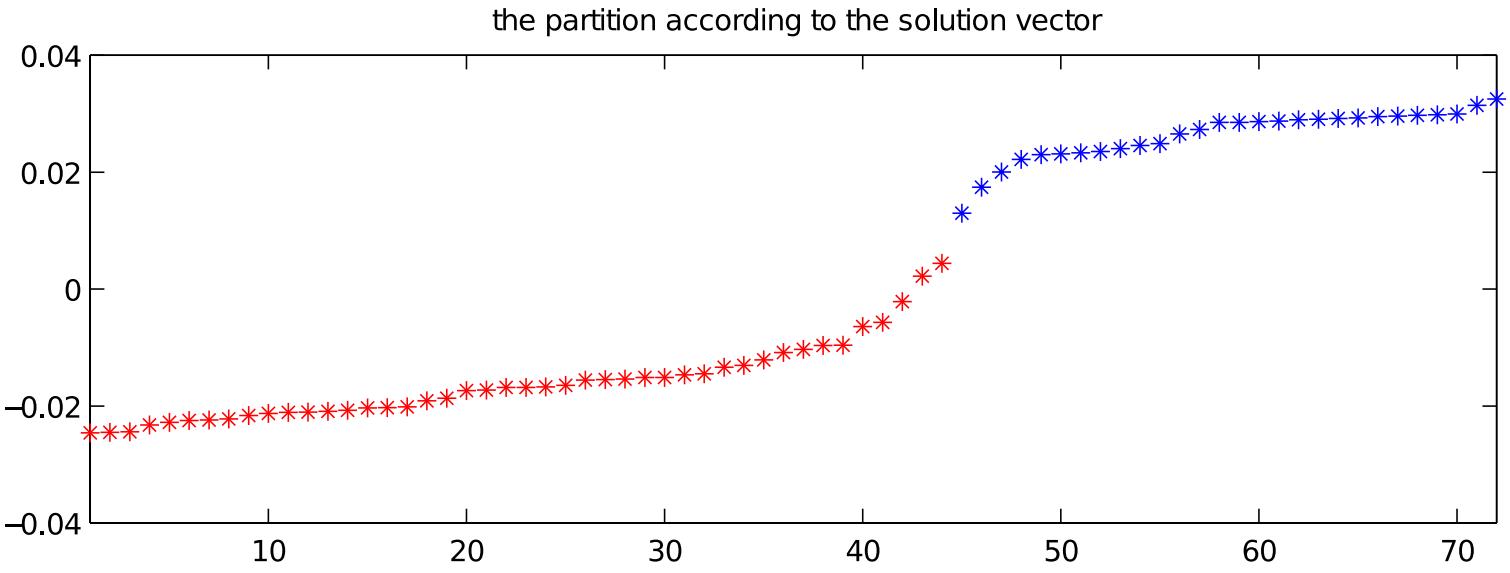
$$\mathbf{L}f = \lambda \mathbf{D}\mathbf{f}$$

- Obter grupos ao separar $\mathbf{f}$ entre positivos e negativos
- Comentários:
  - Para auto-vetor $\mathbf{v}$ e auto-valor λ da matriz $\mathbf{A}$: $\mathbf{A}\mathbf{v} = \lambda \mathbf{v}$
  - Se $\mathbf{f}$ é auto-vetor com auto-valor 0, então $\mathbf{L}\mathbf{f} = \mathbf{0} \Rightarrow \mathbf{f}^T \mathbf{L}\mathbf{f} = 0$
  - Então $\mathbf{0} = \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2$
  - Se v_i e v_j são conectados então $f_i = f_j$
  - Se grafo tem somente um componente conectado então somente um auto-vetor constante $\mathbf{1}$ tem auto-valor 0

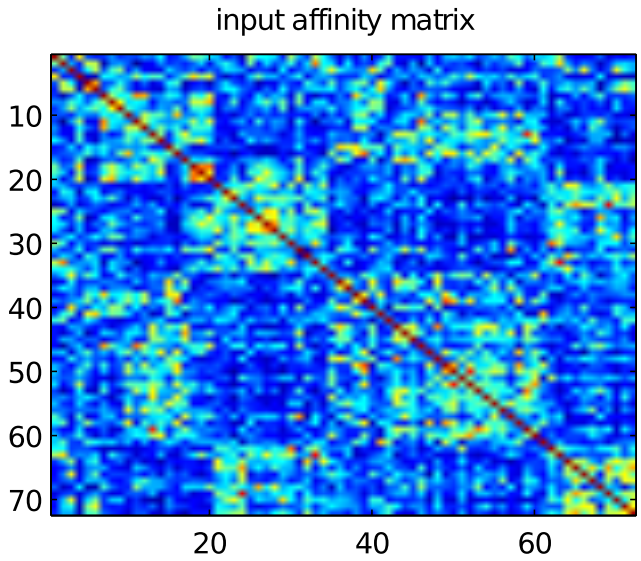
Xing 2001 (DOI: 10.1093/bioinformatics/17.suppl_1.S306)

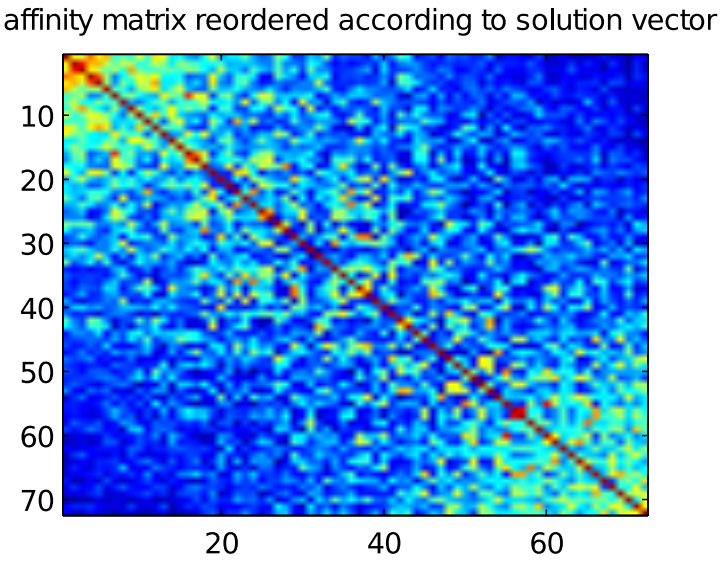


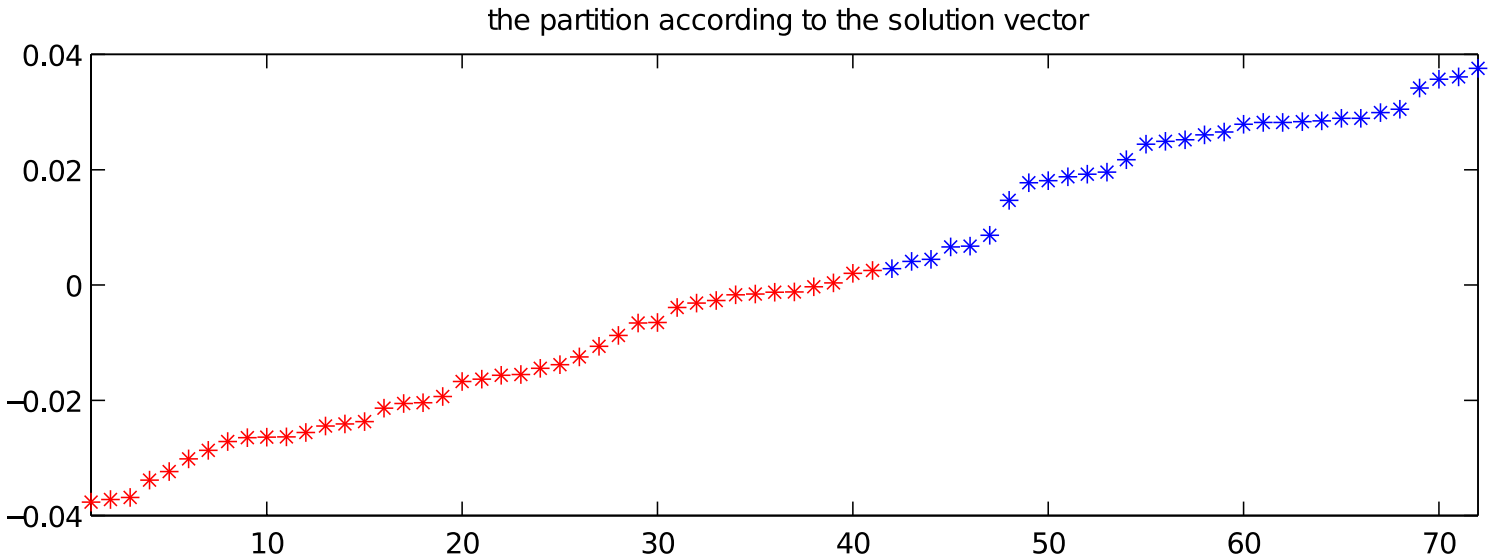




Xing 2001 (DOI: 10.1093/bioinformatics/17.suppl_1.S306)







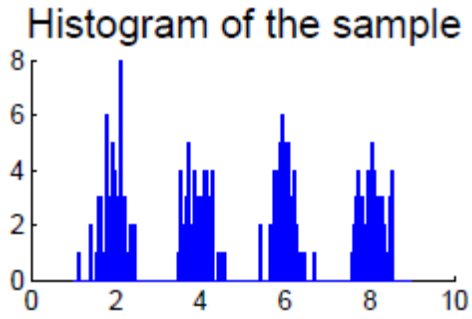
Como particionar em k grupos?

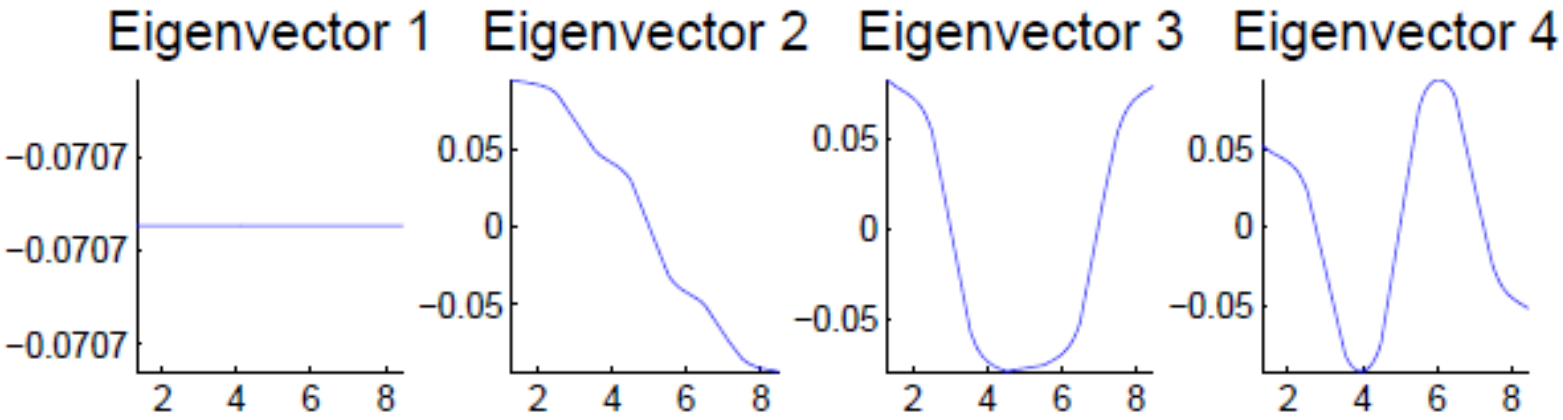
- Receber matriz de similaridades W e k e computar laplaciano L
- Computar k auto-vetores $\mathbf{f}_1, \ldots, \mathbf{f}_k$ com menores auto-valores associados
- Montar matriz $F \in \mathbb{R}^{n \times k}$
- Interpretar colunas de F como novos pontos de dados $F_i \in \mathbb{R}^k$

	f_1	f_2	f_3
Z_1	f_{11}	f_{12}	f_{13}
$\vdots$	$\vdots$	$\vdots$	
Z_n	f_{n1}	f_{n2}	f_{n3}

- Agrupar pontos Z_i usando k-means

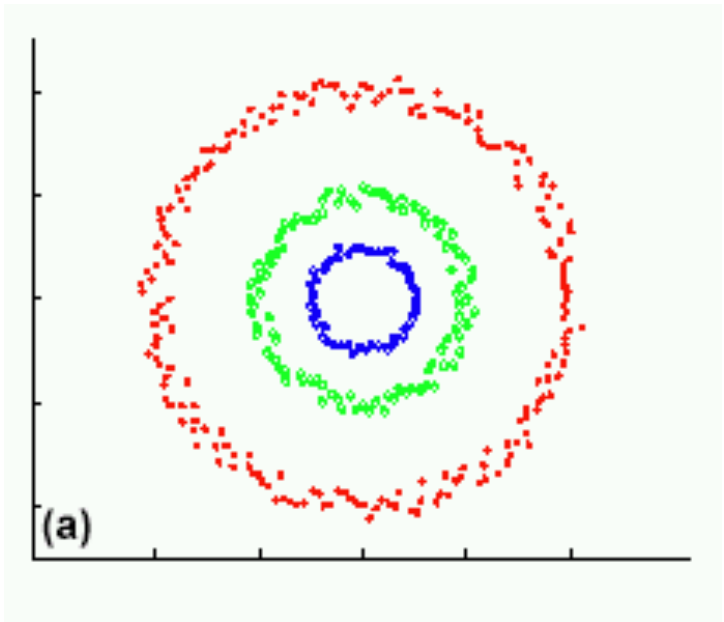
Como funciona

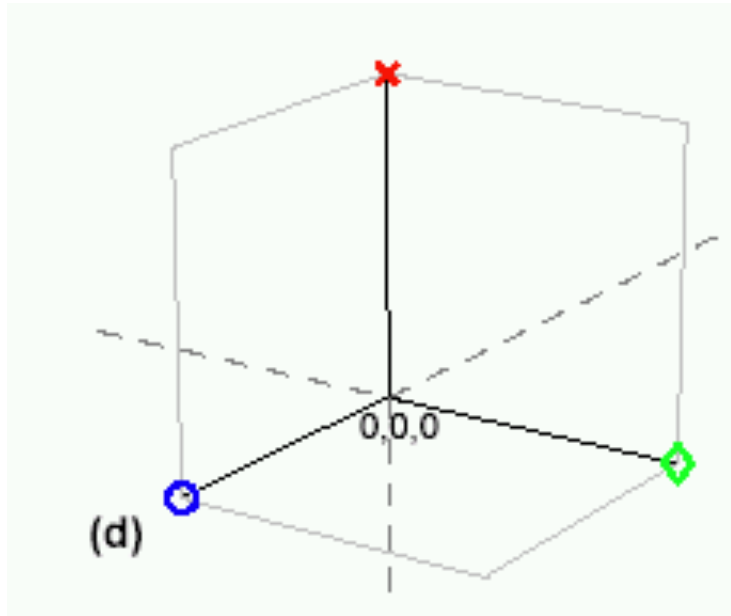




- Se grafo é conectado, então eigenvector 1 é constante
- Outros autovetores podem ser usados para separar grupos

Como funciona



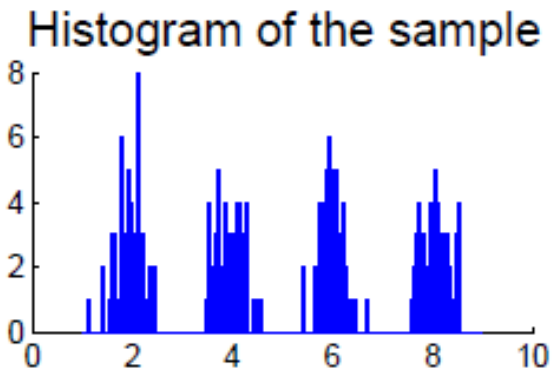


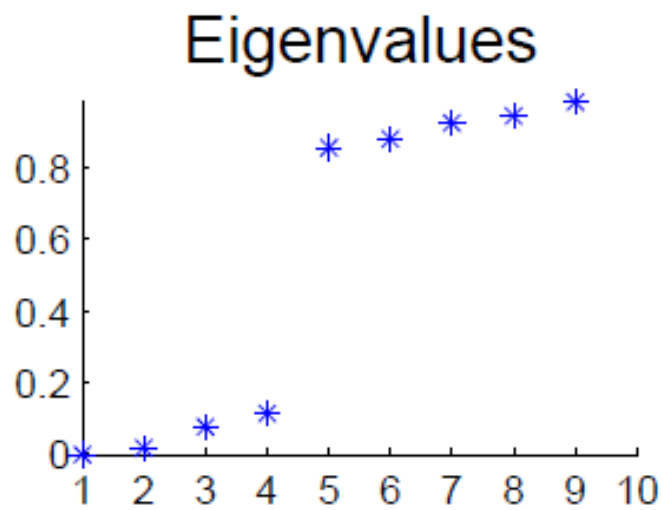
- Dados são projetados em um espaço fácil de separar os grupos

Escolha de número de grupos k

- Como escolher número de grupos k
- Usar o Δ_k que é a maior diferença entre auto-valores consecutivos

$$\Delta_k = |\lambda_k - \lambda_{k-1}|$$





Outras questões

- Escolha de medida de similaridade: kernel
- Escolha de parâmetros do kernel: gaussiana σ
- Modo de agrupamento
  - De dois em dois
  - Usando k-médias
- Dificuldade computacional para obter auto-vetores
- Equivalência a k-médias com kernel apropriado (Dhillon et al., 2007)

Algoritmo para computar auto-vetores

- Queremos encontrar k autovetores $\mathbf{f}$ de L
- Exemplo de algoritmo
  - Usar algoritmo de Lanczos para transformar L (matriz positiva-definida) em uma matriz tri-diagonal $\mathbf{\Delta}$
  - Autovalores λ_i estarão na diagonal de $\mathbf{\Delta}$
  - Autovetores para cada λ_i serão solução de $(\mathbf{\Delta} - \lambda_i \mathbf{I})\mathbf{v} = \mathbf{0}$

Algoritmo de Lanczos

- Calcula-se auto-vetor da matrix **A**
- Elementos na diagonal são α_i e fora da diagonal são β_j
- $m \approx 1.5k$, se queremos k auto-vetores
- Usar métodos para garantir estabilidade

$\mathbf{v}_0 \leftarrow \mathbf{0}$
$\mathbf{v}_1 \leftarrow$ vetor aleatório com norma 1
$\beta_1 \leftarrow 0$
for $j = 1, \ldots, m-1$ **do**
  $\mathbf{w}_j \leftarrow \mathbf{A}\mathbf{v}_j$
  $\alpha_j \leftarrow \mathbf{w}_j' \cdot \mathbf{v}_j$
  $\mathbf{w}_j \leftarrow \mathbf{w}_j - \alpha_j \mathbf{v}_j - \beta_j \mathbf{v}_{j-1}$
  $\beta_{j+1} \leftarrow ||\mathbf{w}_j||$
  $\mathbf{v}_{j+1} \leftarrow \frac{\mathbf{w}_j}{\beta_{j+1}}$
end for
$\mathbf{w}_m \leftarrow \mathbf{A}\mathbf{v}_m$
$\alpha_m \leftarrow \mathbf{w}_m' \cdot \mathbf{v}_m$

Equivalência entre clustering spectral e kernel k-means

- kernel k-means = weighted graph clustering (Dhillon et al, 2007)
- Ideia da prova:
  - kernel k-means pode ser escrito como maximização do traço de uma matriz baseado nas similaridade entre instâncias
  - agrupamento espectral também pode ser escrito no mesmo tipo de maximização do traço

Kernel k-means

- Kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$
- As k-médias estão no espaço de características ϕ: μ_i^ϕ

$$\mu_i^\phi = \frac{1}{n_i} \sum_{x_j \in C_i} \phi(\mathbf{x}_j)$$

- A função objetivo será

$$\min_{\mathcal{C}} SSE(\mathcal{C}) = \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in \mathcal{C}_i} \|\phi(\mathbf{x}_j) - \mu_i^\phi\|^2$$

- Objetivo somente em termos do Kernel $K(\cdot, \cdot)$

$$SSE(\mathcal{C}) = \sum_{j=1}^{n} K(\mathbf{x}_j, \mathbf{x}_j) - \sum_{i=1}^{k} \frac{1}{n_i} \sum_{x_a \in \mathcal{C}_i} \sum_{x_b \in \mathcal{C}_i} K(\mathbf{x}_a, \mathbf{x}_b)$$

Atribuição de pontos a grupos

- Para calcular a distância à média

$$\|\phi(\mathbf{x}_j) - \mu_i^\phi\|^2 = \|\phi(\mathbf{x}_j)\|^2 - 2\phi(\mathbf{x}_j)^T\mu_i^\phi + \|\mu_i^\phi\|^2$$

- Após expandir

$$\|\phi(\mathbf{x}_j) - \mu_i^\phi\|^2 = K(\mathbf{x}_i, \mathbf{x}_j) - \frac{2}{n_i}\sum_{\mathbf{x}_a \in C_i} K(\mathbf{x}_a, \mathbf{x}_j) + \frac{1}{n_i^2}\sum_{\mathbf{x}_a \in C_i}\sum_{\mathbf{x}_b \in C_i} K(\mathbf{x}_a, \mathbf{x}_b)$$

- Usar somente os dois últimos termos para encontrar a média mais próxima

ALGORITHM 13.2. Kernel K-means Algorithm

KERNEL-KMEANS($\mathbf{K}, k, \epsilon$):

$t \leftarrow 0$

$\mathcal{C}^t \leftarrow \{C_1^t, \ldots, C_k^t\}$ // Randomly partition points into k clusters

repeat

  $t \leftarrow t+1$

  foreach $C_i \in \mathcal{C}^{t-1}$ **do** // Compute squared norm of cluster means

    $\text{sqnorm}_i \leftarrow \frac{1}{n_i^2} \sum_{\mathbf{x}_a \in C_i} \sum_{\mathbf{x}_b \in C_i} K(\mathbf{x}_a, \mathbf{x}_b)$

  foreach $\mathbf{x}_j \in \mathbf{D}$ **do** // Average kernel value for $\mathbf{x}_j$ and C_i

    foreach $C_i \in \mathcal{C}^{t-1}$ **do**

      $\text{avg}_{ji} \leftarrow \frac{1}{n_i} \sum_{\mathbf{x}_a \in C_i} K(\mathbf{x}_a, \mathbf{x}_j)$

  // Find closest cluster for each point

  foreach $\mathbf{x}_j \in \mathbf{D}$ **do**

    foreach $C_i \in \mathcal{C}^{t-1}$ **do**

      $d(\mathbf{x}_j, C_i) \leftarrow \text{sqnorm}_i - 2 \cdot \text{avg}_{ji}$

    $j^* \leftarrow \arg\min_i \{d(\mathbf{x}_j, C_i)\}$

    $C_{j^*}^t \leftarrow C_{j^*}^t \cup \{\mathbf{x}_j\}$ // Cluster reassignment

  $\mathcal{C}^t \leftarrow \{C_1^t, \ldots, C_k^t\}$

until $1 - \frac{1}{n} \sum_{i=1}^{k} \left| C_i^t \cap C_i^{t-1} \right| \leq \epsilon$

Expectation-maximization soft clustering

- Cada cluster é gerado por uma gaussiana

$$f_i(\mathbf{x}) = f(\mathbf{x}|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\left\{-\frac{(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)}{2}\right\}$$

- Onde:
  - $f_i(\mathbf{x})$ é a distribuição de probabilidades do cluster i
  - $\mu_i \in \mathbb{R}^d$ é a média do cluster i
  - Σ_i é a matriz de correlações do cluster i
- Cada $\mathbf{x} \in \mathbf{X}$ tem função probabilidade

$$f(\mathbf{x}) = \sum_{i=1}^{k} f_i(\mathbf{x})P(C_i) = \sum_{i=1}^{k} f(\mathbf{x}|\mu_i, \Sigma_i)P(C_i)$$

Expectation-maximization soft clustering

- Objetivo: encontrar parâmetros $\theta = \{\mu_i, \Sigma_i, P(C_i), \ldots\}$
- Supondo que pontos $\mathbf{x}_j \in \mathbf{D}$ são n amostras independentes de $\mathbf{X}$ a verossimilhança é definida como:

$$P(D|\theta) = \prod_{j=1}^{n} f(\mathbf{x}_j)$$

- Para encontrar θ, buscamos pela máxima log-verossimilhança:

$$\theta^* = \arg\max_{\theta}\{\log P(\mathbf{D}|\theta)\}$$

- onde

$$\log P(\mathbf{D}|\theta) = \sum_{j=1}^{n} \log \left(\sum_{i=1}^{k} f(\mathbf{x}_j|\mu_i, \Sigma_i) P(C_i) \right)$$

Maximização da verossimilhança - passo expectation

- Assumindo θ atual correto, atualizamos $P(C_i|\mathbf{x}_j)$
- Usando o teorema de Bayes

$$P(C_i|\mathbf{x}_j) \leftarrow \frac{P(C_i \text{ e } \mathbf{x}_j)}{P(\mathbf{x}_j)} = \frac{P(\mathbf{x}_j|C_i)P(C_i)}{\sum_{a=1}^{k} P(\mathbf{x}_j|C_a)P(C_a)}$$

- Sendo que $P(\mathbf{x}_j|C_i)$ pode ser aproximado por

$$\approx 2\epsilon \cdot f(\mathbf{x}_j|\mu_i, \Sigma_i) = 2\epsilon \cdot f_i(\mathbf{x}_j)$$

  para um $\epsilon > 0$ pequeno
- $P(C_i|\mathbf{x}_j)$ é o "peso" de $\mathbf{x}_j$ para grupo C_i

Maximização da verossimilhança - passo maximization

- Usar "pesos" $P(C_i|\mathbf{x}_j)$ para reestimar parâmetros θ:

  $\mu_i \leftarrow \frac{1}{n}\sum_{j=1}^{n} \mathbf{x}_j \cdot P(C_i|\mathbf{x}_j)$

  $\Sigma_i \leftarrow$ "correlação entre cada par de dimensões de $P(C_i|\mathbf{x}_j) \cdot \mathbf{x}_j$

  - Se tem poucos dados em comparação ao número de dimensões assumir que dimensões não são correlacionadas

  $P(\mathbf{x}_j|C_i) \leftarrow$ "proporção entre soma de pesos para C_i e soma total de pesos"

ALGORITHM 13.3. Expectation-Maximization (EM) Algorithm

EXPECTATION-MAXIMIZATION $(\mathbf{D}, k, \epsilon)$:

$t \leftarrow 0$

// Initialization

Randomly initialize $\boldsymbol{\mu}_1^t, \ldots, \boldsymbol{\mu}_k^t$

$\boldsymbol{\Sigma}_i^t \leftarrow \mathbf{I}, \; \forall i = 1, \ldots, k$

$P^t(C_i) \leftarrow \frac{1}{k}, \; \forall i = 1, \ldots, k$

repeat

  $t \leftarrow t + 1$

  // Expectation Step

  for $i = 1, \ldots, k$ *and* $j = 1, \ldots, n$ **do**

    $w_{ij} \leftarrow \frac{f(\mathbf{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \cdot P(C_i)}{\sum_{a=1}^{k} f(\mathbf{x}_j | \boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a) \cdot P(C_a)}$ // posterior probability $P^t(C_i | \mathbf{x}_j)$

  // Maximization Step

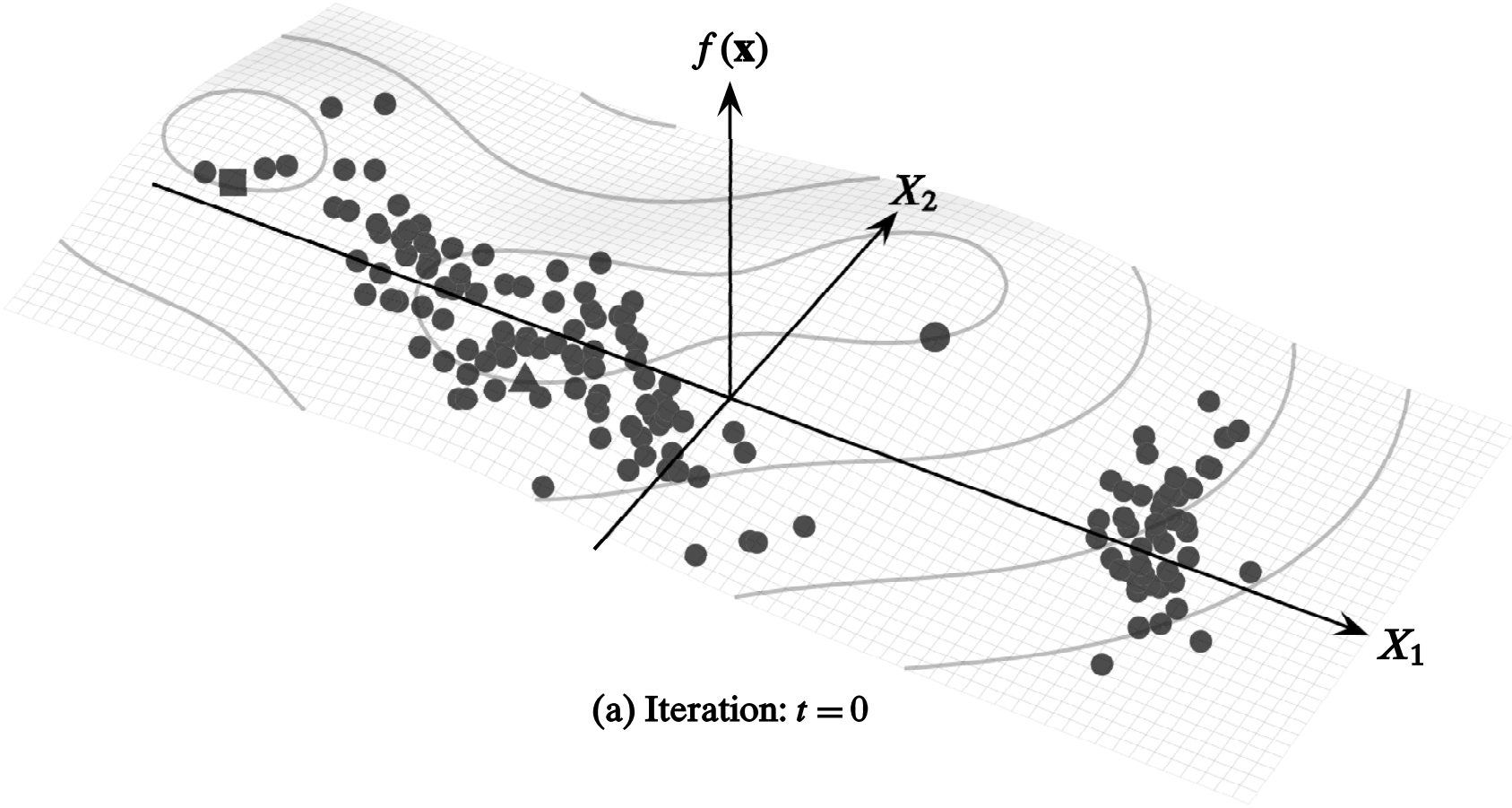
  for $i = 1, \ldots, k$ **do**

    $\boldsymbol{\mu}_i^t \leftarrow \frac{\sum_{j=1}^{n} w_{ij} \cdot \mathbf{x}_j}{\sum_{j=1}^{n} w_{ij}}$ // re-estimate mean

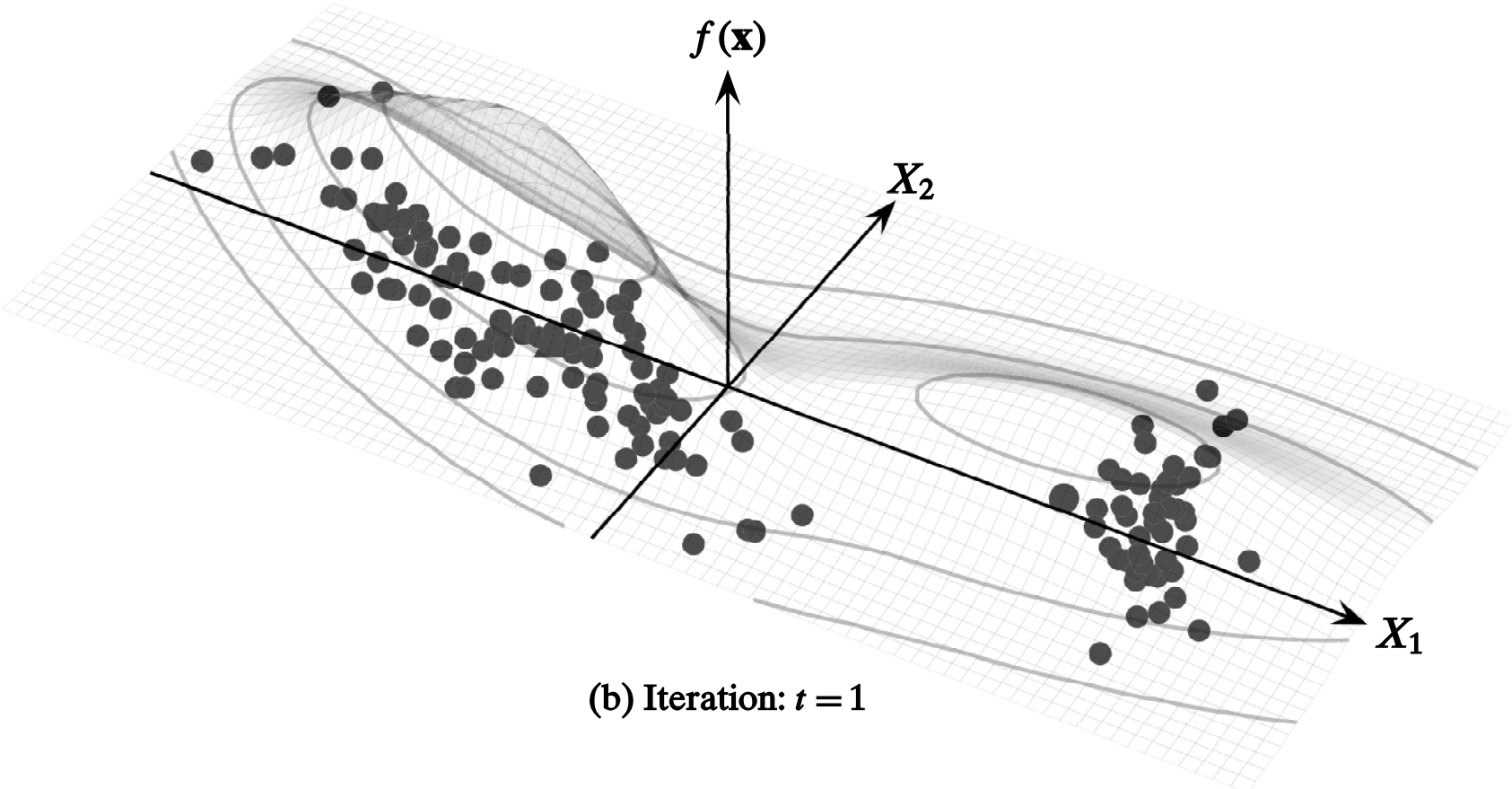
    $\boldsymbol{\Sigma}_i^t \leftarrow \frac{\sum_{j=1}^{n} w_{ij} (\mathbf{x}_j - \boldsymbol{\mu}_i)(\mathbf{x}_j - \boldsymbol{\mu}_i)^T}{\sum_{j=1}^{n} w_{ij}}$ // re-estimate covariance matrix

    $P^t(C_i) \leftarrow \frac{\sum_{j=1}^{n} w_{ij}}{n}$ // re-estimate priors

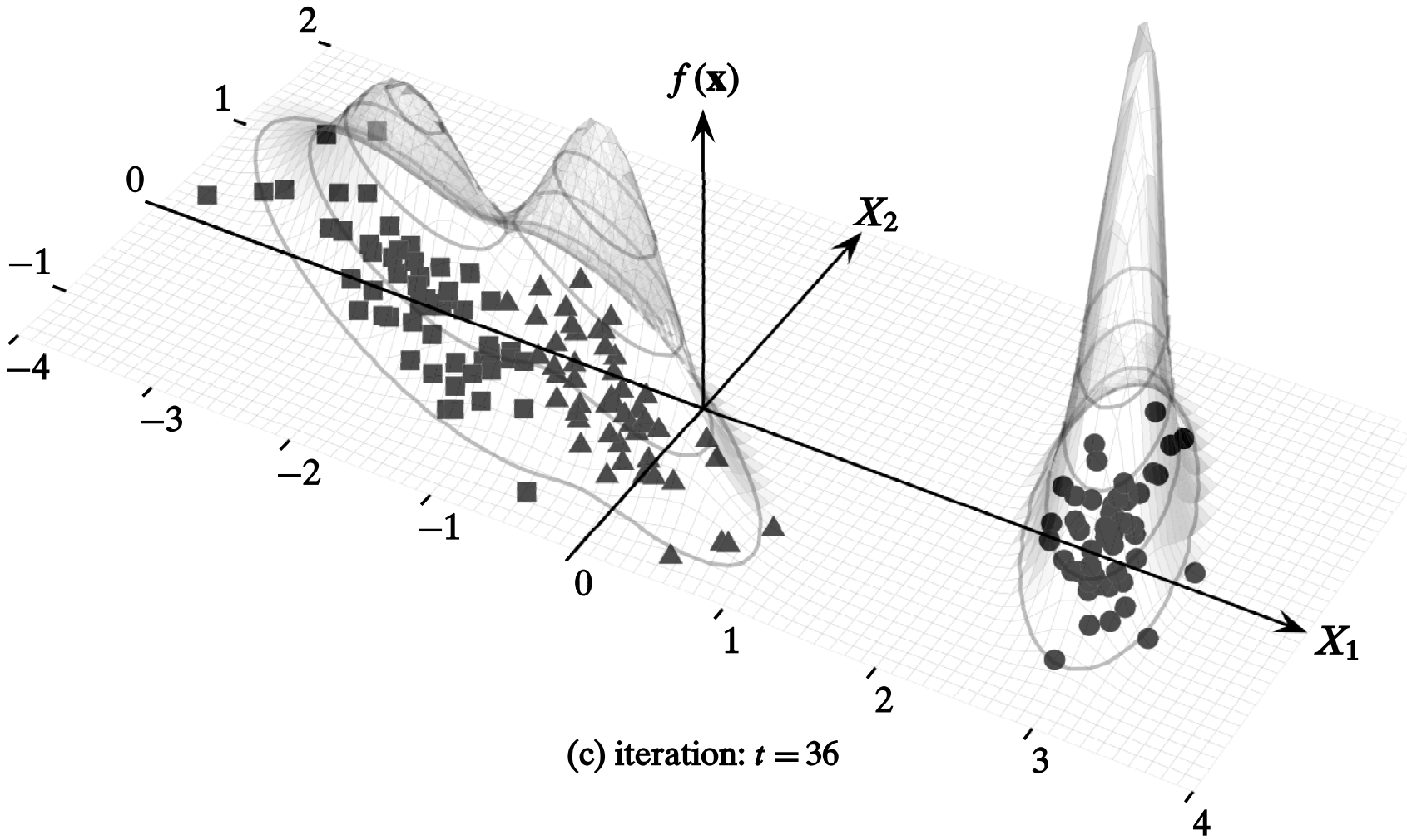
until $\sum_{i=1}^{k} \left\| \boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1} \right\|^2 \leq \epsilon$



(a) Iteration: $t = 0$



(b) Iteration: $t = 1$



(c) iteration: $t = 36$

k-means é um tipo de soft clustering EM

$$P(\mathbf{x}_j|C_i) = \begin{cases} 1 & \text{se } C_i = \arg\min_{C_a} \left\{ \|\mathbf{x}_j - \mu_a\|^2 \right\} \\ 0 & \text{caso contrário} \end{cases}$$

- Como em EM temos

$$P(C_i|\mathbf{x}_j) = \frac{P(\mathbf{x}_j|C_i)P(C_i)}{\sum_{a=1}^{k} P(\mathbf{x}_j|C_a)P(C_a)}$$

- Para k-means temos

$$P(C_i|\mathbf{x}_j) = \begin{cases} 1 & \text{se } \mathbf{x}_j \in C_i, \text{ i.e. se } C_i = \arg\min_{C_a} \{\|\mathbf{x}_j - \mu_a\|^2\} \\ 0 & \text{caso contrário} \end{cases}$$

Referências

- Xing et al. (DOI: 10.1093/bioinformatics/17.suppl_1.S306)
- Apresentação de Aarti Singh (2010)
- A Tutorial on Spectral Clustering, U. Luxburg (2007)
- Slides de Eric Xing, M. Hein e U. V. Luxburg
- `http://www.ml.uni-saarland.de/code/GraphDemo/DemoSpectralClustering.htm`
- `http://www.ml.uni-saarland.de/code/pSpectralClustering/pSpectralClustering.htm`
- `http://www.cs.berkeley.edu/%7Emalik/papers/SM-ncut.pdf`
- Weighted Graph Cuts without Eigenvectors vectors: A Multilevel Approach, Dhillon et al. (2007)