Krótkie wprowadzenie do spektralnej analizy skupień
Duże zbiory danych

Sławomir T. Wierzchoń
http://www.ipipan.waw.pl/~stw/sem190218/ML_PG.pdf

Instytut Podstaw Informatyki PAN

Seminarium „Machine Learning Gdańsk"
Gdańsk, 19 lutego 2018

Treść

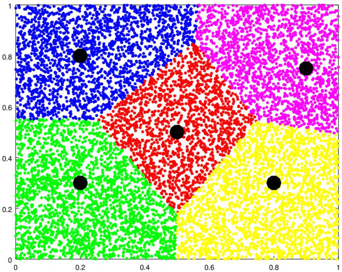
Literatura

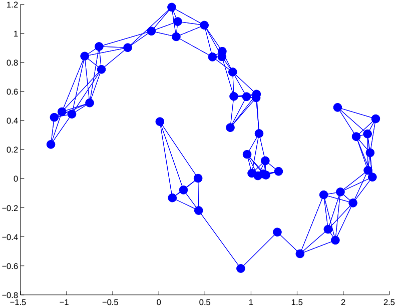
- strona: Applications of spectral graph theory
- U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395-416, 2007
- H. Jia, *et al*. The latest research progress on spectral clustering. *Neural Comput. & Applic.*, 24(7-8):1477-1486, 2014
- M. Filippone, *et al*. A survey on spectral and kernel methods for clustering. *Pattern Recognition*, 41(1):176-190, 2008
- F. Chung. *Spectral Graph Theory*. AMS, Providence, RI, 1997
- S.T. Wierzchoń, M.A. Kłopotek. Modern Algorithms of Cluster Analysis, Springer 2018

Szybkie intro

- $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m)^{\mathrm{T}} \in \mathbb{R}^{m \times n}$ – macierz danych.
- $S \in \mathbb{R}^{m \times m}$ – macierz podobieństwa; $s_{ij} = \mathfrak{s}(\mathbf{x}_i, \mathbf{x}_j)$, np. $s_{ij} = \exp\left(\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2\right)$, przy czym $s_{ij} = 0$ gdy $\exp\left(\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2\right) \leqslant \tau$.
- Podsumowaniem jest nieskierowany, ważony graf $\Gamma = (V, E, S)$, gdzie $V = \{v_1, \ldots, v_m) = \{1, \ldots, m\}$, $\{i, j\} \in E$ gdy $s_{ij} > 0$.

Skupienie – różne spojrzenia





Rysunek: Spojrzenie „globalne”: skupienia to zwarte i dobrze separowalne grupy obiektów vs spojrzenie „lokalne”: skupienie to zbiór wzajemnie podobnych obiektów.

Szybkie intro: kryteria rozcinania grafu

Niech $\text{cut}(A, B) = \sum_{i \in A} \sum_{j \in B} s_{ij}$. Wówczas

$$cut(C_1, \ldots, C_k) = \sum_{j=1}^{k} \text{cut}(C_j, C_j^c)$$

$$RCut(C_1, \ldots, C_k) = \sum_{j=1}^{k} \frac{\text{cut}(C_j, C_j^c)}{|C_j|}$$

$$NCut(C_1, \ldots, C_k) = \sum_{j=1}^{k} \frac{\text{cut}(C_j, C_j^c)}{vol(C_j)}$$

$$\text{gdzie } C_j^c = V \backslash C_j, \quad vol(C_j) = \sum_{i \in C_j} \mathrm{d}_i$$

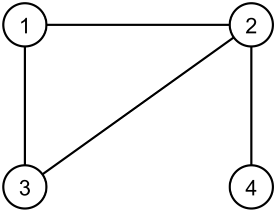
Szybkie intro: Laplasjan kombinatoryczny

- $L = D - S$, gdzie $D = \mathrm{diag}(S\mathbf{e}) = \mathrm{diag}(\mathrm{d}_1, \ldots, \mathrm{d}_m)$.
- L symetryczna i dodatnio pół-określona, tzn. $\mathbf{x}^{\mathrm{T}} L \mathbf{x} = \frac{1}{2}\sum_{i,j=1}^{m} s_{ij}(x_i - x_j)^2$
- $L = U \Lambda U^{\mathrm{T}}$, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$, $U = (\mathbf{u}_1, \ldots, \mathbf{u}_m)$, oraz $0 = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_m$. $\mathbf{u}_1 = c\mathbf{e} \Rightarrow \sum_j u_{j2} = 0$.
- (Γ spójny) $\Rightarrow \lambda_2 > 0$. Jeżeli $\Gamma' = (V, E')$, $E' \subset E$ to $\lambda_2(\Gamma') \leqslant \lambda_2(\Gamma)$

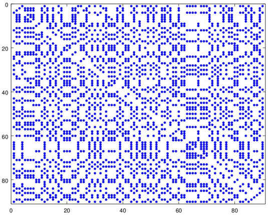
Przykład: Graf i jego laplasjan

$$S = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \ \mathbf{d} = \begin{bmatrix} 2 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

$$L = D - S = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$



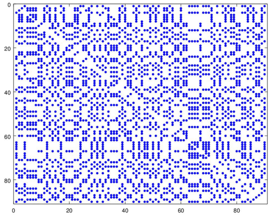
Przykład: Porządek Fiedlera



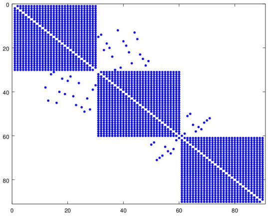
(a) (b)

Rysunek: macierz L: (a) oryginalna, (b) po posortowaniu względem wartości $\mathbf{u}_2$

Przykład: Porządek Fiedlera



(a)



(b)

Rysunek: macierz L: (a) oryginalna, (b) po posortowaniu względem wartości $\mathbf{u}_2$

Szybkie intro: normalizowany Laplasjan

- $\mathcal{L} = D^{-1/2}LD^{-1/2} = \mathbb{I} - D^{-1/2}SD^{-1/2}$.
- $\mathcal{L}$ symetryczna i dodatnio pół-określona, tzn. $\mathbf{x}^{\mathrm{T}}\mathcal{L}\mathbf{x} = \frac{1}{2}\sum_{i,j=1}^{m} s_{ij}(x_i/\sqrt{\mathrm{d}_i} - x_j/\sqrt{\mathrm{d}_j})^2$
- $\mathcal{L} = U\Lambda U^{\mathrm{T}}$, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$, $U = (\mathbf{u}_1, \ldots, \mathbf{u}_m)$, oraz $0 = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_m \leqslant 2$. $\mathbf{u}_1 = D^{1/2}\mathbf{e}$. $\lambda_m = 2$ w.t.w. Γ – graf dwudzielny.

Szybkie intro: więcej o *RCut* (1)

Niech $H = (\mathbf{h}_1, \dots, \mathbf{h}_k)$ będzie macierzą taką, że

$$h_{ij} = \begin{cases} 1/\sqrt{|C_j|} & v_i \in C_j \\ 0 & wp.p. \end{cases}, \quad i = 1, \dots, m, j = 1, \dots, k$$

Mamy $\|\mathbf{h}_j\| = \mathbf{h}_j^{\mathrm{T}}\mathbf{h}_j = 1$, $\mathbf{h}_i^{\mathrm{T}}\mathbf{h}_j = 0$, $i \neq j$ tzn. $H^{\mathrm{T}}H = \mathbb{I}$.

$$\mathbf{h}_j^{\mathrm{T}} L \mathbf{h}_j = \sum_{\alpha,\beta} s_{\alpha,\beta}(h_{\alpha j} - h_{\beta j})^2 = \frac{\mathrm{cut}(C_j, C_j^c)}{|C_j|} = (H^{\mathrm{T}}LH)_{jj}$$

$$RCut(C_1, \dots, C_k) = \sum_{j=1}^{k} \mathbf{h}_j^{\mathrm{T}} L \mathbf{h}_j = \sum_{j=1}^{k} (H^{\mathrm{T}}LH)_{jj} = \mathtt{tr}(H^{\mathrm{T}}LH)$$

Szybkie intro: więcej o *RCut* (2)

$$\min_{(C_1,\dots,C_k)\in\mathcal{C}^k} \mathtt{tr}(H^{\mathrm{T}}LH), \text{ gdzie } H \text{ jak wyżej}$$

Relaksacja:

$$\min_{\substack{H\in\mathbb{R}^{m\times k}\\ H^{\mathrm{T}}H=\mathbb{I}}} \mathtt{tr}(H^{\mathrm{T}}LH)$$

Na mocy tw. Fan'a (1947), por. (Mirzal & Furukawa, 2010)

$$\min_{\substack{H\in\mathbb{R}^{n\times k}\\ H^{\mathrm{T}}H=\mathbb{I}}} \mathtt{tr}(H^{\mathrm{T}}LH) = \sum_{j=1}^{k}\lambda_j \quad \text{jeżeli } H = (\mathbf{u}_1,\dots,\mathbf{u}_k)R$$

gdzie $\mathbf{u}_i$ - wektor własny odpowiadający i-tej minimalnej w.w. λ_i laplasjanu L, oraz $R \in \mathbb{C}^{k\times k}$ – macierz unitarna.

Szybkie intro: więcej o *NCut* (1)

Niech $H = (\mathbf{h}_1, \dots, \mathbf{h}_k)$ będzie macierzą taką, że

$$h_{ij} = \begin{cases} 1/\sqrt{vol(C_j)} & v_i \in C_j \\ 0 & wp.p. \end{cases}, \quad i = 1, \dots, m, j = 1, \dots, k$$

Mamy $\mathbf{h}_j^{\mathrm{T}} D\mathbf{h}_j = 1$, $\mathbf{h}_i^{\mathrm{T}}\mathbf{h}_j = 0$, $i \neq j$ tzn. $H^{\mathrm{T}} DH = \mathbb{I}$.

$$\mathbf{h}_j^{\mathrm{T}} L\mathbf{h}_j = \sum_{\alpha,\beta} s_{\alpha,\beta}(h_{\alpha j} - h_{\beta j})^2 = \frac{\mathrm{cut}(C_j, C_j^c)}{vol(C_j)} = (H^{\mathrm{T}} LH)_{jj}$$

$$NCut(C_1, \dots, C_k) = \sum_{j=1}^{k} \mathbf{h}_j^{\mathrm{T}} L\mathbf{h}_j = \sum_{j=1}^{k} (H^{\mathrm{T}} LH)_{jj} = \mathtt{tr}(H^{\mathrm{T}} LH)$$

Szybkie intro: więcej o *Ncut* (2)

Relaksacja zadania NCut przyjmuje postać

$$\min_{\substack{H \in \mathbb{R}^{n \times k} \\ H^{\mathrm{T}} D H = \mathbb{I}}} \mathtt{tr}(H^{\mathrm{T}} L H)$$

Niech $Q = D^{1/2} H$. Wówczas

$$\min_{\substack{Q \in \mathbb{R}^{n \times k} \\ Q^{\mathrm{T}} Q = \mathbb{I}}} \mathtt{tr}\left[Q^{\mathrm{T}} (D^{-1/2} L D^{-1/2}) Q\right] = \min_{\substack{Q \in \mathbb{R}^{n \times k} \\ Q^{\mathrm{T}} Q = \mathbb{I}}} \mathtt{tr}\left(Q^{\mathrm{T}} \mathcal{L} Q\right)$$

i można skorzystać z tw. Fan'a (1949), tzn. rozwiązaniem jest $Q = (\mathbf{u}_1, \ldots, \mathbf{u}_k) R$, gdzie $\mathbf{u}_i$ to i-ty wektor własny macierzy $\mathcal{L} = D^{-1/2} L D^{-1/2}$.

Algorytm NJW (Ng, Jordan & Weiss, 2002) – skąd taki pomysł?

Dane są 3 rozłączne grupy C_1, C_2, C_3

$$S = \begin{bmatrix} S^{(1)} & 0 & 0 \\ 0 & S^{(2)} & 0 \\ 0 & 0 & S^{(3)} \end{bmatrix} \quad \Rightarrow \quad \mathcal{L} = \begin{bmatrix} \mathcal{L}^{(1)} & 0 & 0 \\ 0 & \mathcal{L}^{(2)} & 0 \\ 0 & 0 & \mathcal{L}^{(3)} \end{bmatrix}$$

Dominujące wektory własne

$$Y = \begin{bmatrix} y^{(1)} & 0 & 0 \\ 0 & y^{(2)} & 0 \\ 0 & 0 & y^{(3)} \end{bmatrix}, y^{(i)} = \sqrt{D^{(i)}}\mathbf{e}_{C_i} \Rightarrow Z = \begin{bmatrix} \mathbf{e}_{C_1} & 0 & 0 \\ 0 & \mathbf{e}_{C_2} & 0 \\ 0 & 0 & \mathbf{e}_{C_3} \end{bmatrix}$$

Algorytm NJW (Ng, Jordan & Weiss, 2002) – skąd taki pomysł?

Dane są 3 rozłączne grupy C_1, C_2, C_3

$$S = \begin{bmatrix} S^{(1)} & 0 & 0 \\ 0 & S^{(2)} & 0 \\ 0 & 0 & S^{(3)} \end{bmatrix} \quad \Rightarrow \quad \mathcal{L} = \begin{bmatrix} \mathcal{L}^{(1)} & 0 & 0 \\ 0 & \mathcal{L}^{(2)} & 0 \\ 0 & 0 & \mathcal{L}^{(3)} \end{bmatrix}$$

Dominujące wektory własne

$$Y = \begin{bmatrix} y^{(1)} & 0 & 0 \\ 0 & y^{(2)} & 0 \\ 0 & 0 & y^{(3)} \end{bmatrix}, y^{(i)} = \sqrt{D^{(i)}}\mathbf{e}_{C_i} \Rightarrow Z = \begin{bmatrix} \mathbf{e}_{C_1} & 0 & 0 \\ 0 & \mathbf{e}_{C_2} & 0 \\ 0 & 0 & \mathbf{e}_{C_3} \end{bmatrix}$$

Algorytm NJW (Ng, Jordan & Weiss, 2002) – skąd taki pomysł?

Dane są 3 rozłączne grupy C_1, C_2, C_3

$$S = \begin{bmatrix} S^{(1)} & 0 & 0 \\ 0 & S^{(2)} & 0 \\ 0 & 0 & S^{(3)} \end{bmatrix} \quad \Rightarrow \quad \mathcal{L} = \begin{bmatrix} \mathcal{L}^{(1)} & 0 & 0 \\ 0 & \mathcal{L}^{(2)} & 0 \\ 0 & 0 & \mathcal{L}^{(3)} \end{bmatrix}$$

Dominujące wektory własne

$$Y = \begin{bmatrix} y^{(1)} & 0 & 0 \\ 0 & y^{(2)} & 0 \\ 0 & 0 & y^{(3)} \end{bmatrix}, y^{(i)} = \sqrt{D^{(i)}}\mathbf{e}_{C_i} \Rightarrow Z = \begin{bmatrix} \mathbf{e}_{C_1} & 0 & 0 \\ 0 & \mathbf{e}_{C_2} & 0 \\ 0 & 0 & \mathbf{e}_{C_3} \end{bmatrix}$$

Algorytm NJW (Ng, Jordan & Weiss, 2002) – skąd taki pomysł?

Dane są 3 rozłączne grupy C_1, C_2, C_3

$$S = \begin{bmatrix} S^{(1)} & 0 & 0 \\ 0 & S^{(2)} & 0 \\ 0 & 0 & S^{(3)} \end{bmatrix} \quad \Rightarrow \quad \mathcal{L} = \begin{bmatrix} \mathcal{L}^{(1)} & 0 & 0 \\ 0 & \mathcal{L}^{(2)} & 0 \\ 0 & 0 & \mathcal{L}^{(3)} \end{bmatrix}$$

Dominujące wektory własne

$$Y = \begin{bmatrix} y^{(1)} & 0 & 0 \\ 0 & y^{(2)} & 0 \\ 0 & 0 & y^{(3)} \end{bmatrix}, y^{(i)} = \sqrt{D^{(i)}}\mathbf{e}_{C_i} \Rightarrow Z = \begin{bmatrix} \mathbf{e}_{C_1} & 0 & 0 \\ 0 & \mathbf{e}_{C_2} & 0 \\ 0 & 0 & \mathbf{e}_{C_3} \end{bmatrix}$$

Algorytm NJW

WE: symetryczna macierz S taka, że $s_{ii} = 0$

1. wyznacz $\mathcal{S} = D^{-1/2} S D^{-1/2}$,
   $D = \mathrm{diag}(S\mathbf{e}) = \mathrm{diag}(\mathrm{d}_1, \ldots, \mathrm{d}_m)$
2. wyznacz $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$; $\mathbf{y}_j$ – j-ty dominujący wektor własny $\mathcal{S}$
3. rzutuj wiersze macierzy Y na sferę jednostkową, tzn. utwórz macierz $Z = [z_{ij}]$, $i = 1, \ldots, m$, $j = 1, \ldots, k$ o elementach

$$z_{ij} = y_{i,j} / \|\mathbf{y}_{i\cdot}\|$$

4. wyznacz finalny podział danych reprezentowanych przez macierz Z.

Algorytm NJW

WE: symetryczna macierz S taka, że $s_{ii} = 0$

1. wyznacz $\mathcal{S} = D^{-1/2} S D^{-1/2}$,
   $D = \mathrm{diag}(S\mathbf{e}) = \mathrm{diag}(\mathsf{d}_1, \ldots, \mathsf{d}_m)$
2. wyznacz $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$; $\mathbf{y}_j$ – j-ty dominujący wektor własny $\mathcal{S}$
3. rzutuj wiersze macierzy Y na sferę jednostkową, tzn. utwórz macierz $Z = [z_{ij}]$, $i = 1, \ldots, m$, $j = 1, \ldots, k$ o elementach

$$z_{ij} = y_{i,j} / \|\mathbf{y}_{i\cdot}\|$$

4. wyznacz finalny podział danych reprezentowanych przez macierz Z.

Algorytm NJW

WE: symetryczna macierz S taka, że $s_{ii} = 0$

1. wyznacz $\mathcal{S} = D^{-1/2} S D^{-1/2}$,
   $D = \mathrm{diag}(S\mathbf{e}) = \mathrm{diag}(\mathrm{d}_1, \ldots, \mathrm{d}_m)$
2. wyznacz $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$; $\mathbf{y}_j$ – j-ty dominujący wektor własny $\mathcal{S}$
3. rzutuj wiersze macierzy Y na sferę jednostkową, tzn. utwórz macierz $Z = [z_{ij}]$, $i = 1, \ldots, m$, $j = 1, \ldots, k$ o elementach

$$z_{ij} = y_{i,j} / \|\mathbf{y}_{i\cdot}\|$$

4. wyznacz finalny podział danych reprezentowanych przez macierz Z.

Algorytm NJW

WE: symetryczna macierz S taka, że $s_{ii} = 0$

1. wyznacz $\mathcal{S} = D^{-1/2} S D^{-1/2}$,
   $D = \mathrm{diag}(S\mathbf{e}) = \mathrm{diag}(\mathrm{d}_1, \ldots, \mathrm{d}_m)$
2. wyznacz $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$; $\mathbf{y}_j$ – j-ty dominujący wektor własny $\mathcal{S}$
3. rzutuj wiersze macierzy Y na sferę jednostkową, tzn. utwórz macierz $Z = [z_{ij}]$, $i = 1, \ldots, m$, $j = 1, \ldots, k$ o elementach

$$z_{ij} = y_{i,j} / \|\mathbf{y}_{i\cdot}\|$$

4. wyznacz finalny podział danych reprezentowanych przez macierz Z.

Algorytm NJW

WE: symetryczna macierz S taka, że $s_{ii} = 0$

1. wyznacz $\mathcal{S} = D^{-1/2} S D^{-1/2}$,
   $D = \mathrm{diag}(S\mathbf{e}) = \mathrm{diag}(\mathsf{d}_1, \ldots, \mathsf{d}_m)$
2. wyznacz $Y = (\mathbf{y}_1, \ldots, \mathbf{y}_k)$; $\mathbf{y}_j$ – j-ty dominujący wektor własny $\mathcal{S}$
3. rzutuj wiersze macierzy Y na sferę jednostkową, tzn. utwórz macierz $Z = [z_{ij}]$, $i = 1, \ldots, m$, $j = 1, \ldots, k$ o elementach

$$z_{ij} = y_{i,j} / \|\mathbf{y}_{i\cdot}\|$$

4. wyznacz finalny podział danych reprezentowanych przez macierz Z.

Algorytm NJW - wyznaczanie finalnego podziału danych

Zastosuj:

1. algorytm k-średnich,
2. algorytm rotacji spektralnej (Huang, Nie, & Huang, 2013): $\min_{Q,H} \|YQ - H\|_F$ p.o. $Q^T Q = \mathbb{I}$, $H \in \{0,1\}^{m \times k}$,
3. QR faktoryzację (Zha, He, Ding, Gu, & Simon, 2001), (Freder, & van Barel, 2010), (Damle, Minden, & Ying)
   [Q,R,P]=qr(Y^T); %$Q^T Q = \mathbb{I}$, P – macierz permut.
   u=inv(R(1:K,1:K))*R*P';
   [tmp,r1]=max(abs(u^T),[],2);

Dla $Z \in \mathbb{R}^{2000\times 3}$: wariant (1) trwa 0.03737 ± 0.00958, a randomizowany wariat (3): $10^{-4}(4.8034 \pm 5.0504) \approx 0.001067$ (średnie z 50 powtórzeń).

Algorytm NJW - wyznaczanie finalnego podziału danych

Zastosuj:

1. algorytm k-średnich,
2. algorytm rotacji spektralnej (Huang, Nie, & Huang, 2013): $\min_{Q,H} \|YQ - H\|_F$ p.o. $Q^{\mathrm{T}}Q = \mathbb{I}$, $H \in \{0,1\}^{m \times k}$,
3. QR faktoryzację (Zha, He, Ding, Gu, & Simon, 2001), (Freder, & van Barel, 2010), (Damle, Minden, & Ying)
   [Q,R,P]=qr(Y^{T}); %$Q^{\mathrm{T}}Q = \mathbb{I}$, P – macierz permut.
   u=inv(R(1:K,1:K))*R*P';
   [tmp,r1]=max(abs(u^{T}),[],2);

Dla $Z \in \mathbb{R}^{2000\times 3}$: wariant (1) trwa 0.03737 ± 0.00958, a randomizowany wariat (3): $10^{-4}(4.8034 \pm 5.0504) \approx 0.001067$ (średnie z 50 powtórzeń).

Algorytm NJW - wyznaczanie finalnego podziału danych

Zastosuj:

1. algorytm k-średnich,
2. algorytm rotacji spektralnej (Huang, Nie, & Huang, 2013): $\min_{Q,H} \|YQ - H\|_F$ p.o. $Q^{\mathrm{T}}Q = \mathbb{I}$, $H \in \{0,1\}^{m\times k}$,
3. QR faktoryzację (Zha, He, Ding, Gu, & Simon, 2001), (Freder, & van Barel, 2010), (Damle, Minden, & Ying) [Q,R,P]=qr(Y$^{\mathrm{T}}$); %$Q^{\mathrm{T}}Q = \mathbb{I}$, P – macierz permut. u=inv(R(1:K,1:K))*R*P'; [tmp,r1]=max(abs(u$^{\mathrm{T}}$),[],2);

Dla $Z \in \mathbb{R}^{2000\times 3}$: wariant (1) trwa 0.03737 ± 0.00958, a randomizowany wariat (3): $10^{-4}(4.8034 \pm 5.0504) \approx 0.001067$ (średnie z 50 powtórzeń).

Algorytm NJW - wyznaczanie finalnego podziału danych

Zastosuj:

1. algorytm k-średnich,
2. algorytm rotacji spektralnej (Huang, Nie, & Huang, 2013): $\min_{Q,H} \|YQ - H\|_F$ p.o. $Q^T Q = \mathbb{I}$, $H \in \{0,1\}^{m\times k}$,
3. QR faktoryzację (Zha, He, Ding, Gu, & Simon, 2001), (Freder, & van Barel, 2010), (Damle, Minden, & Ying)
   [Q,R,P]=qr(Y^T); %$Q^T Q = \mathbb{I}$, P – macierz permut.
   u=inv(R(1:K,1:K))*R*P';
   [tmp,r1]=max(abs(u^T),[],2);

Dla $Z \in \mathbb{R}^{2000\times 3}$: wariant (1) trwa 0.03737 ± 0.00958, a randomizowany wariat (3): $10^{-4}(4.8034 \pm 5.0504) \approx 0.001067$ (średnie z 50 powtórzeń).

Algorytm NJW - wyznaczanie finalnego podziału danych

Zastosuj:

1. algorytm k-średnich,
2. algorytm rotacji spektralnej (Huang, Nie, & Huang, 2013): $\min_{Q,H} \|YQ - H\|_F$ p.o. $Q^{\mathrm{T}}Q = \mathbb{I}$, $H \in \{0,1\}^{m \times k}$,
3. QR faktoryzację (Zha, He, Ding, Gu, & Simon, 2001), (Freder, & van Barel, 2010), (Damle, Minden, & Ying)
   [Q,R,P]=qr(Y^{T}); %$Q^{\mathrm{T}}Q = \mathbb{I}$, P – macierz permut.
   u=inv(R(1:K,1:K))*R*P';
   [tmp,r1]=max(abs(u^{T}),[],2);

Dla $Z \in \mathbb{R}^{2000\times 3}$: wariant (1) trwa 0.03737 ± 0.00958, a randomizowany wariat (3): $10^{-4}(4.8034 \pm 5.0504) \approx 0.001067$ (średnie z 50 powtórzeń).

Algorytm NJW - wyznaczanie finalnego podziału danych

Zastosuj:

1. algorytm k-średnich,
2. algorytm rotacji spektralnej (Huang, Nie, & Huang, 2013): $\min_{Q,H} \|YQ - H\|_F$ p.o. $Q^T Q = \mathbb{I}$, $H \in \{0,1\}^{m\times k}$,
3. QR faktoryzację (Zha, He, Ding, Gu, & Simon, 2001), (Freder, & van Barel, 2010), (Damle, Minden, & Ying)
   [Q,R,P]=qr(Y^T); %$Q^T Q = \mathbb{I}$, P – macierz permut.
   u=inv(R(1:K,1:K))*R*P';
   [tmp,r1]=max(abs(u^T),[],2);

Dla $Z \in \mathbb{R}^{2000\times 3}$: wariant (1) trwa 0.03737 ± 0.00958, a randomizowany wariat (3): $10^{-4}(4.8034 \pm 5.0504) \approx 0.001067$ (średnie z 50 powtórzeń).

Wyzwania podejścia spektralnego

- Złożoność pamięciowa: $O(m^2)$,
- Złożoność czasowa $O(m^3)$
- Konieczność reklasyfikacji
- Jak klasyfikować nowe obserwacje?

Wyzwania podejścia spektralnego

- Złożoność pamięciowa: $O(m^2)$,
- Złożoność czasowa $O(m^3)$
- Konieczność reklasyfikacji
- Jak klasyfikować nowe obserwacje?

Wyzwania podejścia spektralnego

- Złożoność pamięciowa: $O(m^2)$,
- Złożoność czasowa $O(m^3)$
- Konieczność reklasyfikacji
- Jak klasyfikować nowe obserwacje?

Wyzwania podejścia spektralnego

- Złożoność pamięciowa: $O(m^2)$,
- Złożoność czasowa $O(m^3)$
- Konieczność reklasyfikacji
- Jak klasyfikować nowe obserwacje?

Wyzwania podejścia spektralnego

- Złożoność pamięciowa: $O(m^2)$,
- Złożoność czasowa $O(m^3)$
- Konieczność reklasyfikacji
- Jak klasyfikować nowe obserwacje?

Nowe podejścia do pokonywania złożoności

1. **RanNLA** – Randomized Numerical Linear Algebra (Drineas, & Mahoney, 2016),(Mahoney, 2016): Randomizacja jako narzędzie wspomagające konstrukcję algorytmów o małym czasie wykonywania i lepszej stabilności. Jednym z głównych chwytów jest *sketching*: zastąpienie dużej macierzy A mniejszą, np. $\widehat{A} = A\Omega$, która reprezentuje istotne własności A.
2. **Sparsity** (wybór podzbioru obserwacji z X): (Alzate, & Suykens, 2011), (Federix, & van Barel, 2010), oraz
3. **Compressed sensing**, np. (Tremblay, Puy, Gribonval, & Vandergheynst, 2016)

Nowe podejścia do pokonywania złożoności

1. **RanNLA** – Randomized Numerical Linear Algebra (Drineas, & Mahoney, 2016),(Mahoney, 2016): Randomizacja jako narzędzie wspomagające konstrukcję algorytmów o małym czasie wykonywania i lepszej stabilności. Jednym z głównych chwytów jest *sketching*: zastąpienie dużej macierzy A mniejszą, np. $\widehat{A} = A\Omega$, która reprezentuje istotne własności A.
2. **Sparsity** (wybór podzbioru obserwacji z X): (Alzate, & Suykens, 2011), (Federix, & van Barel, 2010), oraz
3. **Compressed sensing**, np. (Tremblay, Puy, Gribonval, & Vandergheynst, 2016)

Nowe podejścia do pokonywania złożoności

1. **RanNLA** – Randomized Numerical Linear Algebra (Drineas, & Mahoney, 2016),(Mahoney, 2016): Randomizacja jako narzędzie wspomagające konstrukcję algorytmów o małym czasie wykonywania i lepszej stabilności. Jednym z głównych chwytów jest *sketching*: zastąpienie dużej macierzy A mniejszą, np. $\widehat{A} = A\Omega$, która reprezentuje istotne własności A.
2. **Sparsity** (wybór podzbioru obserwacji z X): (Alzate, & Suykens, 2011), (Federix, & van Barel, 2010), oraz
3. **Compressed sensing**, np. (Tremblay, Puy, Gribonval, & Vandergheynst, 2016)

Nowe podejścia do pokonywania złożoności

1. **RanNLA** – Randomized Numerical Linear Algebra (Drineas, & Mahoney, 2016),(Mahoney, 2016): Randomizacja jako narzędzie wspomagające konstrukcję algorytmów o małym czasie wykonywania i lepszej stabilności. Jednym z głównych chwytów jest *sketching*: zastąpienie dużej macierzy A mniejszą, np. $\widehat{A} = A\Omega$, która reprezentuje istotne własności A.
2. **Sparsity** (wybór podzbioru obserwacji z X): (Alzate, & Suykens, 2011), (Federix, & van Barel, 2010), oraz
3. **Compressed sensing**, np. (Tremblay, Puy, Gribonval, & Vandergheynst, 2016)

Sygnały na grafach – literatura

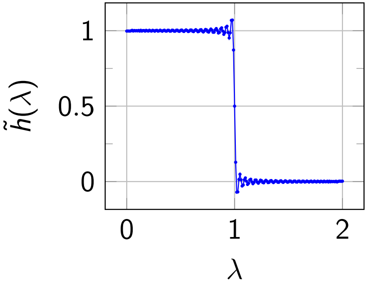
1. A. Sandryhaila, J.M.F. Moura: Big data analysis with signal processing on graphs
2. D. Shuman *et al*. The emerging field of signal processing on graphs
3. D.K. Hammond *eta al*. Wavelets on graphs via spectral graph theory
4. I. Pesenson: Sampling in Paley-Wiener spaces on combinatorial graphs

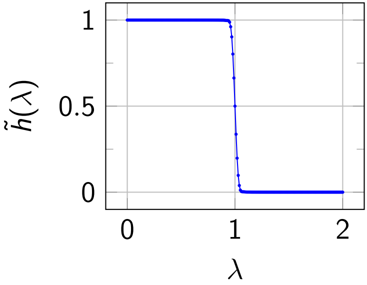
Sygnały i ich filtrowanie

- Sygnał: $\mathfrak{s}\colon V \to \mathbb{R}$; $\mathfrak{s}$ zastępujemy przez wektor $\mathbf{s} \in \mathbb{R}^m$.
- Jeżeli $h\colon [0,2] \to \mathbb{R}$, to $H = h(\mathcal{L}) = Uh(\Lambda)U^{\mathrm{T}}$ – filtr grafowy, przy czym $h(\Lambda) = \mathrm{diag}(h(\lambda_1, \ldots, h(\lambda_m))$. $H = H^{\mathrm{T}}$.
  $\hat{\mathbf{s}} = H\mathbf{s} = (\mathbf{s}^{\mathrm{T}} H)^{\mathrm{T}}$ – przefiltrowany sygnał.
- Jeżeli $h_{\lambda_k}(\lambda) = 1$ gdy $\lambda \leqslant \lambda_k$ i $h_{\lambda_k}(\lambda) = 0$ w p.p., to $H_{\lambda_k} = U_k U_k^{\mathrm{T}}$, gdzie $U_k = U(:, 1:k)$ – filtr dolnoprzepustowy.
- Przyjmując wielomianową aproksymację $h(\xi) \approx \hat{h}(\xi) = \sum_{\ell=0}^{p} \alpha_\ell \xi^\ell$. W tym przypadku

$$H\mathbf{s} \approx \tilde{H}\mathbf{s} = \sum_{\ell=0}^{p} \alpha_\ell \mathcal{L}^\ell \mathbf{s}; \quad \mathcal{L}^\ell \mathbf{s} = \mathcal{L}(\mathcal{L}^{\ell-1}\mathbf{s})$$

Filtr dolnoprzepustowy – aproksymacja





Rysunek: Aproksymacja Czebyszewa (lewy rys.) oraz Czebyszewa-Jacksona (prawy). $\tilde{H}\mathbf{s} = \sum_{\ell=0}^{p} \alpha_\ell T_\ell(\mathcal{L})$, gdzie $T_0(\mathcal{L}) = \mathbf{s}$, $T_1(\mathcal{L}) = \mathcal{L}\mathbf{s} - \mathbf{s}$, $T_\ell(\mathcal{L}) = 2\big(\mathcal{L}\, T_{\ell-1}(\mathcal{L}) - T_{\ell-1}(\mathcal{L})\big) - T_{\ell-2}(\mathcal{L})$, $\ell > 1$.

Szacowanie odległości w przestrzeni spektralnej

Niech $H_{\lambda_k} = U_k U_k^{\mathrm{T}}$, $\boldsymbol{\delta}_i$ – i-ty wiersz macierzy $\mathbb{I}$. Niech $R^{m\times\eta}$ macierz o elementach $r_{ij} \sim \mathcal{N}(0, 1/\eta)$. Wtedy

$$(H_{\lambda_k} R)^{\mathrm{T}} \boldsymbol{\delta}_i = (R^{\mathrm{T}} U_k)(U_k^{\mathrm{T}} \boldsymbol{\delta}_i) = R' \mathbf{f}_i$$

gdzie $\mathbf{f}_i = (\mathbf{y}^i)^{\mathrm{T}}$ – transponowany i-ty wiersz macierzy $Y = U_k$ (krok 2 w NJW), a $R' = R^{\mathrm{T}} U_k \in \mathbb{R}^{\eta\times m}$ macierz o elementach $r_{ij} \sim \mathcal{N}(0, 1/\eta)$.

Lemat Johnsona-Lindenstraussa(Achlioptas, 2003)

Lemma

Let P be an arbitrary set of m points in $\mathbb{R}^n$ represented as an $m \times n$ matrix A. Given $\epsilon, \beta > 0$ let $\eta_0(m, \epsilon, \beta) = \frac{4+2\beta}{\epsilon^2/2-\epsilon^3/3} \ln m$. For any $\eta \geqslant \eta_0$ let R be a $n \times \eta$ random matrix with elements

$$r_{ij} \sim \mathcal{N}(0, 1/\eta),\ r_{ij} = \begin{cases} +1 & p = 1/2 \\ -1 & p = 1/2 \end{cases},\ r_{ij} = \sqrt{\theta} \begin{cases} +1 & p = 1/(2\theta) \\ -1 & p = 1/(2\theta) \\ 0 & otherwise \end{cases}$$

Let $g \colon \mathbb{R}^n \to \mathbb{R}^\eta$ map i-th row of A to i-th row of $E = AR/\sqrt{\eta}$. Then with prob. at least $1 - m^{-\beta}$ for all $u, v \in P$

$$(1-\epsilon)\|u-v\|^2 \leqslant \|g(u)-g(v)\|^2 \leqslant (1+\epsilon)\|u-v\|^2$$

Zastosowanie lematu J-L

Przyjmując $g(\mathbf{f}_i) = R'\mathbf{f}_i = (H_{\lambda_k} R)^{\mathtt{T}} \boldsymbol{\delta}_i = \tilde{\mathbf{f}}_i$ mamy

$$\tilde{d}_{ij} = \|\tilde{\mathbf{f}}_i - \tilde{\mathbf{f}}_j\| = \|g(\mathbf{f}_i) - g(\mathbf{f}_j)\|$$

Stosując lemat J-L stwierdzamy, że dla dowolnych $\epsilon \in (0, 1]$ i $\beta > 0$, jeżeli $\eta \geqslant \eta_0(m, \epsilon, \beta)$ to z prawdopodobieństwem przynajmniej $1 - m^{-\beta}$ mamy

$$(1 - \epsilon)d_{ij} \leqslant \tilde{d}_{ij} \leqslant (1 + \epsilon)d_{ij}$$

WAŻNE: H_{λ_k} można zastąpić jego aproksymacją $\tilde{H}_{\lambda_k}$

PROBLEM

JAK WYZNACZYĆ λ_k?

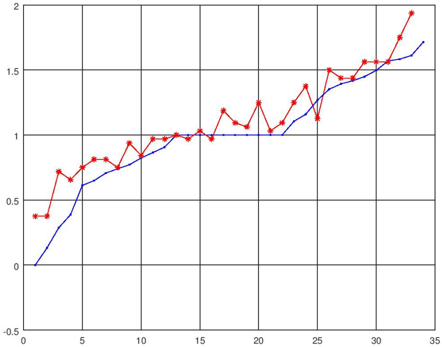
Przybliżone wyznaczanie λ_k (1)

- E. Di Napoli *et al.*: Efficient estimation of eigenvalue counts in an interval, T. Li *et al.* Fast compressive spectral clustering.
- Jeżeli A jest nieosobliwa i hermitowska, to $A = U\Lambda U^{\mathrm{T}}$ przy czym liczba dodatnich wartości własnych (w.w.) A jest równa liczbie dodatnich w.w. Λ (*Sturm count*). Liczba $\mu_{[a,b]}$ w.w. należących do przedziału $[a, b]$ jest równa różnicy miedzy liczbą dodatnich w.w. macierzy $A - b\mathbb{I}$ oraz $A - a\mathbb{I}$.
- Niech $P = \sum_{\lambda\in[a,b]} \mathbf{u}_\lambda \mathbf{u}_\lambda^{\mathrm{T}}$ (*projektor spektralny*).
  $\mu_{[a,b]} = \mathtt{tr}(P) = \frac{1}{n_v} \sum_{j=1}^{n_v} \mathbf{v}_j^{\mathrm{T}} P \mathbf{v}_j$,
  $\mu_{[a,b]} \approx \sum_{j=1}^{n_v} \sum_{i=0} \alpha_i \mathbf{v}_j^{\mathrm{T}} T_i(\mathcal{L}) \mathbf{v}_j$.

Przybliżone wyznaczanie λ_k (2)

- $\lambda_{min} = 0$, $\lambda_{max} = 2$
- **while** (counts =k)
  - $\lambda_{mid} = (\lambda_{min} + \lambda_{max})/2$
  - jch = jackson_cheby_poly_coeffs(0, λ_{mid}, 0, 2, order)
  - wyznacz counts $= \mu_{[0,\lambda_{mid}]}$
  - **if** counts $>$ k, $\lambda_{max} = \lambda_{mid}$, **else** $\lambda_{min} = \lambda_{mid}$;
- $\lambda_k = (\lambda_{min} + \lambda_{max})/2$

Przybliżone wyznaczanie λ_k (Przykład – sieć `karate`)



Rysunek: Rzeczywiste (niebieski) i przybliżone (czerwony) w.w.

Compressive spectral clustering (Tremblay et al., 2016

- WE: laplasjan $\mathcal{L}$, liczba skupień k, parametry: $\eta = 4 \log m$, $p = 50$
- Wyznacz przybliżoną wartość $\tilde{\lambda}_k$
- Wyznacz aproksymacje $\tilde{h}_{\tilde{\lambda}_k}$ oraz $\tilde{H}_{\tilde{\lambda}_k}$
- Filtruj losową macierz R, $F = (\tilde{H}_{\tilde{\lambda}_k} R)$.
- Wyznacz wektory cech $\tilde{f}_i = (F^{\mathrm{T}} \boldsymbol{\delta}_i)/\|F^{\mathrm{T}} \boldsymbol{\delta}_i\|$
- zastosuj algorytm k-średnich do zbioru $\{\tilde{f}_1, \ldots, \tilde{f}_m\}$.

Dziękuję za uwagę