Clustering

Jiali Lin

Virginia Tech

January 15, 2017

Outline

Introduction

Similarity-based clustering

- Input: an $N \times N$ **dissimilarity matrix**.
- Output: **flat clustering**, where we partition the objects into disjoint sets.
- Sensitive to the initial conditions and requires some model selection method for K.

Feature-based clustering

- Input: an $N \times D$ feature matrix.
- Output: **hierarchical clustering**, where we create a nested tree of partitions.
- Most are deterministic and do not require the specification of K.

Clustering Evaluation: Rand Index

The validation of clustering structures is the most difficult.

- The number of assumed clusters may be different.
- No true cluster labels.

Rand index computes following for all pairs of data points

$$R = \frac{TP + TN}{TP + FP + FN + TN}$$

- False positive (FP): target splits but algorithm clusters.
- False negative (FN): target clusters but algorithm splits.
- True positive (TP): algorithm and target both cluster together.
- True negative (TN): algorithm and target both split apart.

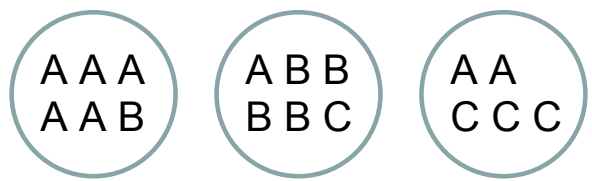


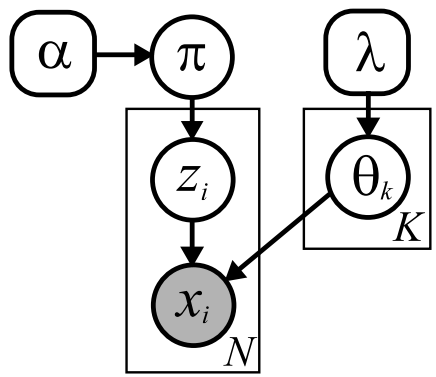
Figure: Circles are proposed clusters, letters are true cluster labels. Invariant to label choices, and takes time linear in N.

Outline

Finite mixture models

Traditional representation of a finite mixture model

$$p(\boldsymbol{x}_i|z_i = k, \boldsymbol{\theta}) = p(\boldsymbol{x}_i|\boldsymbol{\theta_k})$$
$$p(z_i = k|\boldsymbol{\pi}) = \pi_k$$
$$p(\boldsymbol{\pi}|\alpha) = \mathsf{Dir}(\boldsymbol{\pi}|(\alpha/K)\mathbf{1}_K)$$



- The form of $p(\boldsymbol{\theta}_k|\lambda)$ is chosen to be conjugate to $p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)$.
- We can write $p(\boldsymbol{x}_i|\boldsymbol{\theta}_k)$ as $\boldsymbol{x}_i \sim F(\boldsymbol{\theta}_{z_i})$, where F is the observation distribution.
- We can write $\boldsymbol{\theta}_k \sim H(\lambda)$, where H is the prior.

Another representation

Consider

$$G(\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \delta_{\boldsymbol{\theta}_k}(\boldsymbol{\theta})$$

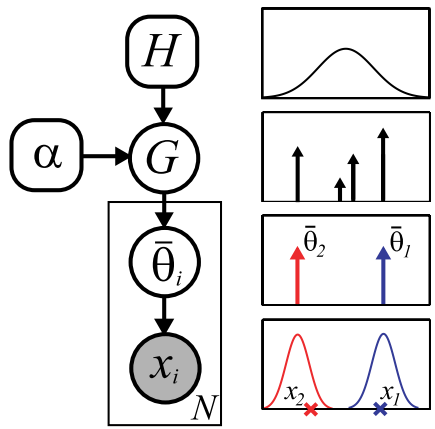


Figure: Here $\bar{\boldsymbol{\theta}}_i$ is the parameter used to generate observation $\boldsymbol{x}_i$; these parameters are sampled from distribution G, which has the form.

- The discrete measure, G is a finite mixture of delta functions, K centered on the cluster parameters $\boldsymbol{\theta}_k$.
- The probability that $\bar{\boldsymbol{\theta}}_i$ is equal to $\boldsymbol{\theta}_k$ is exactly π_k, the prior probability for that cluster.

Generative model

- Finite Gaussian mixture model ($K = 2$ clusters)

$$z_n \overset{iid}{\sim} \text{Categorical}(\rho_1, \rho_2)$$
$$\boldsymbol{x_n} \overset{indep}{\sim} N(\boldsymbol{\mu}_{z_n}, \boldsymbol{\Sigma})$$

- Don't know μ_1, μ_2

$$\boldsymbol{\mu_k} \overset{iid}{\sim} N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

- Don't know ρ_1, ρ_2

$$\rho_1 \sim \text{Beta}(a_1, a_2)$$
$$\rho_2 = 1 - \rho_1$$

- **Inference goal**: assignments of data points to clusters, cluster parameters

Generative model

- Finite Gaussian mixture model (K clusters)

$$\begin{aligned}
\boldsymbol{\rho}_{1:K} &\sim \text{Dirichlet}(a_{1:K}) \\
\boldsymbol{\mu}_k &\overset{iid}{\sim} N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \\
z_n &\overset{iid}{\sim} \text{Categorical}(\boldsymbol{\rho}_{1:K}) \\
\boldsymbol{x}_n &\overset{indep}{\sim} N(\boldsymbol{\mu}_{z_n}, \boldsymbol{\Sigma})
\end{aligned}$$

What if $K \to \infty$

- Now, we will always (with probability one) get exactly K clusters.
- Want more flexible model: generate a variable number of clusters.
- The more data we generate, the more likely to see a new cluster.
- **Solutions:** replace the discrete distribution G with a **random probability measure**.
- Recall

$$G(\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \delta_{\boldsymbol{\theta}_k}(\boldsymbol{\theta})$$

  θ_i can take on the same value θ_k for some k.

Outline

Dirichlet Distribution and Multinomial

- Consider

$$(\pi_1, \ldots, \pi_K) \sim \text{Discrete}(\alpha_1, \ldots, \alpha_K)$$
$$z|(\pi_1, \ldots, \pi_K) \sim \text{Discrete}(\pi_1, \ldots, \pi_K)$$

- Then

$$z \sim \text{Discrete}(\frac{\alpha_1}{\sum_i \alpha_i}, \ldots, \frac{\alpha_K}{\sum_i \alpha_i})$$
$$(\pi_1, \ldots, \pi_K)|z \sim \text{Discrete}(\alpha_1 + \delta_1(z), \ldots, \alpha_K + \delta_K(z))$$

where $\delta_i(z) = 1$ if z takes on value i, 0 otherwise.

Dirichlet process

- **Dirichlet process** is a distribution over probability measures $G : \Theta \to \mathbb{R}^+$.
- Require $G(\theta) \geq 0$ and $\int_\theta G(\theta) d\theta = 1$.
- For any finite partition $(T_1, \ldots, T_K)$ of Θ

$$(G(T_1), \ldots, G(T_k)) \sim \text{Dir}(\alpha H(T_1), \ldots, \alpha H(T_K))$$

- Define: $G \sim \text{DP}(\alpha, H)$, where α is called the **concentration parameter** and H is called the **base measure**.
- Intuitively, G needs to resemble with the basic distribution H.
- α determines how closely the histogram of spikes represents H.

Some properties of DP

- We are interested in

$$p(\theta) = \int p(\theta|G)p(G)dG$$

$$p(G|\theta) = \frac{p(\theta|G)p(G)}{p(\theta)}$$

- Recall **Dirichlet-multinomial conjugacy**.
- If $G \sim \mathsf{DP}(\alpha, H)$, then $p(\boldsymbol{\theta}_i \in T_i) = H(T_i)$ and the posterior is

$$(G(T_1), \ldots, G(T_k))|\boldsymbol{\theta} \sim \mathsf{Dir}(\alpha H(T_1) + \mathbb{I}(\boldsymbol{\theta} \in T_1), \ldots, \alpha H(T_K) + \mathbb{I}(\boldsymbol{\theta} \in T_K))$$

- If we observe multiple samples $\boldsymbol{\theta}_i \sim G$

$$G|\boldsymbol{\theta}_{1:n}, \alpha, H \sim \mathsf{DP}(\alpha + n, \frac{\alpha}{\alpha + n}H + \frac{1}{\alpha + n}\sum_{i=1}^{n} \delta_{\boldsymbol{\theta}_i})$$

Stick-breaking Construction

- Consider a partition $(\boldsymbol{\theta}, \boldsymbol{X}\backslash\boldsymbol{\theta})$ of $\boldsymbol{X}$.
- We know the the posterior process

$$\begin{aligned}(G(\boldsymbol{\theta}), G(\boldsymbol{X}\backslash\boldsymbol{\theta})) &\sim \mathsf{Dir}((\alpha+1)\frac{\alpha H + \delta_{\boldsymbol{\theta}}}{\alpha+1}(\boldsymbol{\theta}), (\alpha+1)\frac{\alpha H + \delta_{\boldsymbol{\theta}}}{\alpha+1}(\boldsymbol{X}\backslash\boldsymbol{\theta})) \\ &= \mathsf{Dir}(1, \alpha)\end{aligned}$$

- G has a point mass located at $\boldsymbol{\theta}$:

$$G = \beta\delta_{\boldsymbol{\theta}} + (1-\beta)G' \quad \text{with} \quad \beta \sim \mathsf{Beta}(1, \alpha)$$

- G' is the (renormalized) probability measure with the point mass removed.

Stick-breaking Construction (cont'd)

- Currently, we have

$$
\begin{aligned}
G &\sim \mathsf{DP}(\alpha+1, \frac{\alpha H + \delta_{\boldsymbol{\theta}}}{\alpha+1}) \\
G &= \beta\delta_{\boldsymbol{\theta}} + (1-\beta)G' \\
\boldsymbol{\theta} &\sim H \\
\beta &\sim \mathsf{Beta}(1,\alpha)
\end{aligned}
$$

- Consider a further partition $(\boldsymbol{\theta}, T_1, \ldots, T_K)$ of $\boldsymbol{X}$

$$
\begin{aligned}
(G(\boldsymbol{\theta}), G(T_1), \ldots, G(T_K)) &= (\beta, (1-\beta)G'(T_1), \ldots, (1-\beta)G'(T_K)) \\
&\sim \mathsf{Dir}(1, \alpha H(T_1), \ldots, \alpha H(T_K))
\end{aligned}
$$

- The agglomerative/decimative property of Dirichlet implies

$$
\begin{aligned}
(G(T_1), \ldots, G(T_k)) &\sim \mathsf{Dir}(\alpha H(T_1), \ldots, \alpha H(T_K)) \\
G' &\sim \mathsf{DP}(\alpha, H)
\end{aligned}
$$

Stick-breaking Construction (cont'd)

- We have

$$
\begin{aligned}
G &\sim \mathsf{DP}(\alpha, H) \\
G &= \beta_1 \delta_{\boldsymbol{\theta}_1} + (1-\beta_1) G_1 \\
G &= \beta_1 \delta_{\boldsymbol{\theta}_1} + (1-\beta_1)(\beta_2 \delta_{\boldsymbol{\theta}_2} + (1-\beta_2) G_2) \\
&\vdots \\
G &= \sum_{k=1}^{\infty} \pi_k \delta_{\boldsymbol{\theta}_k}(\boldsymbol{\theta})
\end{aligned}
$$

  where

$$
\begin{aligned}
\beta_k &\sim \mathsf{Beta}(1, \alpha) \\
\pi_k &= \beta_k \prod_{l=1}^{k=1} = \beta_k (1 - \sum_{l=1}^{k=1} \pi_l)
\end{aligned}
$$

Stick-breaking Construction (cont'd)

- This is often denoted by

$$\pi \sim \text{GEM}(\alpha)$$

  where $\pi \sim \text{GEM}(\alpha)$ and $\boldsymbol{\theta}_k \sim H$.
- Samples from a DP are discrete with probability one.
- In other words, if you keep sampling it, you will get more and more repetitions of previously generated values.

Blackwell-MacQueen Urn Scheme

- Working with infinite dimensional sticks is problematic.
- We can exploit the clustering property to draw samples form a GP.
- **Key:** if $\boldsymbol{\theta}_i \sim G$ are N observations from $G \sim \text{DP}(\alpha, H)$, taking on K distinct values $\boldsymbol{\theta}_k$, then the predictive distribution.

$$p(\bar{\boldsymbol{\theta}}_{N+1} = \boldsymbol{\theta} | \bar{\boldsymbol{\theta}}_{1:N}, \alpha, H) = \frac{1}{\alpha + N}(\alpha H(\boldsymbol{\theta}) + \sum_{k=1}^{K} N_k \delta_{\bar{\boldsymbol{\theta}}_k}(\boldsymbol{\theta}))$$

  where N_k is the number of previous observations equal to $\boldsymbol{\theta}_k$.
- This is the **Polya urn** or **Blackwell-MacQueen** sampling scheme.
- The urn model can be equivalently expressed as

$$\begin{aligned} \boldsymbol{x}_i | \boldsymbol{\theta}_i &\sim F(\boldsymbol{\theta}_i) \\ \boldsymbol{\theta}_i | G &\sim G \\ G &\sim \text{DP}(\alpha, H) \end{aligned}$$

$$\boldsymbol{\theta}_i | \boldsymbol{\theta}_{\mathbf{1:i-1}} \sim \frac{1}{i - 1 + \alpha}(\sum_{j=1}^{i-1} N_j \delta_{\bar{\boldsymbol{\theta}}_j}(\boldsymbol{\theta}) + \alpha H)$$

The Chinese restaurant process (CRP)

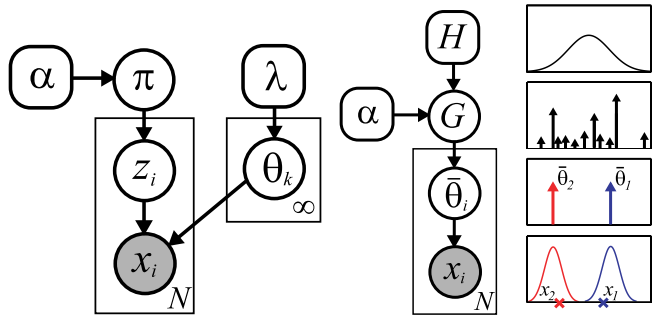
- Let discrete variables z_i specify which value of $\boldsymbol{\theta}_k$ to use.
- That is, we define $\bar{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_{z_i}$.

$$p(z_{N+1} = z|\boldsymbol{z}_{1:N}, \alpha) = \frac{1}{\alpha + N}(\alpha \mathbb{I}(z = k^*) + \sum_{k=1}^{K} N_k \mathbb{I}(z = k))$$

  where k^* represents a new cluster index that has not yet been used.
- This is the **Chinese restaurant process** or **CRP**.
- The tables are like clusters, and the customers are like observations.
  - A person joins an existing table with probability $\frac{N_k}{\alpha+N}$.
  - He may choose to sit at a new table k^* with probability $1/(\alpha + N)$.
- The difference between the CRP and the Polya Urn Model is that the CRP specifies only a distribution over partitions (i.e., table assignments), but doesn't assign parameters to each group, whereas the Polya Urn Model does both.

Outline

Dirichlet Process Mixture



(a) (b)

- The DP is not particularly useful as a model for data directly, since data vectors rarely repeat exactly.
- Useful as a prior for the parameters to generate data.
- Define $G \sim \mathrm{DP}(\alpha, H)$. Equivalently, we can write the model

$$\begin{aligned} \boldsymbol{\pi} &\sim \mathrm{GEM}(\alpha), \quad z_i \sim \boldsymbol{\pi} \\ \boldsymbol{\theta}_k &\sim H(\lambda) \\ \boldsymbol{x}_i &\sim F(\boldsymbol{\theta}_{z_i}) \end{aligned}$$

Gaussian mixture model

$$\boldsymbol{\pi} \sim \mathsf{GEM}(\alpha), \quad z_i \sim \boldsymbol{\pi}$$

$$G = \sum_{k=1}^{K} \pi_k \delta_{\boldsymbol{\theta}_k}(\boldsymbol{\theta}) = \mathsf{DP}(\alpha, N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0))$$

$$\boldsymbol{\mu}_i \sim G$$

$$\boldsymbol{x}_i \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$$

Fitting a DP mixture modeling

- Fit a DPMM by modifying the collapsed Gibbs sampler.

$$p(z_i = k|\boldsymbol{z}_{-i}, \boldsymbol{x}, \alpha, \boldsymbol{\lambda}) \propto p(z_i = k|\boldsymbol{z}_{-i}, \alpha)p(\boldsymbol{x}_i|\boldsymbol{x}_{-i}, z = k, \boldsymbol{z}_{-i}, \boldsymbol{\lambda})$$

- The first term is given by

$$p(\boldsymbol{z}_i|\boldsymbol{z}_{-i}, \boldsymbol{\alpha}) = \frac{\mathbb{I}}{\alpha + N - 1}(\alpha\mathbb{I}(z = k^*) + \sum_{k=1}^{K} N_{k,-i}\mathbb{I}(z = k))$$

- If $z_i = k$, then $\boldsymbol{x}_i$ is conditionally independent of all the data points except those assigned to cluster k. Hence

$$p(\boldsymbol{x}_i|\boldsymbol{x}_{-i}, \boldsymbol{z}_{-i}, z_i = k, \boldsymbol{\lambda}) = p(\boldsymbol{x}_i|\boldsymbol{x}_{-i,k}, \boldsymbol{\lambda}) = \frac{p(\boldsymbol{x}_i, \boldsymbol{x}_{-i,k}|\boldsymbol{\lambda})}{p(\boldsymbol{x}_{-i,k}|\boldsymbol{\lambda})}$$

where

$$p(\boldsymbol{x}_i, \boldsymbol{x}_{-i,k}|\boldsymbol{\lambda}) = \int p(\boldsymbol{x}_i|\boldsymbol{\theta}_k) \left[\prod_{j \neq i, z_j = k} p(\boldsymbol{x}_j|\boldsymbol{\theta}_k) \right] H(\boldsymbol{\theta}_k|\boldsymbol{\lambda}) d\boldsymbol{\theta}_k$$

Fitting a DP mixture modeling(cont'd)

- If $z_i = k^*$, corresponding to a new cluster, we have

$$p(\boldsymbol{x}_i|\boldsymbol{x}_{-i}, \boldsymbol{z}_{-i}, z_i = k^*, \boldsymbol{\lambda}) = p(\boldsymbol{x}_i|\boldsymbol{\lambda}) = \int p(\boldsymbol{x}_i|\boldsymbol{\theta}) H(\boldsymbol{\theta}|\boldsymbol{\lambda}) d\boldsymbol{\theta}$$

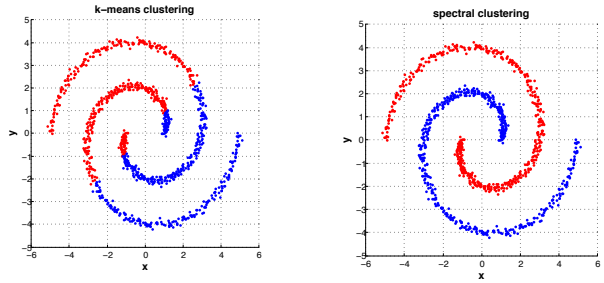
Initialize

for each $i = 1 : N$ in random order **do**
Remove $\boldsymbol{x}_i$'s sufficient statistics from old cluster z_i.
    for each $k = 1 : K$ **do**
    Compute $p_k(\boldsymbol{x}_i) = p(\boldsymbol{x}_i|\boldsymbol{x}_{-i}(k))$.
    Set $N_{k,-i} = \text{dim}(\boldsymbol{x}_{-i}(k))$.
    Compute $p(z_i = k|z_{-i,D}) = \frac{N_{k,-i}}{\alpha+N-1}$.
Compute $p_*(\boldsymbol{x}_i) = p(\boldsymbol{x}_i|\boldsymbol{\lambda})$.
Compute $p(z_i = |\boldsymbol{z}_{-i}, D) = \frac{\alpha}{\alpha+N-1}$.
Normalize $p(z_i|.)$.
Sample $z_i \sim p(z_i|.)$.
Add $\boldsymbol{x}_i$'s sufficient statistics to new cluster z_i.
If any cluster is empty, remove it and decrease K.

Outline

Data Clustering

- Two different criteria
  - Compactness, e.g., k-means, mixture models
  - Connectivity, e.g., spectral clustering



(c) (d)

Figure: Clustering data consisting of 2 spirals. (a) K-means. (b) Spectral clustering. Figure generated by `SpectralClusteringDemo`, written by Wei-Lwun Lu.

Weighted Graph Partitioning

- Some graph terminology
  - Objects (e.g.,pixels,data points) $i \in I =$ vertices of graph G.
  - Edges $(ij) =$ pixel pairs with $W_{ij} > 0$.
  - Similarity matrix $\boldsymbol{W} = [W_{ij}]$.
  - Degree

$$d_i = \sum_{j \in G} W_{ij}$$

$$d_A = \sum_{i \in A} d_i \quad \text{degree of} \quad A \subseteq G$$

  - $\text{Assoc}(A, B) = \sum_{i \in A} \sum_{j \in B} W_{ij}$

Cuts in a Graph

- Edge cut = set of edges whose removal makes a graph disconnected.
- Weight of a cut:

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

- Normalized Cut criteria: minimum $\text{cut}(A, \bar{A})$

$$\text{Ncut}(A, \bar{A}) = \frac{\text{cut}(A, \bar{A})}{d_A} + \frac{\text{cut}(A, \bar{A})}{d_{\bar{A}}}$$

Graph-based Clustering

- Affinity matrix: $\boldsymbol{W} = [W_{ij}]$.
- Degree matrix: $\boldsymbol{D} = \mathsf{diag}(d_i)$.
- Laplacian matrix: $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$.
- (Bipartite) partition vector:

$$\boldsymbol{x} = [x_1, \ldots, x_n] = [1, 1, \ldots, -1, \ldots, -1]$$

Clustering via Optimizing Normalized Cut

- The normalized cut:

$$\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{d_A} + \frac{\text{cut}(A, B)}{d_B}$$

- Transform Ncut equation to a matrix form (Shi & Malik 2000):

$$\min_x \text{Ncut}(x) = \min_y \frac{y^T(\boldsymbol{D} - \boldsymbol{W})y}{y^T \boldsymbol{D} y}$$

$$\text{subject to: } y \in \{1, -b\}^n$$
$$y^T \boldsymbol{D}\mathbf{1} = 0$$

- Relax the continuous domain by solving generalized eigenvalue system:

$$\min_y y^T(\boldsymbol{D}-\boldsymbol{W})y, \quad \text{s.t.} \quad y^T\boldsymbol{D}y = \mathbf{1}$$

  - Which gives:

$$(\boldsymbol{D}-\boldsymbol{W})y = \lambda \boldsymbol{D}y$$

  - Note that $(\boldsymbol{D}-\boldsymbol{W})\mathbf{1} = 0$ so, the first eigenvector is $y_0 = 1$ with eigenvalue 0.
  - The second smallest eigenvector is the real valued solution to this problem.

Algorithm

- Define a similarity function between 2 nodes. i.e.

$$W_{i,j} = \exp(-\frac{\|X_{(i)} - X_{(j)}\|_2^2}{\sigma_X^2})$$

- Compute affinity matrix ($\boldsymbol{W}$) and degree matrix ($\boldsymbol{D}$).
- Solve

$$(\boldsymbol{D} - \boldsymbol{W})y = \lambda \boldsymbol{D} y$$

  - Do singular value decomposition (SVD) of the graph Laplacian $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$.

$$\boldsymbol{L} = \boldsymbol{V}^T \boldsymbol{\Lambda} \boldsymbol{V} \Rightarrow y^*$$

- Use the eigenvector with the second smallest eigenvalue, bipartition the graph
  - For each threshold k,

$$A_k = \{i | y_i \text{ among } k \text{ largest element of } y^*\}$$
$$B_k = \{i | y_i \text{ among } n-k \text{ smallest element of } y^*\}$$

  - Compute Ncut(A_k, B_k)
  - Output: $k^* = \text{argmax}\,\text{Ncut}(A_k, B_k)$, A_{k^*}, B_{k^*}