Nanjing University of Aeronautics and Astronautics
College of Computer Science & Technology

Adaptive Weighted Spectral Clustering

Authors: Muhammad Yousefnezhad, Daoqiang Zhang

Presented by: Muhammad Yousefnezhad

Outline

1 Cluster Ensemble Selection

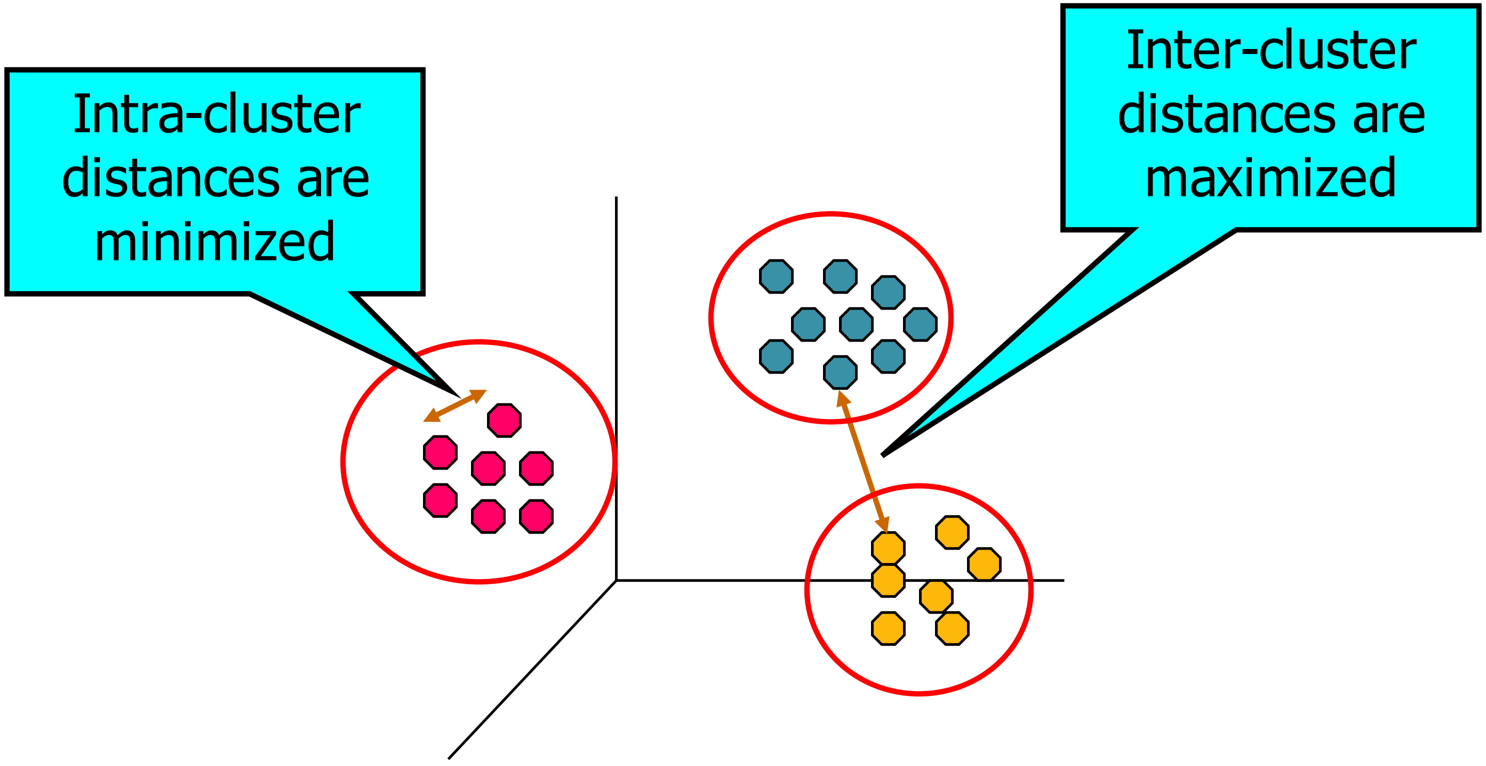
2 The proposed method

3 Experimental Results

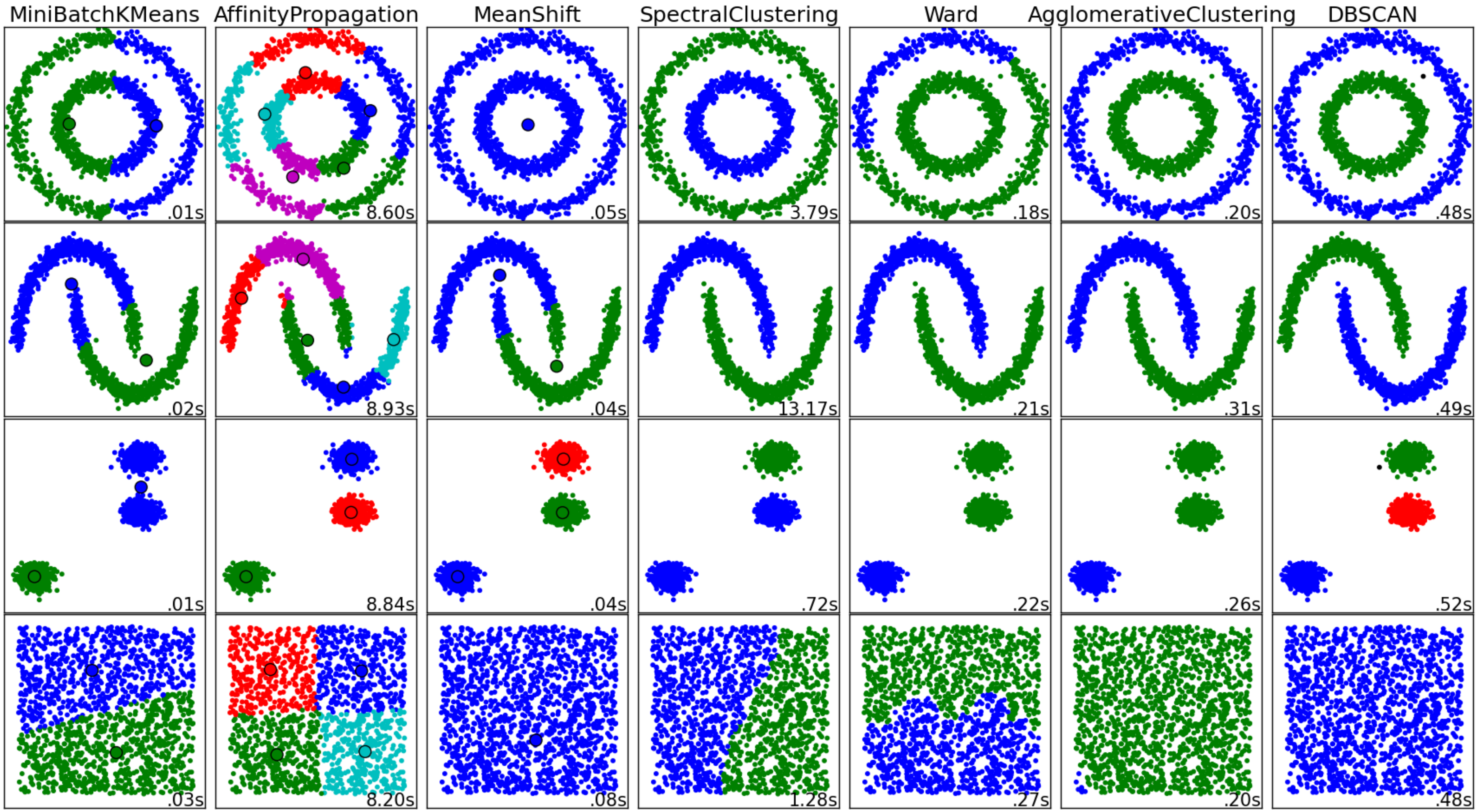
4 Summary

Clustering

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.

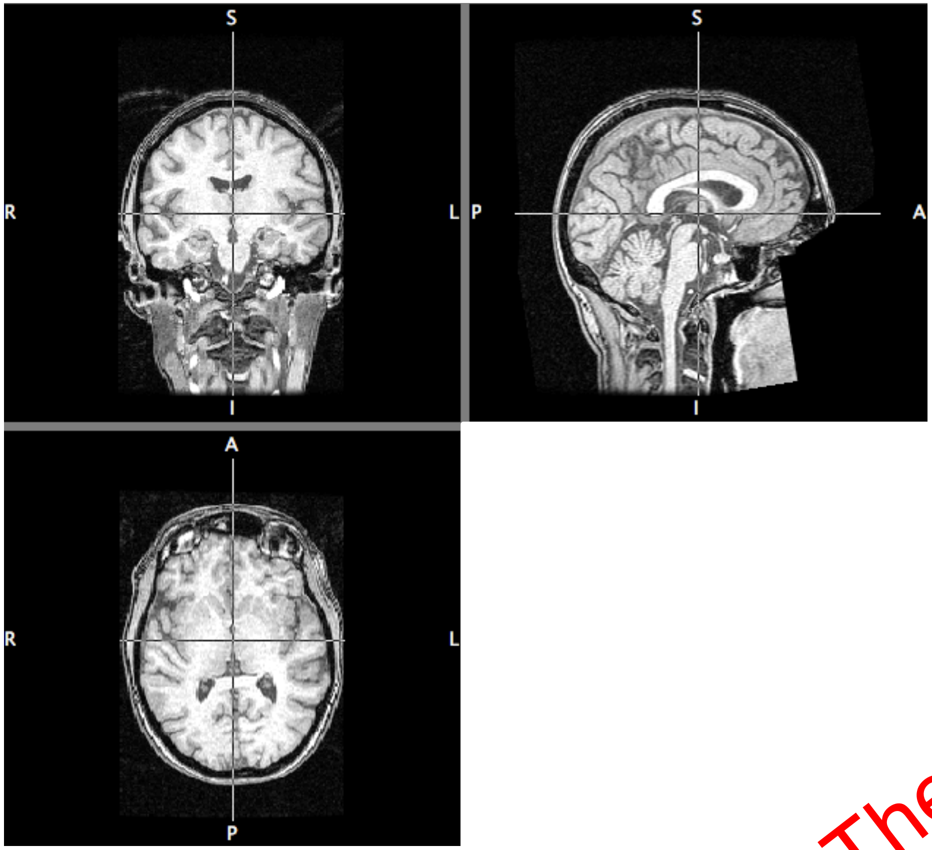


Challenge



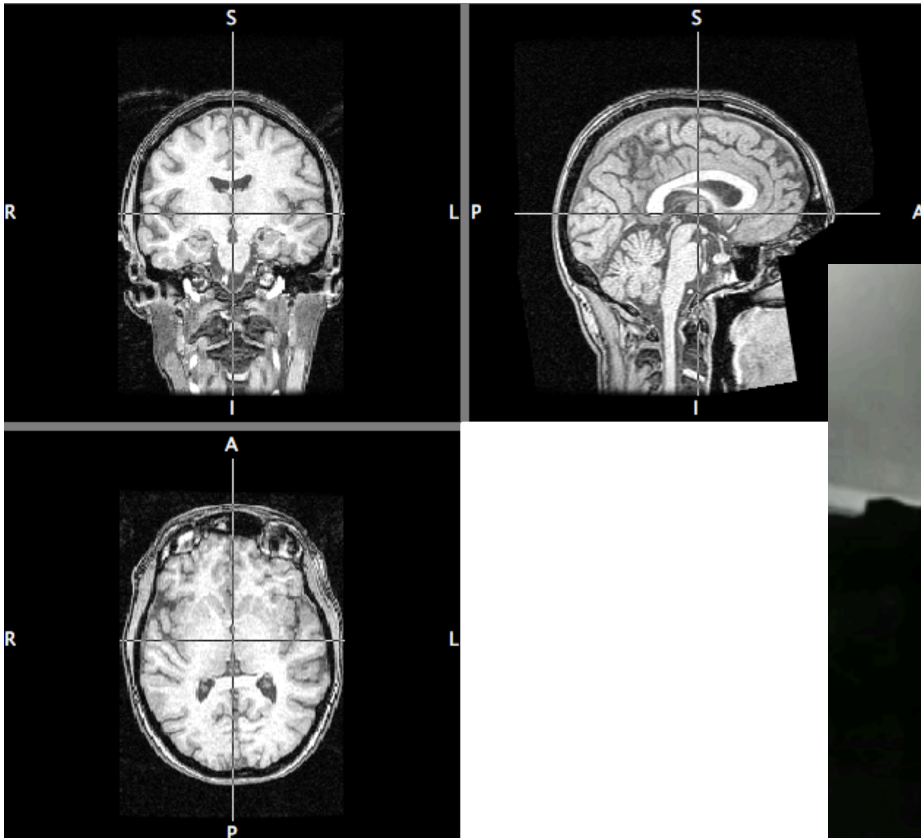
MiniBatchKMeans
AffinityPropagation
MeanShift
SpectralClustering
Ward
AgglomerativeClustering
DBSCAN
.01s
8.60s
.05s
3.79s
.18s
.20s
.48s
.02s
8.93s
.04s
13.17s
.21s
.31s
.49s
.01s
8.84s
.04s
.72s
.22s
.26s
.52s
.03s
8.20s
.08s
1.28s
.27s
.20s
.48s

Brain Extraction Problem

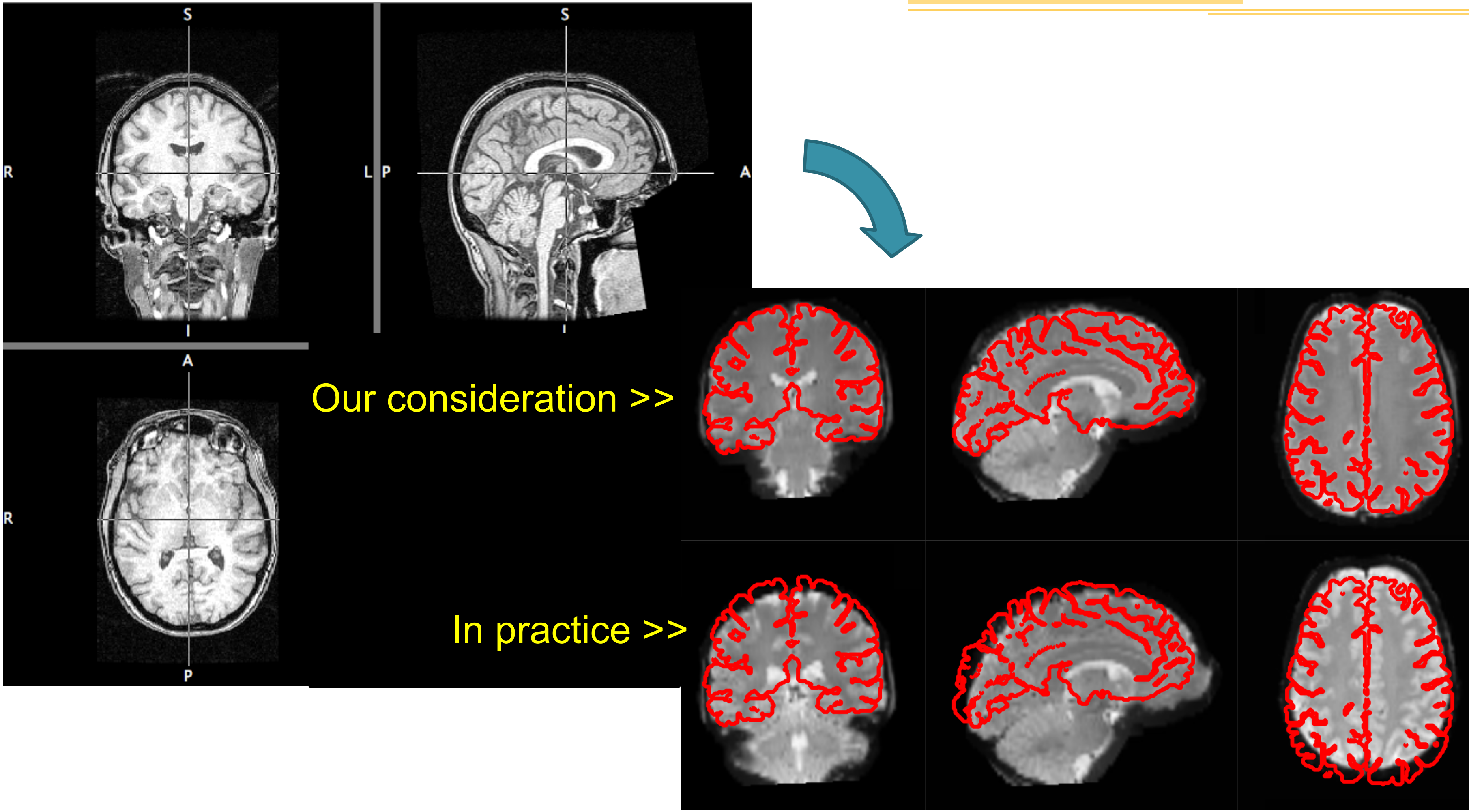


There are two approaches

Brain Extraction Problem

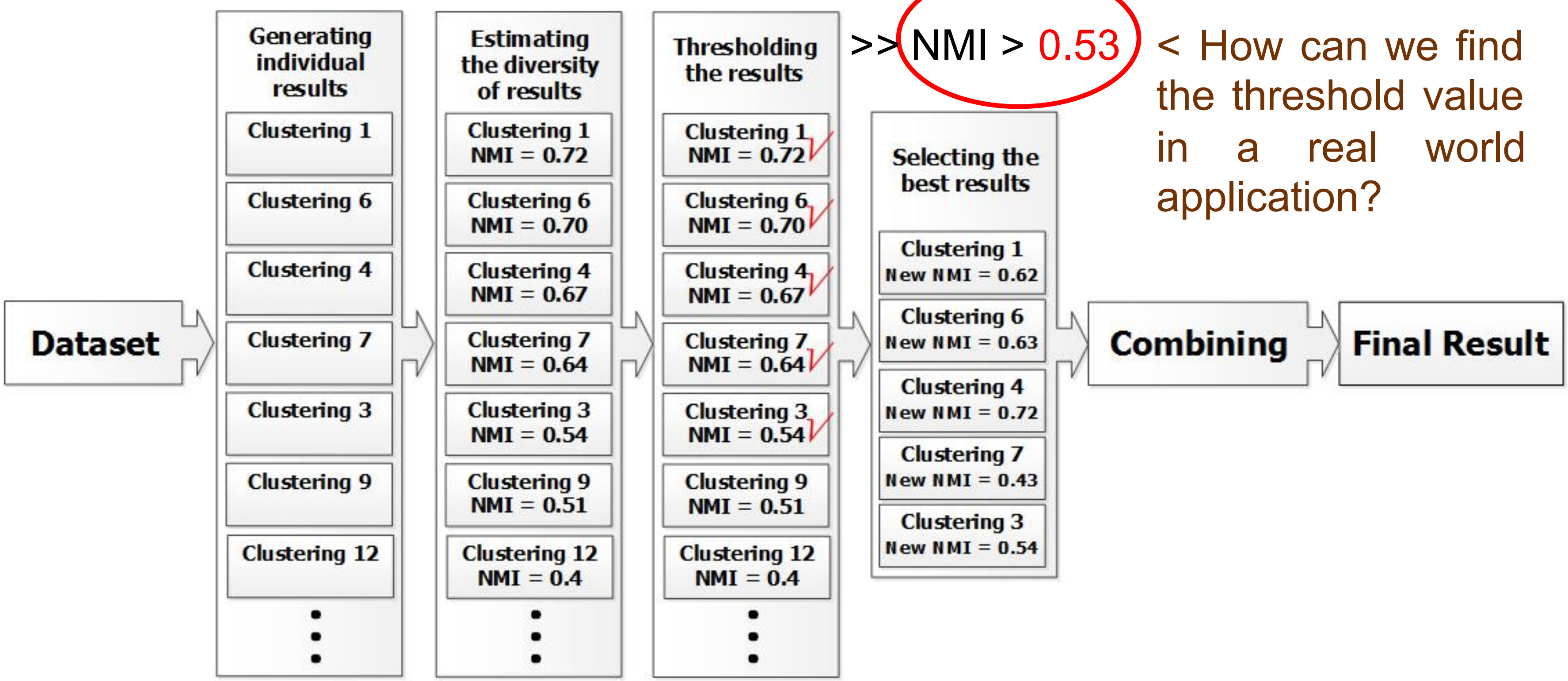


Brain Extraction Problem



Cluster Ensemble Selection Approach

- We need a robust diversity metirc
- The performance of CES is significantly sensitive to the threshold value.

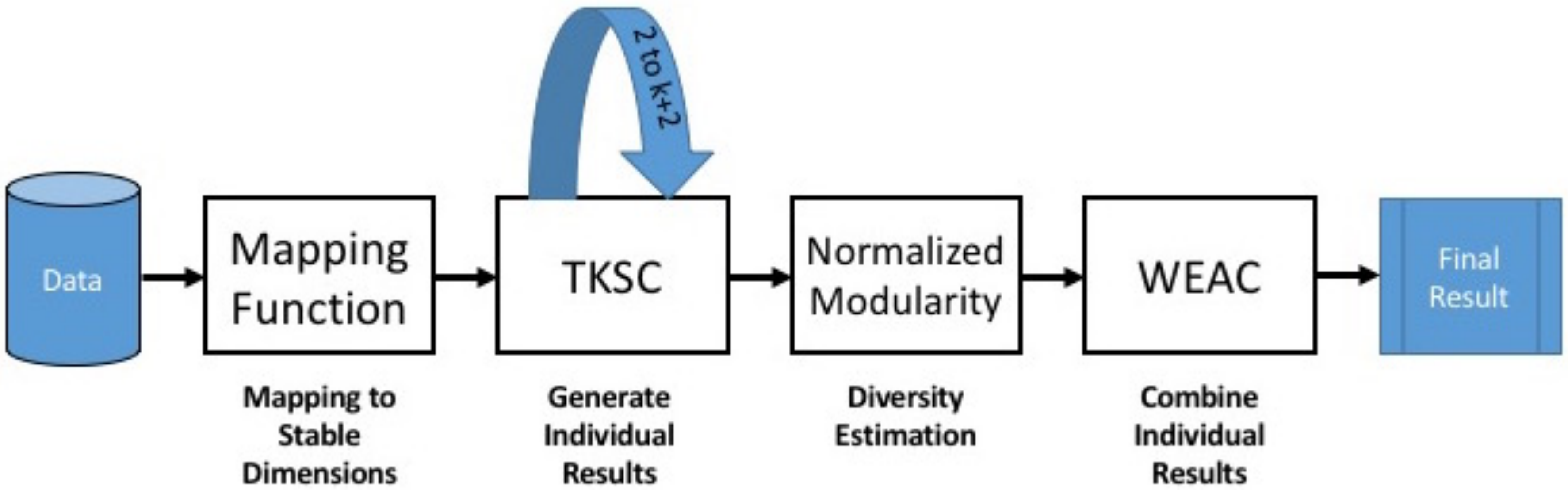


< How can we find the threshold value in a real world application?

Outline

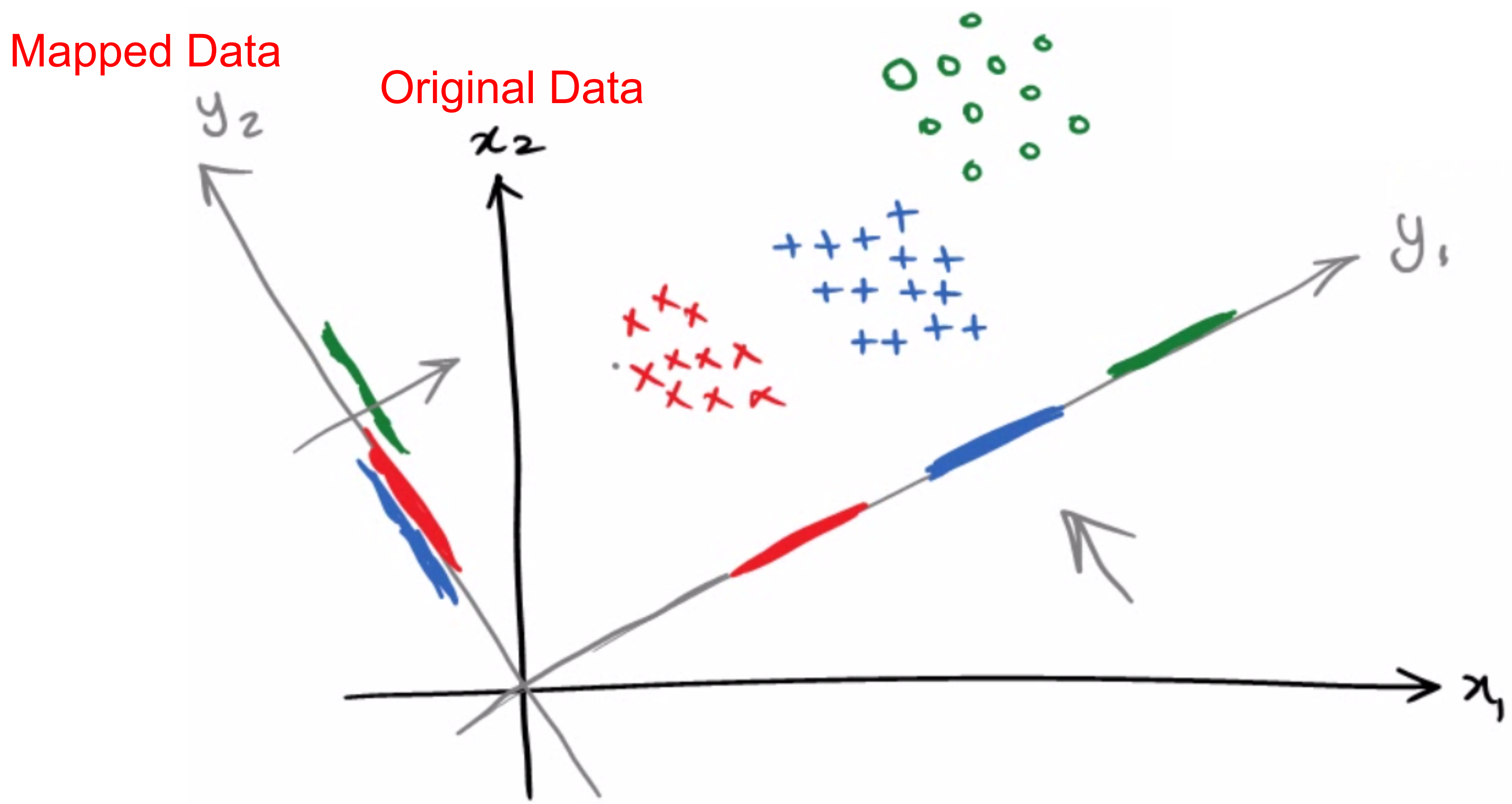
1. Cluster Ensemble Selection
2. The proposed method
3. Experimental Results
4. Summary

The proposed framework



Step 1: The Mapping Function

- Main Idea of mapping function is transforming data to stable dimensions.



Step 1: The Mapping Function

Algorithm 1 The Mapping Function

Input: Data set $\hat{X} \in \mathbb{R}^{m \times n} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}$,
d as number of features:
$d = 0$ *is considered for deactivating the feature selection*

Output: Mapped data set Y

Method:

1. Calculating simple average $\bar{X}$ by using (1).
2. Calculating X by using (2).
3. Calculating $R = \mathbb{E}\{XX^T\} = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$.
4. Calculating Λ and Q as eigenvalues/vectors of R.
5. Sorting Q based on descending values of λ.
6. **if** d is not zero ($d \neq 0$)
   then selecting $[1, d]$ features of Q, and sorting as Q_d,
   else $Q_d = Q$, $d = m$.
   end if
7. Return $Y = Q_d^T X$.

<< calculating the zero-mean of data

Step 1: The Mapping Function

Algorithm 1 The Mapping Function

Input: Data set $\hat{X} \in \mathbb{R}^{m \times n} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$,
d as number of features:
$d = 0$ *is considered for deactivating the feature selection*

Output: Mapped data set Y

Method:

1. Calculating simple average $\bar{X}$ by using (1).
2. Calculating X by using (2).
3. Calculating $R = \mathbb{E}\{XX^T\} = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$. << constructing R
4. Calculating Λ and Q as eigenvalues/vectors of R.
5. Sorting Q based on descending values of λ.
6. **if** d is not zero ($d \neq 0$)
   then selecting $[1, d]$ features of Q, and sorting as Q_d,
   else $Q_d = Q$, $d = m$.
   end if
7. Return $Y = Q_d^T X$.

Step 1: The Mapping Function

Algorithm 1 The Mapping Function

Input: Data set $\hat{X} \in \mathbb{R}^{m \times n} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$,
d as number of features:
$d = 0$ *is considered for deactivating the feature selection*

Output: Mapped data set Y

Method:

1. Calculating simple average $\bar{X}$ by using (1).
2. Calculating X by using (2).
3. Calculating $R = \mathbb{E}\{XX^T\} = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$.
4. Calculating Λ and Q as eigenvalues/vectors of R.
5. Sorting Q based on descending values of λ.
6. **if** d is not zero ($d \neq 0$)
   then selecting $[1, d]$ features of Q, and sorting as Q_d,
   else $Q_d = Q$, $d = m$.
   end if
7. Return $Y = Q_d^T X$.

<< calculating the eigenvalues and eigenvectors and sort them

Step 1: The Mapping Function

Algorithm 1 The Mapping Function

Input: Data set $\hat{X} \in \mathbb{R}^{m \times n} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$,
d as number of features:
$d = 0$ *is considered for deactivating the feature selection*

Output: Mapped data set Y

Method:

1. Calculating simple average $\bar{X}$ by using (1).
2. Calculating X by using (2).
3. Calculating $R = \mathbb{E}\{XX^T\} = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$.
4. Calculating Λ and Q as eigenvalues/vectors of R.
5. Sorting Q based on descending values of λ.
6. **if** d is not zero ($d \neq 0$)
   then selecting $[1, d]$ features of Q, and sorting as Q_d,
   else $Q_d = Q$, $d = m$. << the optional feature selection
   end if
7. Return $Y = Q_d^T X$.

Step 1: The Mapping Function

Algorithm 1 The Mapping Function

Input: Data set $\hat{X} \in \mathbb{R}^{m \times n} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$,
d as number of features:
$d = 0$ is considered for deactivating the feature selection

Output: Mapped data set Y

Method:

1. Calculating simple average $\bar{X}$ by using (1).
2. Calculating X by using (2).
3. Calculating $R = \mathbb{E}\{XX^T\} = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$.
4. Calculating Λ and Q as eigenvalues/vectors of R.
5. Sorting Q based on descending values of λ.
6. **if** d is not zero ($d \neq 0$)
   then selecting $[1, d]$ features of Q, and sorting as Q_d,
   else $Q_d = Q$, $d = m$.
   end if
7. Return $Y = Q_d^T X$. << apply mapping function on data points

Step 2: Generating individual results

- Transforming data point to similarity matrix S

$$S_{i,j} = \begin{cases} exp\left(\frac{-\|y_i - y_j\|_2}{\phi^2}\right) & if\ i \neq j \\ 0 & if\ i = j \end{cases}$$

$$\begin{array}{c c} & \begin{array}{cccc} X1 & X2 & \cdots & Xn \end{array} \\ \begin{array}{c} X1 \\ X2 \\ \vdots \\ Xn \end{array} & \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix} \end{array}$$

Ø is the scaling parameter for controlling how rapidly affinity $S_{i,j}$

Ø can be calculated automatically by Ng et al., 2001.

Step 2: Generating individual results

Algorithm Two Kernels Spectral Clustering (TKSC)

Input: Distance matrix A, Number of clusters K

Output: Partitional result P, Modular result M

Method:

1. Generate similarity matrix S by using A
2. Generate diagonal matrix D by using S.

<< calculating the similarity and its diagonal matrix

3. Generate L_P by applying S and D on $L_P = I - D^{1/2}SD^{1/2}$
4. Generate L_M by using S and D on $L_M = D - S$
5. Generate the matrix V as eigenvectors of L_p.
6. Generate U as normalized V by using $SQ_i = \left(\sum_{i=1}^{M} V_{i1} \times V_{i2}\right)^{\frac{1}{2}} + \epsilon$ and $U_{ij} = V_{ij} \times SQ_i$
7. Generate M by applying L_M on $M = \frac{1}{max(L_M)} L_M$
8. $P = kmeans(U, K)$

Step 2: Generating individual results

Algorithm Two Kernels Spectral Clustering (TKSC)

Input: Distance matrix A, Number of clusters K

Output: Partitional result P, Modular result M

Method:

1. Generate similarity matrix S by using A
2. Generate diagonal matrix D by using S.
3. Generate L_P by applying S and D on $L_P = I - D^{1/2}SD^{1/2}$
4. Generate L_M by using S and D on $L_M = D - S$
5. Generate the matrix V as eigenvectors of L_p.
6. Generate U as normalized V by using $SQ_i = \left(\sum_{i=1}^{M} V_{i1} \times V_{i2}\right)^{\frac{1}{2}} + \epsilon$ and $U_{ij} = V_{ij} \times SQ_i$
7. Generate M by applying L_M on $M = \frac{1}{max(L_M)} L_M$
8. $P = kmeans(U, K)$

<< calculating the first Laplacian matrix as the paritional kernel

Step 2: Generating individual results

Algorithm Two Kernels Spectral Clustering (TKSC)

Input: Distance matrix A, Number of clusters K

Output: Partitional result P, Modular result M

Method:

1. Generate similarity matrix S by using A
2. Generate diagonal matrix D by using S.
3. Generate L_P by applying S and D on $L_P = I - D^{1/2}SD^{1/2}$
4. $\boxed{\text{Generate } L_M \text{ by using } S \text{ and } D \text{ on } L_M = D - S}$
5. Generate the matrix V as eigenvectors of L_p.
6. Generate U as normalized V by using $SQ_i = \left(\sum_{i=1}^{M} V_{i1} \times V_{i2}\right) + \epsilon$ and $U_{ij} = V_{ij} \times SQ_i$
7. Generate M by applying L_M on $M = \frac{1}{max(L_M)} L_M$
8. $P = kmeans(U, K)$

<< calculating the second Laplacian matrix as the modular kernel

Step 2: Generating individual results

Algorithm Two Kernels Spectral Clustering (TKSC)

Input: Distance matrix A, Number of clusters K

Output: Partitional result P, Modular result M

Method:

1. Generate similarity matrix S by using A
2. Generate diagonal matrix D by using S.
3. Generate L_P by applying S and D on $L_P = I - D^{1/2}SD^{1/2}$
4. Generate L_M by using S and D on $L_M = D - S$
5. Generate the matrix V as eigenvectors of L_p. << calculating the eigenvectors of Lp
6. Generate U as normalized V by using $SQ_i = \left(\sum_{i=1}^{M} V_{i1} \times V_{i2}\right) + \epsilon$ and $U_{ij} = V_{ij} \times SQ_i$
7. Generate M by applying L_M on $M = \frac{1}{max(L_M)} L_M$
8. $P = kmeans(U, K)$

Step 2: Generating individual results

Algorithm Two Kernels Spectral Clustering (TKSC)

Input: Distance matrix A, Number of clusters K
Output: Partitional result P, Modular result M
Method:

1. Generate similarity matrix S by using A
2. Generate diagonal matrix D by using S.
3. Generate L_P by applying S and D on $L_P = I - D^{1/2}SD^{1/2}$
4. Generate L_M by using S and D on $L_M = D - S$
5. Generate the matrix V as eigenvectors of L_p.
6. Generate U as normalized V by using $SQ_i = \left(\sum_{i=1}^{M} V_{i1} \times V_{i2}\right)^{\frac{1}{2}} + \epsilon$ and $U_{ij} = V_{ij} \times SQ_i$

Normalizing the eigenvectors >>

7. Generate M by applying L_M on $M = \frac{1}{max(L_M)} L_M$
8. $P = kmeans(U, K)$

Step 2: Generating individual results

Algorithm Two Kernels Spectral Clustering (TKSC)

Input: Distance matrix A, Number of clusters K
Output: Partitional result P, Modular result M
Method:

1. Generate similarity matrix S by using A
2. Generate diagonal matrix D by using S.
3. Generate L_P by applying S and D on $L_P = I - D^{1/2}SD^{1/2}$
4. Generate L_M by using S and D on $L_M = D - S$
5. Generate the matrix V as eigenvectors of L_p.
6. Generate U as normalized V by using $SQ_i = \left(\sum_{i=1}^{M} V_{i1} \times V_{i2}\right)^{\frac{1}{2}} + \epsilon$ and $U_{ij} = V_{ij} \times SQ_i$
7. Generate M by applying L_M on $\boxed{M = \frac{1}{max(L_M)} L_M}$
8. $P = kmeans(U, K)$

<< Normalizing the modular result

Step 2: Generating individual results

Algorithm Two Kernels Spectral Clustering (TKSC)

Input: Distance matrix A, Number of clusters K

Output: Partitional result P, Modular result M

Method:

1. Generate similarity matrix S by using A
2. Generate diagonal matrix D by using S.
3. Generate L_P by applying S and D on $L_P = I - D^{1/2}SD^{1/2}$
4. Generate L_M by using S and D on $L_M = D - S$
5. Generate the matrix V as eigenvectors of L_p.
6. Generate U as normalized V by using $SQ_i = \left(\sum_{i=1}^{M} V_{i1} \times V_{i2}\right)^{\frac{1}{2}} + \epsilon$ and $U_{ij} = V_{ij} \times SQ_i$
7. Generate M by applying L_M on $M = \frac{1}{max(L_M)} L_M$
8. $\boxed{P = kmeans(U, K)}$ << calculating the partitional results

Step 3: Diversity evaluation

- This paper proposes Normalized Modularity for calculating the diversity by exploiting the partitional and modular results.

- This metric employs the concept of Expected Value for calculating the diversity.

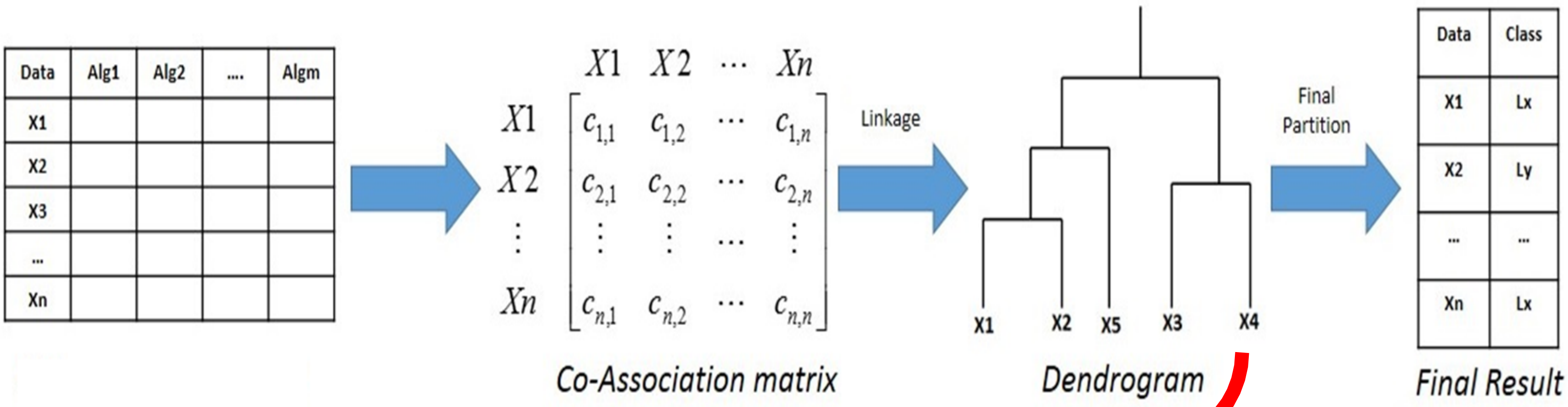
- This metric is a new branch of famous Modularity, which is an effective metric in the field of community detection, for general clustering problem.

$$NM(P^l, M) = \frac{1}{2} + \frac{1}{4z}\sum_{ij}\left[\Gamma_{ij} - \frac{\sigma_i \sigma_j}{2z}\right]\Theta\left(c_i, c_j\right)$$

$$\Gamma_{ij} = \begin{cases} 0 & \textit{if } M_{ij} = 0 \\ 1 & \textit{Otherwise} \end{cases} \qquad \Theta\left(c_i, c_j\right) = \begin{cases} 1 & \textit{if } c_i = c_j \\ 0 & \textit{Otherwise} \end{cases}$$

P is Partitional result. M is Modular result.
z is sum of all cells in the matrix M ($m = \sum_M M_{ij}$).
c_i and c_j are the number of classes for the i-th and j-th instances in the P.
σ_i, σ_j show the degree of i-th and j-th nodes in the graph of the M.
This diversity evaluation is $0 \leq NM \leq 1$.

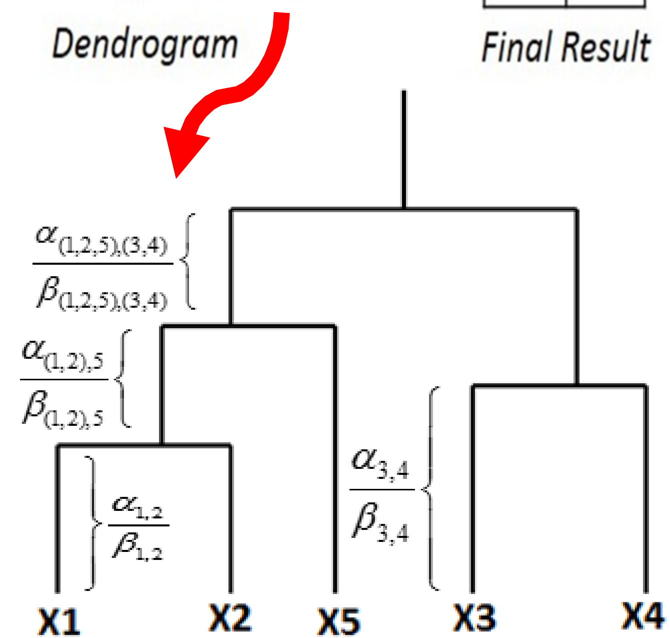
Step 4: Evidence Accumulation Clustering



- α represents the number of clusters shared by objects with indices i and j.

- β is the number of partitions in which this pair of instances (i and j) is simultaneously presented.

- In fact, EAC considers that the weights of all algorithms results are the same.



Step 4: Weighted EAC

- WEAC: $$c\left(i,j\right)=\frac{\sum_{\alpha(i,j)}\rho_{i,j}}{\beta\left(i,j\right)}$$
- Although the weight can have different definitions in the other applications, this paper uses average of Normalized Modularity of two algorithms as follows for combining individual results:
$$\rho_{ij}=\frac{1}{2}(NM_i+NM_j)$$
- Final co-association matrix:
$$\xi = WEAC(\zeta) = \begin{pmatrix} c(1,1) & c(1,2) & \dots & c(1,n) \\ c(2,1) & c(2,2) & \dots & c(2,n) \\ \vdots & \vdots & \vdots & \vdots \\ c(i,1) & c(i,2) & c(i,j) & c(i,n) \\ \vdots & \vdots & \vdots & \vdots \\ c(n,1) & c(n,2) & \dots & c(n,n) \end{pmatrix}$$

Outline

Experiment Setup

- Data Set: we employ 26 standard data
  - Image based data set
    - ✓ Alzheimer's Disease data set (MRI and PET images from human brain)
    - USPS: a handwriting data set
  - Document based data set
    - ✓ 20 Newsgroups, Reuters-21578
  - More than 20 data set mostly from UCI data repository

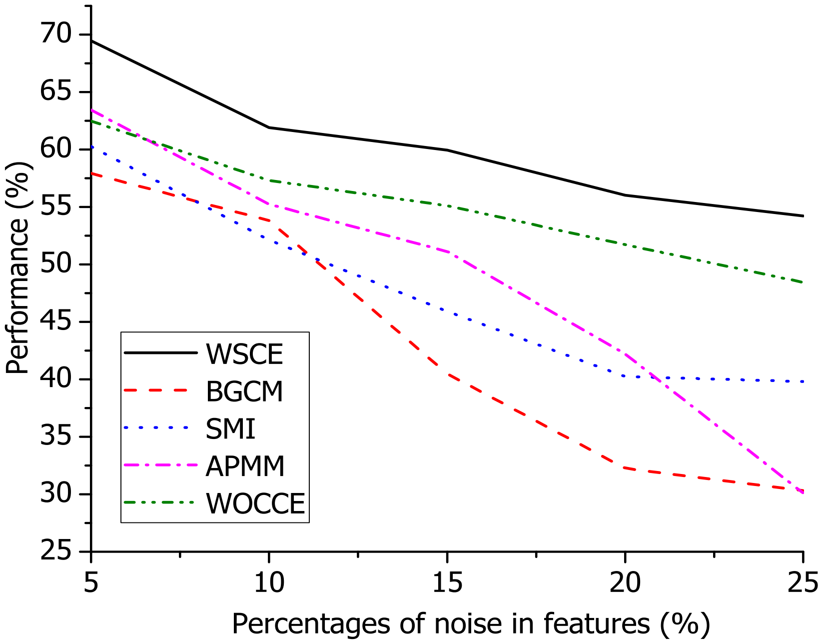
- Algorithms:
  - Individual Clustering methods:
    - Spectral clustering (Ng et al., 2001), MLE (Chen el al., 2014)
  - Cluster Ensemble (Selection) methods:
    - APMM (Alizadeh et al., 2014), WOCCE (Alizadeh et al., 2015), SMI (Romano et al., 2014), BGCM (Gao et al., 2013)

Performance Analysis

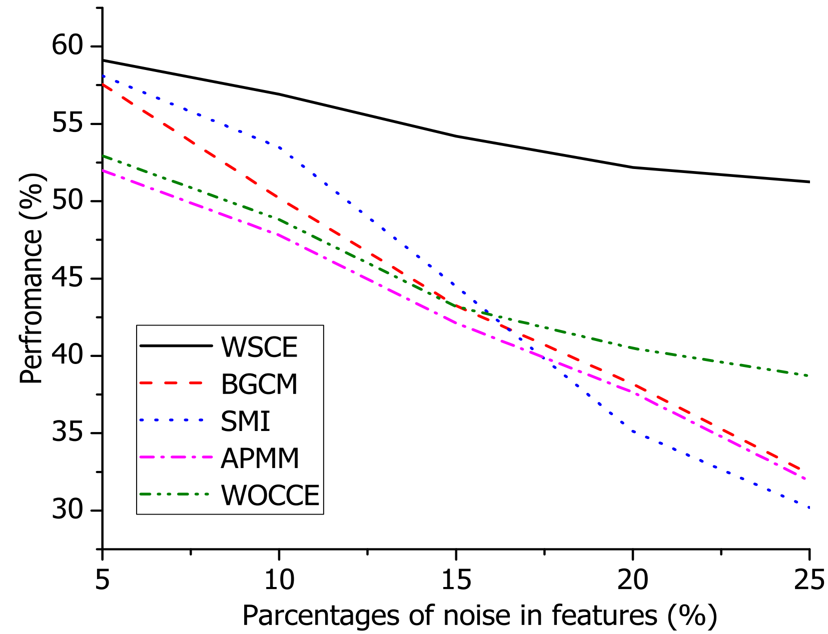
Data Sets	Spectral	MLE	APMM	WOCCE	SMI	BGCM	WSCE
20 Newsgroups	14.31±2.14	21.89±1.02	28.03±0.87	32.62±0.52	29.14±0.91	40.61±0.83	**52.06±0.17**
ADNI-MRI-C1	39.24±0.21	39.84±0.42	48.01±0.56	48.82±0.37	**50.69±0.69**	45.54±0.99	49.53±0.19
ADNI-MRI-C2	32.72±0.98	26.32±0.67	39.93±0.29	40.22±0.44	38.32±0.41	**42.62±1.04**	41.14±0.71
ADNI-PET-C1	43.71±0.52	37.96±0.87	48.37±0.82	49.19±0.26	49.45±0.62	42.1±0.78	**52.05±0.37**
ADNI-PET-C2	37.27±0.23	37.91±0.83	38.53±0.17	39.43±0.79	41.76±0.47	39.1±1.2	**43.11±0.42**
ADNI-FUL-C1	42.63±0.63	42.62±0.58	47.22±0.93	48.82±0.41	47.93±0.83	48.56±1.26	**49.06±0.36**
ADNI-FUL-C2	39.51±1.19	41.06±0.17	50.09±0.35	49.39±0.63	49.16±0.26	46.91±0.42	**50.11±0.09**
Arcene	58.31±1.22	64.19±0.498	66.28±0.216	65.16±0.32	67.14±0.93	64.23±0.28	**73.34±0.92**
Bala. Scale	49.21±0.87	52.76±0.12	52.65±0.63	54.88±0.61	59.98±0.812	59.62±0.32	**61.64±0.12**
Breast Can.	94.88±1.14	82.65±0.342	96.04±0.88	96.92±0.77	80.87±0.652	99.12±0.62	**99.21±0.43**
Bupa	56.72±1.18	53.98±0.274	55.07±0.28	57.02±0.46	58.49±0.21	53.17±0.21	**60.93±0.09**
CNAE-9	65.32±0.43	77.72±0.591	77.42±0.792	79.2±0.579	74.25±0.614	80.12±0.459	**88.42±0.02**
Galaxy	31.24±0.67	34.25±0.872	33.72±0.36	35.88±0.81	35.21±0.413	36.91±0.17	**39.89±0.82**
Glass	45.78±0.87	50.32±0.42	47.19±0.21	51.82±0.92	54.19±0.144	53.66±0.98	**55.19±0.51**
Half Ring	80.61±1.15	73.91±0.762	80±0.42	87.2±0.14	71.19±0.621	98.37±0.59	**99.92±0.08**
Ionosphere	69.71±0.67	25.67±0.53	70.94±0.13	70.52±0.132	70.87±0.226	73.67±0.341	**76.25±0.28**
Iris	83.45±0.82	89.02±0.61	74.11±0.25	92±0.59	93.79±0.21	**97.29±0.09**	96.53±0.32
Optdigit	54.19±0.45	73.81±0.69	77.1±0.841	77.16±0.21	80.21±0.79	71.56±0.692	**82.82±0.33**
Pendigits	53.94±0.25	59.36±0.31	47.4±0.699	58.68±0.18	63.74±0.37	63.13±0.42	**65.02±0.91**
Reuters-21578	48.78±3.19	52.58±1.92	65.23±0.62	68.85±0.32	62.92±1.02	71.69±0.51	**78.34±0.15**
SA Hart	69.59±0.08	61.69±0.44	70.91±0.42	68.7±0.46	70.05±0.51	**73.92±0.72**	72.8±0.82
Sonar	53.24±0.62	54.93±0.26	54.1±0.91	54.39±0.25	57.64±0.47	52.06±0.873	**61.29±0.11**
Statlog	42.87±0.62	52.35±0.79	54.88±0.528	55.77±0.719	53.73±0.52	55.76±0.591	**57.92±0.26**
USPS	62.67±0.13	59.72±0.62	63.91±0.94	65.21±0.69	68.73±0.66	65.38±1.02	**70.37±0.01**
Wine	73.09±1.38	83.81±0.41	64.6±0.231	71.34±0.542	88.46±0.71	87.34±0.24	**90.44±0.02**
Yeast	32.96±0.71	30.49±0.63	31.06±0.245	32.76±0.268	35.19±0.57	28.12±0.462	**36.92±0.81**

Noise Analysis

- The effect noisy data on the performance of the proposed method



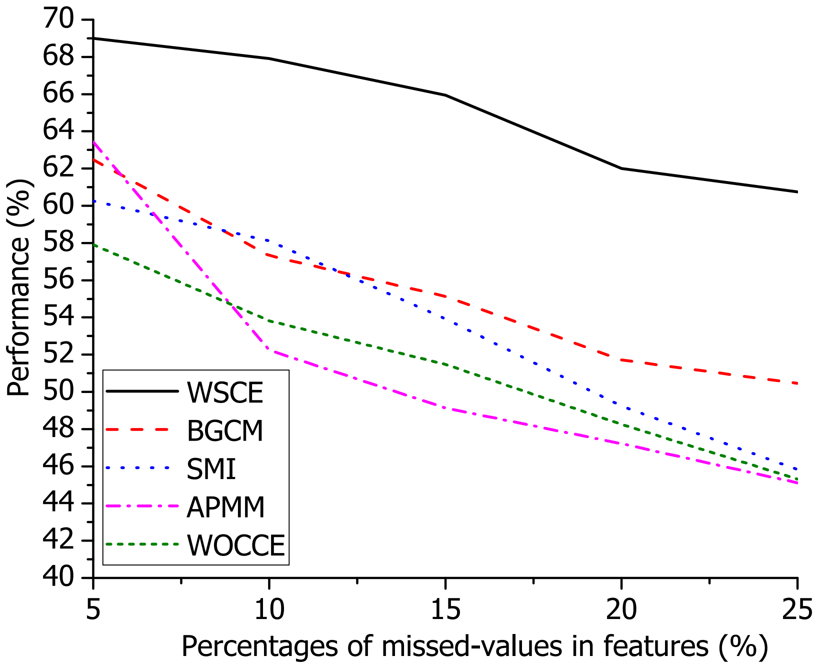
(a) Arcene



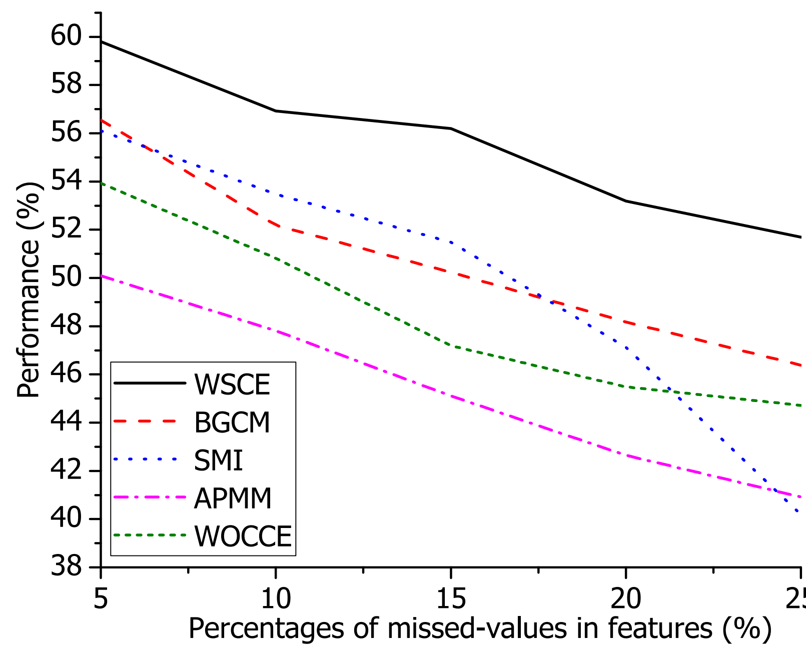
(b) CNAE-9

Missed-values Analysis

- The effect missed-values on the performance of the proposed method



(a) Arcene



(b) CNAE-9

Outline

1. Cluster Ensemble Selection
2. The proposed method
3. Experimental Results
4. Summary

Summary

- There are two challenges in Cluster Ensemble Selection:
  - Proposing a robust consensus metric(s) for diversity evaluation.
  - Estimating optimum parameters in the thresholding procedure for selecting the evaluated results.

- This paper introduces a novel solution for solving mentioned challenges:
  - Mapping function and Optional feature selection (preparing raw data)
  - Two Kernel Spectral Clustering (TKSC) algorithm (generating individual results)
  - Normalized Modularity (estimating diversity)
  - Weighted Evidence Accumulation Clustering (generating final result)

- An extensive experimental study is performed by comparing with individual clustering methods as well as cluster ensemble (selection) methods on a large number of data sets.

- Results clearly show the superiority of our approach on both normal data sets and those with noise or missing values.

- In the future, we will develop a new version of Normalized Modularity for estimating the diversity of Partitional results, directly.

Acknowledgement

- Dr. Sheng-Jun Huang (NUAA)
- National Natural Science Foundation of China
- Jiangsu Natural Science Foundation
- NUAA Fundamental Research Funds

Thanks for your attention!

For more details, contact:

myousefnezhad@nuaa.edu.cn

myousefnezhad@outlook.com

http://ibrain.nuaa.edu.cn/myousefnezhad