# Ensemble Learning for Spectral Clustering

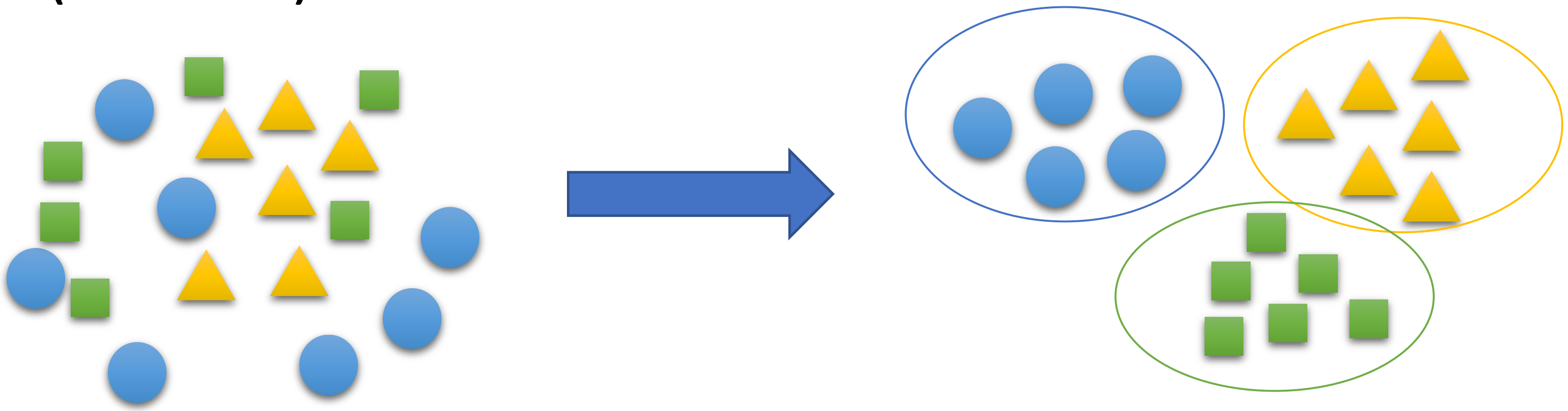
Hongmin Li, Xiucai Ye, Akira Imakura and Tetsuya Sakurai
Department of Computer Science
University of Tsukuba
Tsukuba, Japan

# Outlines

- **Introduction**
- Proposed method
- Experimental results
- Conclusion

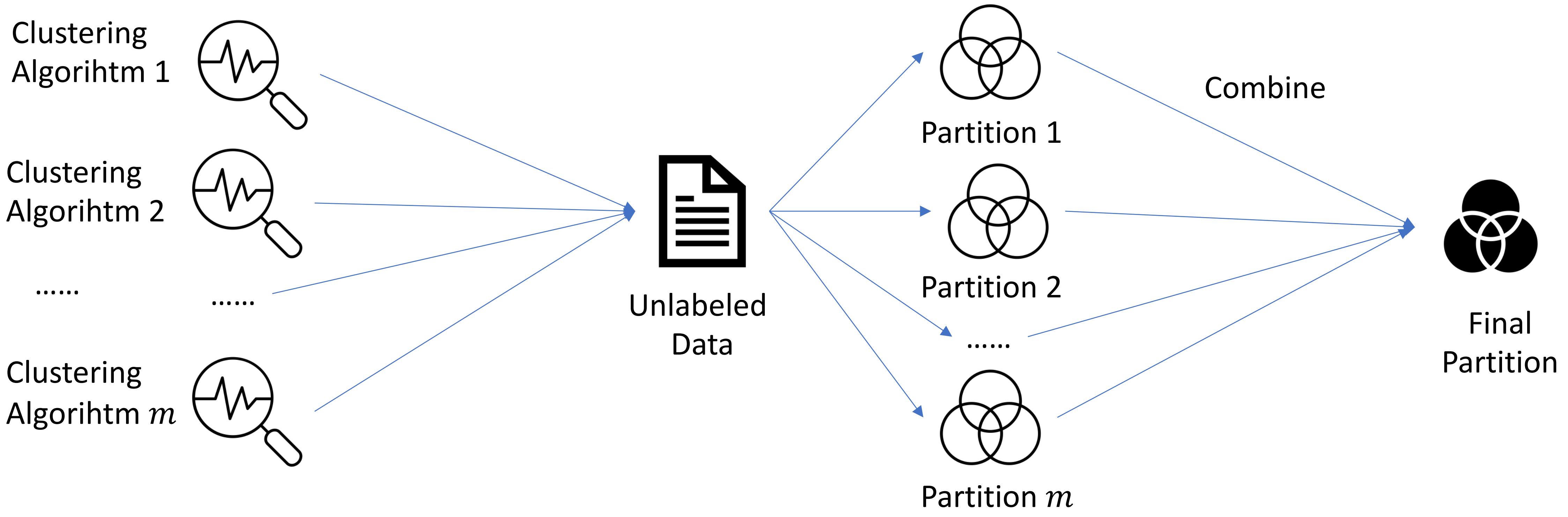
# Clustering

- Unsupervised learning technique to find hidden structure in unlabeled data
- Grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar each other than to those in other groups (clusters).



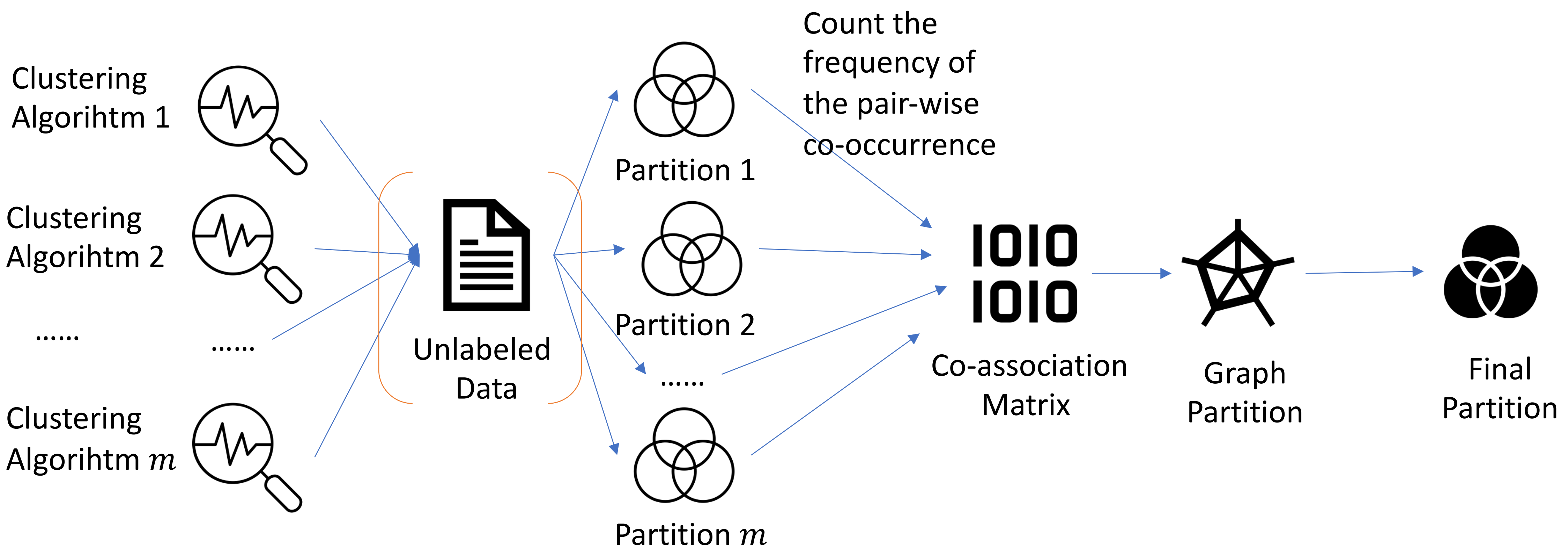
# Ensemble Clustering

If you are not satisfied with one single clustering result, you may use ensemble clustering.




Combine multiple partitions of given data into a single partition of better quality.

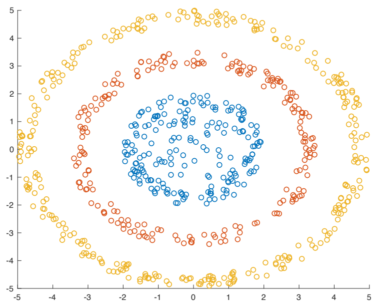
# Widely used ensemble clustering framework



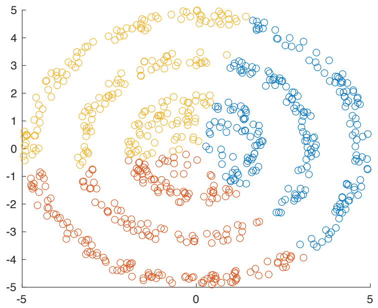

The intrinsic data structures are not fully used.

# Spectral Clustering

- A clustering algorithm:
  - More efficient in finding clusters than traditional algorithms like $k$-means
  - Some studies use the spectral clustering to improve the performance of ensemble clustering.



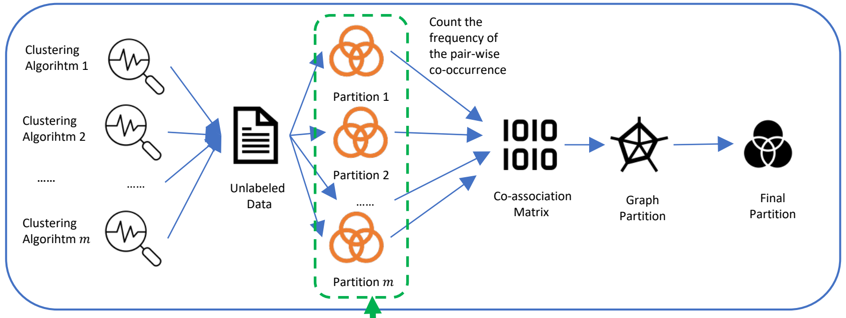
Clustering results
by spectral clustering



Clustering results
by $k$-means

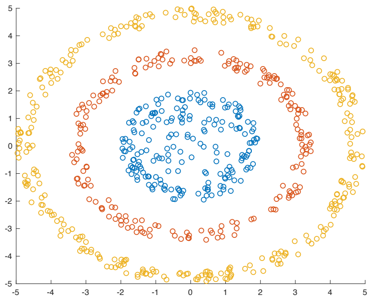
# How does spectral clustering work

Existing methods directly use the clustering results for ensemble learning.



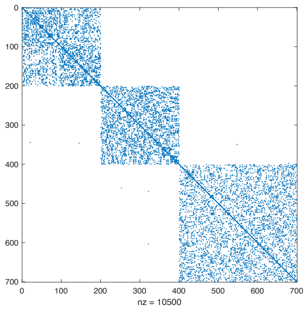

The spectrum of the graph Laplacian is used to reveal the cluster structure.

Original data space

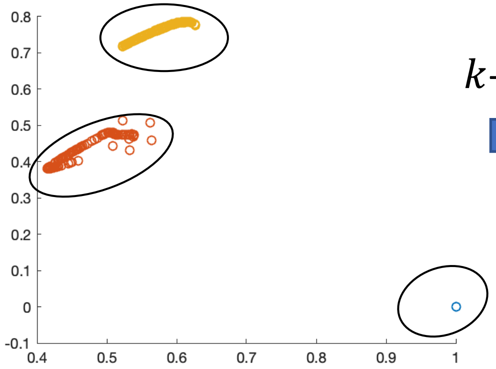


$N$ data points

Graph Laplacian

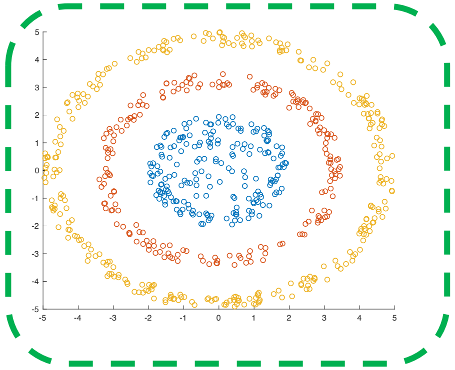



Low dimensional embeddings



$k$-means

Clustering result



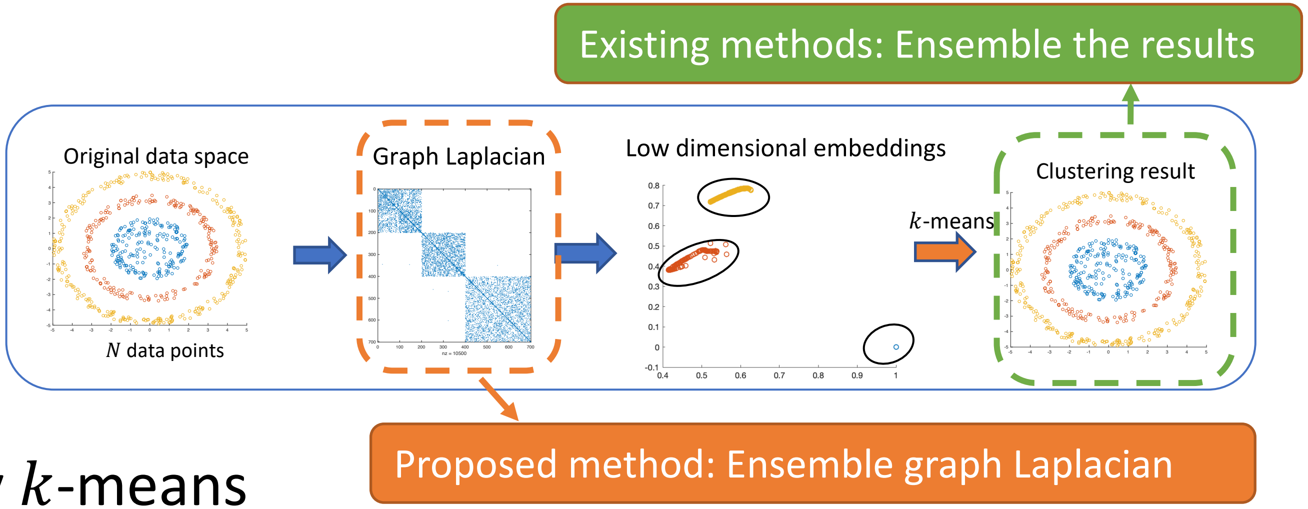
Directly using the clustering results (in the second step) for ensemble clustering may not good enough.

- It cannot make good use of the intrinsic data structures.
- The mis-clustering may be caused due to $k$-means.

# Outlines

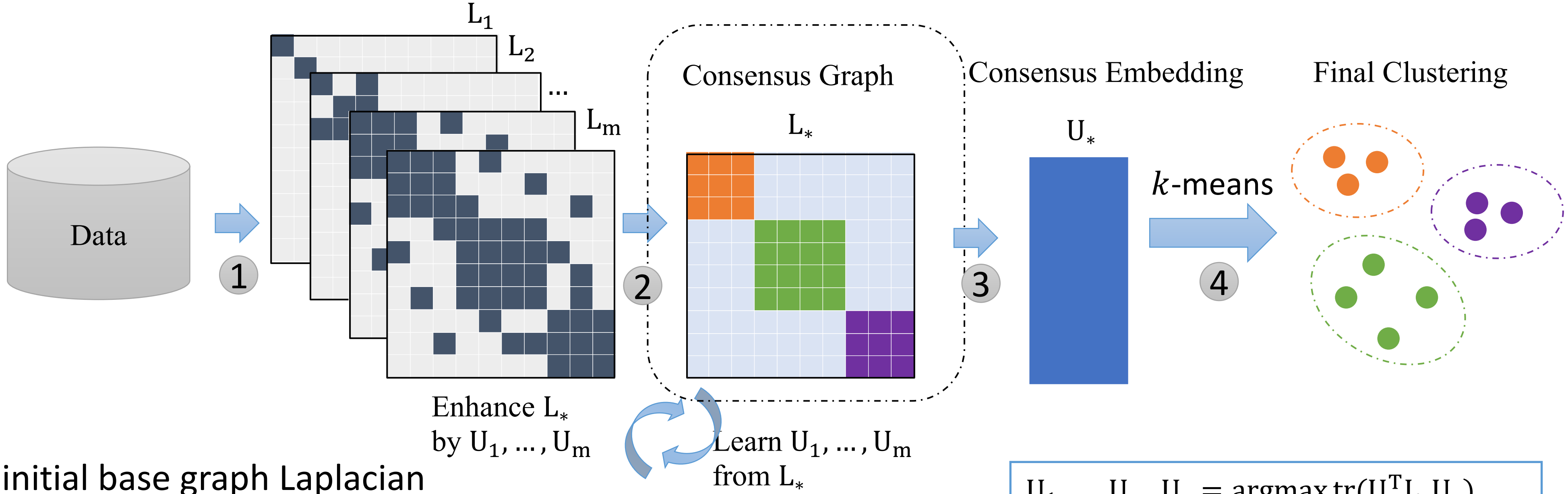
- Introduction
- **Proposed method**
- Experimental results
- Conclusion

# Ensemble learning for spectral clustering



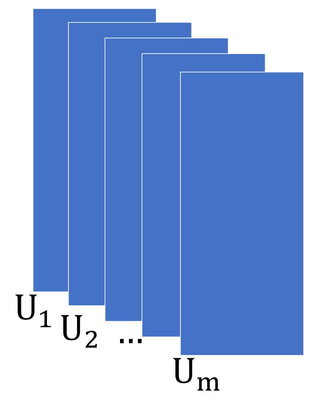

- Objectives
  - Avoiding the mis-clustering caused by $k$-means
  - Learning a robust representation of the graph Laplacian
- Contributions
  - Taking advantage of the intrinsic data structures explored by the graph Laplacians to learn a robust representation
  - Simultaneously optimizing both base graph Laplacian and the consensus graph to improve the final clustering result
  - An adaptive parameter setting to turn the parameter automatically.

# Overview of the propose method




Steps：

1. Computes the initial base graph Laplacian
2. Iteratively optimizes consensus graph Laplacian and base embeddings until convergence to learn a more robust consensus graph
3. Computes consensus embedding
4. Use $k$-means to obtain the final clustering.




$$U_1, \dots, U_m, U_* = \arg\max \mathrm{tr}(U_*^T L_* U_*) + \alpha \sum \mathrm{tr}(U_i^T L_i U_i)$$

$L_* = \sum U_i U_i^T$

$L_1, \dots, L_m$: Base Graph Laplacians

$L_*$: Consensus Graph Laplacians

$U_*$: Consensus Embeddings

# Problem Formulation

- Objective function of base spectral clustering

$$\max_{U} tr(U^T L U), s.t.\ U^T U = I\ ,$$

$L$: the laplacian graph

$U$: the low dimensional represntations of base clustering

- Objective function of consensus Laplacian graph

$$\max_{U_*} tr(U_*^T L_* U_*), s.t.\ U_*^T U_* = I\ ,$$

$L_*$: the consensus laplacian graph

$U_*$: the low dimensional represntations of consensus clustering

# Problem Formulation

- Objective function for the proposed method as follows.

$$\max_{U_1,\ldots,U_m,U_*} \sum_{i=1}^{m} tr\left(U_i^T L_i U_i\right) + \alpha tr\left(U_*^{\mathrm{T}} L_* U_*\right),$$
$$\mathrm{s.t.}\ \ U_i^T U_i = I, \forall 1 \le i \le m;\ \ U_*^T U_* = I,$$

with

$$L_* = \textstyle\sum_{i=1}^{m} U_i U_i^T,$$

$\alpha$: the balanced parameter ($\alpha > 0$ )

# Solution

- The optimization problem is not convex when $U_1, \ldots, U_m$, $U_*$ and $L_*$ are optimized simultaneously.
- We divide the problem into two steps:
  - Update $U_*$ and $L_*$ while fixing $U_1, \ldots, U_m$
    - We solve the following optimization problem.

$$\max_{U_*^{(t+1)}} tr({U_*^{(t+1)}}^T L_*^{(t+1)} U_*^{(t+1)}),$$

$$\text{s.t.} \quad {U_*^{(t+1)}}^T U_*^{(t+1)} = \mathrm{I}.$$

    - This is equivalent to the standard spectral clustering objective function, where $U_*^{(t+1)}$ can be solved as the top $k$ eigenvectors of $L_*$.
  - Update $U_1, \ldots, U_m$ while fixing $U_*$ and $L_*$
    - We solve the following optimization problem.

$$\max_{U_i^{(t+1)}} tr({U_i^{(t+1)}}^T \left(L_i + \alpha U_*^{(t)} {U_*^{(t)}}^T\right) U_i^{(t+1)}),$$

$$\text{s.t.} \quad {U_*^{(t+1)}}^T U_*^{(t+1)} = \mathrm{I}.$$

      - $U_*^{(t+1)}$ can be solved as the top $k$ eigenvectors of $L_i + \alpha U_*^{(t)} {U_*^{(t)}}^T$.
- Repeat those two steps until the objective function converges.

# An adaptive parameter setting

- Parameter $\alpha$ essentially controls how fast the base spectral is optimized, i.e., the learning rate of ensemble learning.
- We design an adaptive parameter setting to turn $\alpha$ automatically in each iteration.

$$\alpha^{(t+1)} = \alpha_0 * std\left\{obj_1^{(t)}, \dots, obj_m^{(t)}\right\}^2,$$

$obj_i^{(t)}$: the sub-objective function

$$obj_i^{(t)} = \begin{cases} tr\left({U_i^{(t)}}^T L_i U_i^{(t)}\right), & t = 1, \\ tr\left({U_i^{(t)}}^T \left(L_i + \alpha U_*^{(t)} {U_*^{(t)}}^T\right) U_i^{(t)}\right), & t \geq 1. \end{cases}$$

# Outlines

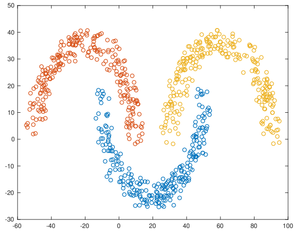
- Introduction
- Proposed method
- **Experimental results**
- Conclusion

# Experimental Setting

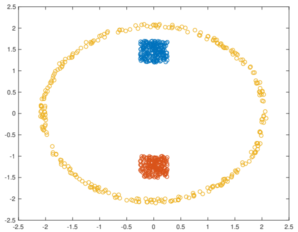
- Datasets

PROPERTIES OF DATASETS.

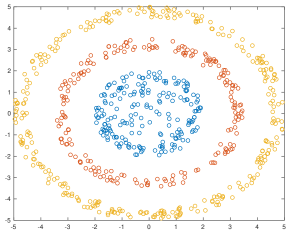
| | Dataset | #Instance | #Feature | #Class |
|---|---|---|---|---|
| Synthetic | 3half-circles | 750 | 2 | 3 |
| | circle-2squares | 700 | 2 | 3 |
| | three-rings | 700 | 2 | 3 |
| Real | pixraw10P | 100 | 10000 | 10 |
| | iris | 150 | 4 | 3 |
| | wine | 178 | 13 | 3 |
| | warpPIE10P | 210 | 2420 | 10 |
| | jaffe | 213 | 676 | 10 |
| | australian | 690 | 14 | 2 |
| | BA | 1404 | 320 | 36 |



(a) half-circle



(b) circle-2squares



(c) three-circles

Three synthetic datasets.

- Compared methods
  - Baseline clustering
    - Spectral clustering (SC)
    - $k$-means clustering
  - Ensemble clustering
    - The evidence accumulation clustering (EAC),
    - Probability trajectory based graph partitioning (PTGP),
    - Ensemble clustering by propagating cluster-wise similarities (ECPCS-MC)
    - Robust spectral ensemble clustering (RSEC),
    - Ultra-scalable spectral clustering based ensemble clustering (USENC).

# Experimental Setting

- Parameter setting
  - Base Clustering Generation
    - Distance metrics
      - Euclidean distance
      - Pearson's linear correlation coefficient
      - Spearman's rho
    - k-nearest-neighborhood: {5, 6, . . . , 9}
  - $\alpha_0$ is set to be 0.001.
- Experimental environment
  - OS: Ubuntu 16.04.5
  - Software: MATLAB 9.4.0 (R2018a)
  - Cpu: Intel CPU E5-1650 v4 @ 3.60GHz (6Core/12Thread, Broadwell)
  - Memory: 16GB x4 = 64GB

# Experimental resutls

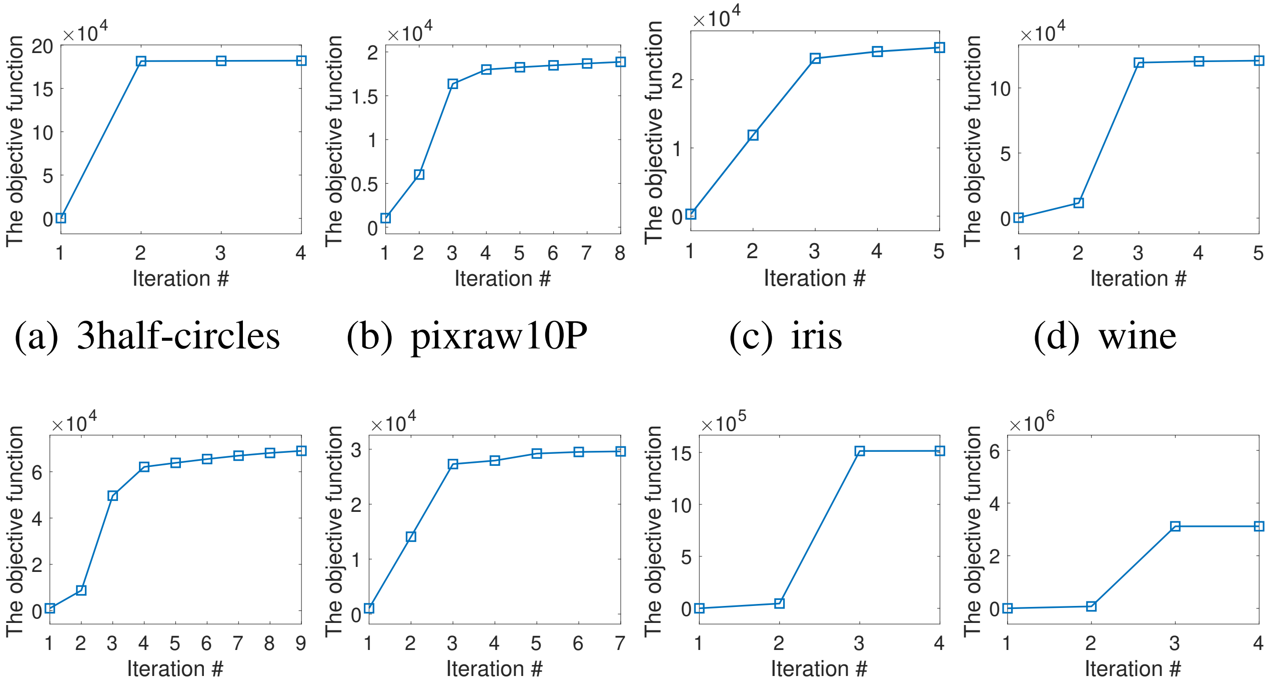
CLUSTERING PERFORMANCE (ACC % ± STD) ON DIFFERENT DATASETS

| Dataset | Baseline Methods | | Ensemble Clustering Methods | | | | | ELSC |
|---|---|---|---|---|---|---|---|---|
| | SC | k-means | EAC | PTPG | ECPCS-MC | RSEC | USENC | |
| 3half-circles | **100.0±0.0** | 88.1±0.4 | **100.0±0.0** | **100.0±0.0** | 99.8±0.4 | 99.8±0.4 | 96.6±1.6 | **100.0±0.0** |
| circle-2squares | **100.0±0.0** | 68.5±0.3 | **100.0±0.0** | 96.7±5.2 | 74.0±1.3 | 71.9±3.8 | 95.4±2.9 | **100.0±0.0** |
| three-circles | **100.0±0.0** | 36.6±0.9 | **100.0±0.0** | 66.9±1.4 | 41.7±6.4 | 38.7±2.8 | 66.4±0.9 | **100.0±0.0** |
| pixraw10P | 93.0±1.0 | 72.3±8.5 | 81.9±5.8 | 10.0±0.0 | 29.3±13.1 | 84.8±5.9 | 94.0±0.0 | **96.0±3.1** |
| iris | 79.0±0.0 | 85.4±10.9 | 80.2±4.6 | 91.5±3.0 | 90.0±0.1 | 90.0±0.5 | 90.7±0.0 | **97.7±0.1** |
| wine | 70.2±0.0 | 65.7±6.3 | 66.0±6.5 | 71.5±5.2 | 71.6±1.3 | 68.5±4.3 | 71.3±0.0 | **75.8±0.0** |
| warpPIE10P | 32.10±1.9 | 25.4±2.3 | 32.5±3.3 | 45.6±2.9 | 38.5±2.9 | 41.1±2.7 | 31.1±1.6 | **53.4±3.3** |
| jaffe | 96.2±0.0 | 73.1±11.2 | 72.4±5.9 | 94.2±5.1 | 84.4±6.8 | 94.4±4.6 | 93.0±2.3 | **98.6±0.0** |
| australian | 56.8±0.0 | 56.1±0.2 | 55.7±0.4 | 56.2±0.1 | 56.1±0.2 | 56.1±2.1 | 67.3±0.2 | **70.9±0.1** |
| BA | 44.4±1.7 | 41.2±1.7 | 5.5±1.1 | 48.9±1.5 | 50.9±0.7 | 48.3±1.3 | 49.4±1.6 | **51.8±2.3** |

The top performance is highlighted by bold font.

ELSC outperforms compared methods significantly.

# Convergence



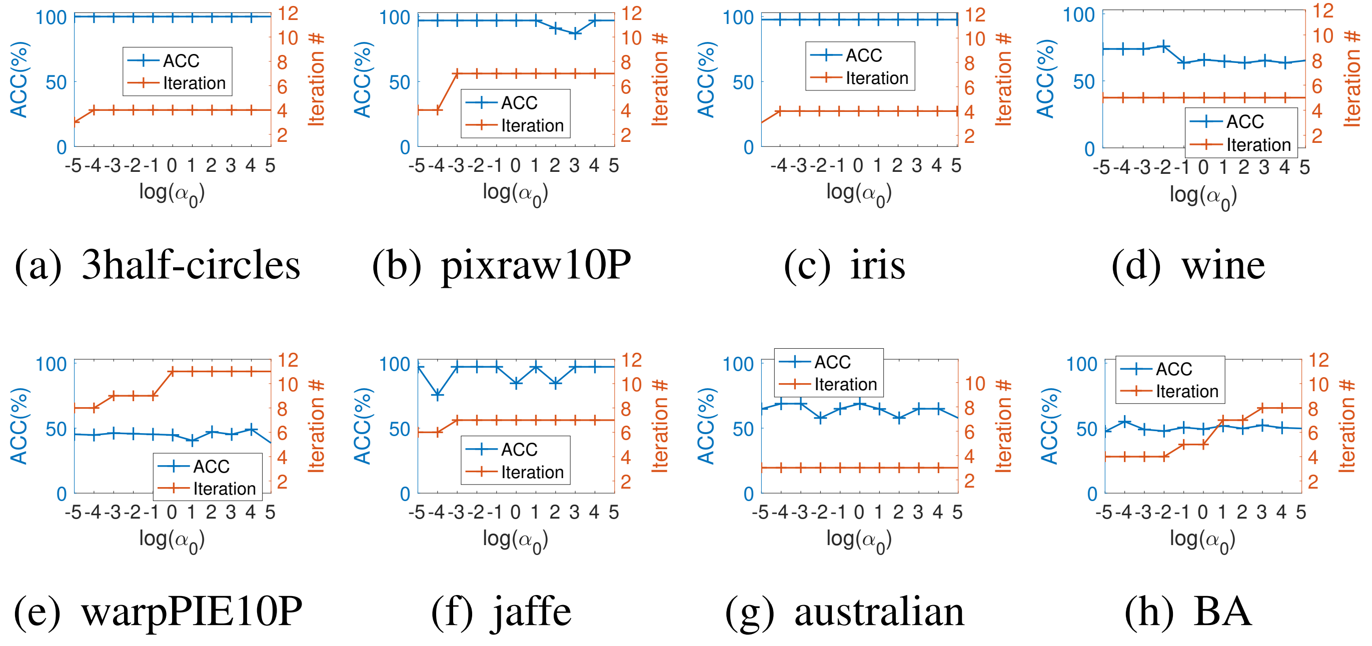

(a) 3half-circles (b) pixraw10P (c) iris (d) wine

(e) warpPIE10P (f) jaffe (g) australian (h) BA

Curves of objective function (3) vs. iteration number. The proposed ELSC method converges within ten iterations on all eight datasets.

The proposed method converges within ten iterations on all eight datasets.

# Parameter analysis




ACC (blue line) and iteration number (orange line) curves with respect to the parameter settings of $\log(\alpha_0)$. The performance of the proposed method shows stability and efficiency with parameter $\alpha_0$ in the range of $[10^{-4}, 10^{-3}]$. The proposed method shows its fast convergence and stability.

The performance of the proposed method show stability and efficiency with a similar range of $[10^{-4}, 10^{-3}]$ on most datasets.

# Outlines

- Introduction
- Proposed method
- Experimental results
- **Conclusion**

# Conclusion

- We proposed an effective ensemble learning method for spectral clustering
  - Optimizing the consensus graph Laplacian and base spectral clustering simultaneously to obtain a more robust representation
  - An adaptive parameter setting to turn the parameter automatically
- Experiments on three synthetic and seven real-world datasets show the effectiveness of the proposed method compared with five state-of-the-art ensemble clustering methods.