REVISITING FAST SPECTRAL CLUSTERING WITH ANCHOR GRAPH

Cheng-Long Wang* Feiping Nie* Rong Wang† Xuelong Li*

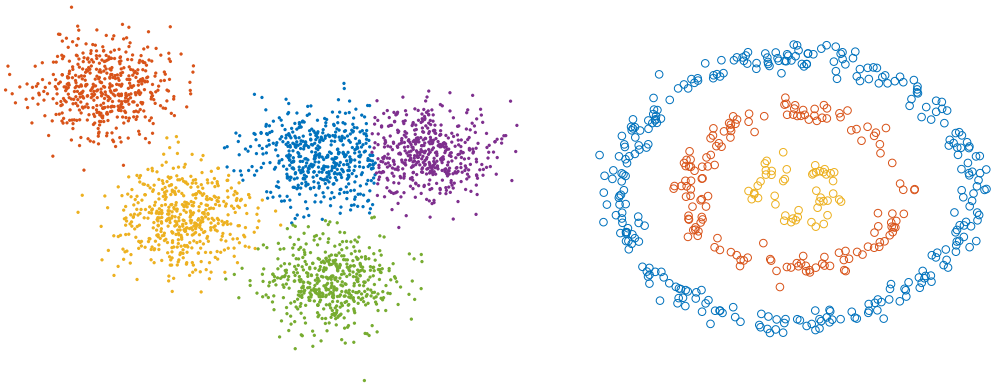
ch.l.w.reason@gmail.com

*School of Computer Science and Center for OPTIMAL, Northwestern Polytechnical University

†School of Cybersecurity and Center for OPTIMAL, Northwestern Polytechnical University

Introduction

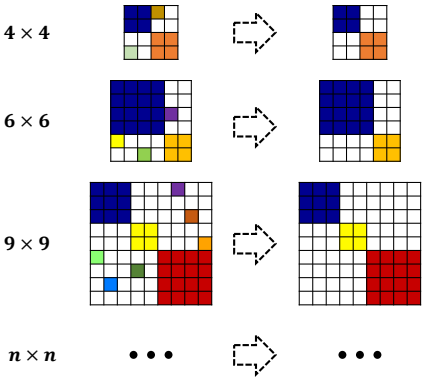
Spectral Clustering



Finding groups of highly similar objects in unlabeled datasets

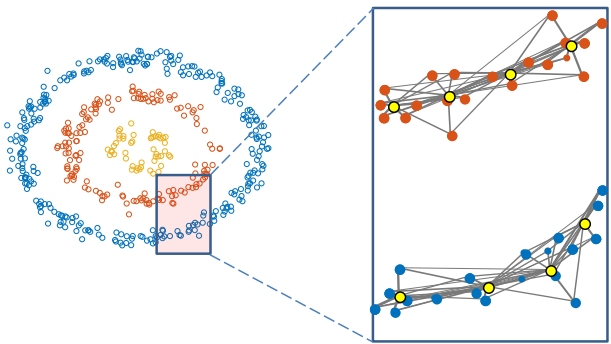
When it comes to large-scale data · · ·

- However, when it comes to large-scale data, the Spectral Clustering methods based on data graph are computationally prohibitive.
- The time complexity of Spectral Clustering is $O(n^3)$.



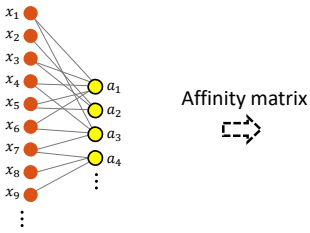
Anchors: representative data point

- A small number of anchor points often adequately cover the entire point cloud. K-means, BKHK, · · ·

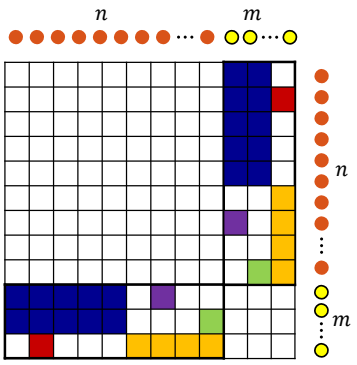


Anchors: representative data point

- The clustering problem based on data graph can be transformed into the optimization problem related to the bipartite graph.
- Given the data point set $\mathcal{X}$, the anchors set $\mathcal{A}$ and the edges set $\mathcal{E}$ where the data points belong to c clusters, a bipartite graph can be denoted by $\mathcal{B}(\mathcal{X}, \mathcal{A}, \mathcal{E})$.



Affinity matrix



Bipartite graph $B = (X, A, E)$,
Where $X = \{x_1, x_2, x_3, \dots, x_n\}$,
$A = \{a_1, a_2, \dots, a_m\}$

Anchors: representative data point

- Denote $Z \in \mathbb{R}^{n \times m}$ as the cross similarity matrix between data points and anchors. Thus the full adjacency matrix for the bipartite graph $\mathcal{B}$ can be denoted by $B = \begin{bmatrix} & Z \\ Z^T & \end{bmatrix}$, where $Z\mathbf{1} = \mathbf{1}$.
- A theoretical analysis of the relationship between $W = Z\Delta^{-1}Z^T$ and random walk provided by Liu et al.[1], where $\Delta = diag(Z^T\mathbf{1})$ balances the popularity of the anchors.
- However, due to the special structure of the bipartite graph, there is no stable distribution of the random walk process. The designed similarity matrix W may resulting in breaking the independence of data points and leading to undesired artifacts for boundary samples.

[1] W. Liu, J. He, and S. Chang, "Large graph construction for scalable semi-supervised learning," in Proc. ICML, pp. 679–686, 2010.

Problem

How to balance the popularity of the anchors and the independence of the data points explicitly?

Our Method

Cross Similarity Matrix

The cross similarity matrix Z can be calculated as follows:

- For i-th data point x_i , $z_{ij} > 0$ if i-th point x_i and j-th anchor a_j is connected, otherwise $z_{ij} = 0$.
- Define the distance between x_i and a_j as $h(x_i, a_j) = \|x_i - a_j\|_2^2$. Denote a sort function $\hat{h}_i = \theta(h_i)$ which sorts the distance in ascending order. Following [2], z_{ij} can be computed by

$$z_{ij} = \frac{\hat{h}\left(x_i, u_{k+1}\right) - \hat{h}\left(x_i, u_j\right)}{\sum_{j'=1}^{k}\left(\hat{h}\left(x_i, u_{k+1}\right) - \hat{h}\left(x_i, u_{j'}\right)\right)}. \tag{2.1}$$

[2] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in Proc. KDD, pp. 977–986, 2014.

Symmetric Normalized Laplacian

- Denote $L_{sym} = D^{-\frac{1}{2}} B D^{-\frac{1}{2}}$, the normalized spectral clustering problem on B can be written as follows

$$\begin{aligned} &\min_{F} \ Tr\left(F^T L_{sym} F\right) \\ &s.t. \ \ F \in \mathbb{R}^{(n+m)\times c}, F^T F = I. \end{aligned} \tag{2.2}$$

Theorem

Define the one-step transition probability matrix as $P = D^{-1}B$, the spectral embedding of data points based on the symmetric normalized Laplacian matrix of B is equivalent to the spectral embedding on the data similarity matrix $W \in \mathbb{R}^{n\times n}$ obtained using the second-order transition probabilities.

Our Method

- Define the random walk normalized graph Laplacian as $L_{rw} = D^{-1}(D-B) = I - D^{-1}B$. Due to $D^{-1}B$ is not symmetrical, the spectral embedding of the graph corresponding to the random walk transition matrix can be obtained by solving the problem as follows:

$$\begin{aligned} &\min_{F} Tr\left(F^T\left(I - \frac{D^{-1}B + BD^{-1}}{2}\right)F\right). \\ &\text{s.t. } F \in \mathbb{R}^{(n+m)\times c}, F^TF = I \end{aligned} \tag{2.3}$$

- The problem (2.4) seems complicated. While due to the special structure of the block matrix of the bipartite graph, it can be reformulated as follows.

Our Method

- The degree matrix of B can be written in the form of block matrix $D = \begin{bmatrix} D_U & \\ & D_V \end{bmatrix}$, where $d_{ii} = \sum_{j=1}^{n+m} b_{ij}$, $D_U \in \mathbb{R}^{n\times n}$, $D_V \in \mathbb{R}^{m\times m}$. Rewrite F as the block matrix

$$F = \begin{bmatrix} U \\ V \end{bmatrix}, \tag{2.4}$$

  where $U \in \mathbb{R}^{n\times c}$, $V \in \mathbb{R}^{m\times c}$.

- According to the definition of F in Eq (2.4) and the structure of L_{rw} which is also a block matrix, the problem (2.3) can be further rewritten as

$$\max_{U,V} Tr\left(U^T Z\left(\frac{I + D_V^{-1}}{2}\right)V\right). \tag{2.5}$$
$$\text{s.t. } U \in \mathbb{R}^{n\times c},\ V \in \mathbb{R}^{m\times c},\ U^T U + V^T V = I$$

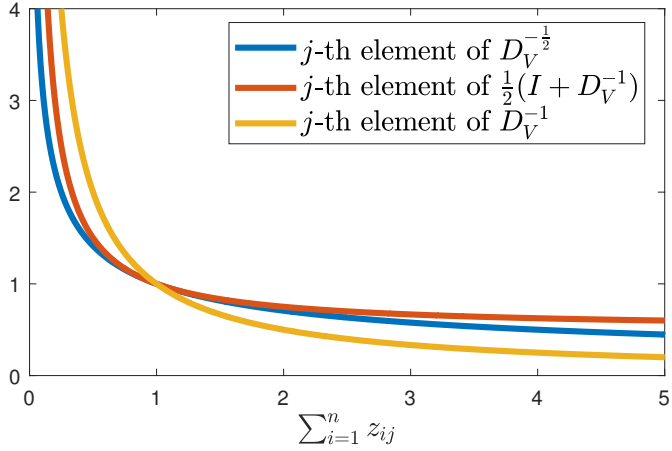


Figure 1: Comparison of three column normalization methods

Our Method

Algorithm 1: Fast Spectral Clustering based on RandomWalk Laplacian (FRWL)

Input : Data matrix $X \in \mathbb{R}^{d \times n}$, Anchor matrix $A \in \mathbb{R}^{d \times m}$, cluster number c, number of nearest neighbor k

Output: c clusters

1 Construct a sparse cross similarity matrix $Z \in \mathbb{R}^{n \times m}$ between data points and anchors, with the similarity calculated according to Eq. (2.1).

2 Compute $F = \begin{bmatrix} U^* \\ V^* \end{bmatrix}$ which is formed by the top c left and right singular vectors of $Z\left(\frac{I+D_V^{-1}}{2}\right)$ according to Eq. (2.5) respectively.

3 Each row of U^* is a data point and apply k-means to get the clusters.

Our Method

Matlab Code[4]

```
% Input:    - Z: the initial cross similarity matrix between data ...
      points and anchors
%           - c: the number of clusters
% Output:   - clustering: the cluster assignment for each point

Dv=diag(1./sum(Z,1))
U = mySVD(Z+Z*Dv,c+1);
U(:,1) = [];
U=U./repmat(sqrt(sum(U.^2,2)),1,c);
clustering=litekmeans(U,c,'MaxIter',100,'Replicates',10);
```

[4]https://github.com/CHLWR/ICASSP_FRWL

Experiments

Experiments on Real-world Datasets

Table 1: Statistic of Datasets

Dataset	Samples	Features	Classes
USPS20	1854	256	10
Coil20	1440	1024	20
Palm	2000	256	100
Coil100	7200	1024	100
USPS	9298	256	10
Mnist	70000	784	10

Experiments on Real-world Datasets

Table 2: Performance (ACC) on 6 data sets(%)

Dataset	Nystrom	LSC-K	FSC	FRWL-K	FRWL-B
USPS20	66.03	68.68	65.43	65.49	**70.81**
Coil20	52.76	68.40	67.57	68.61	**73.09**
Palm	75.87	70.35	76.40	74.44	**76.90**
Coil100	41.19	51.55	54.85	**55.13**	55.01
USPS	65.78	69.12	**70.90**	70.80	68.64
Mnist	34.51	63.93	62.60	62.64	**68.04**

Experiments on Real-world Datasets

Table 3: Performance (NMI) on 6 data sets(%)

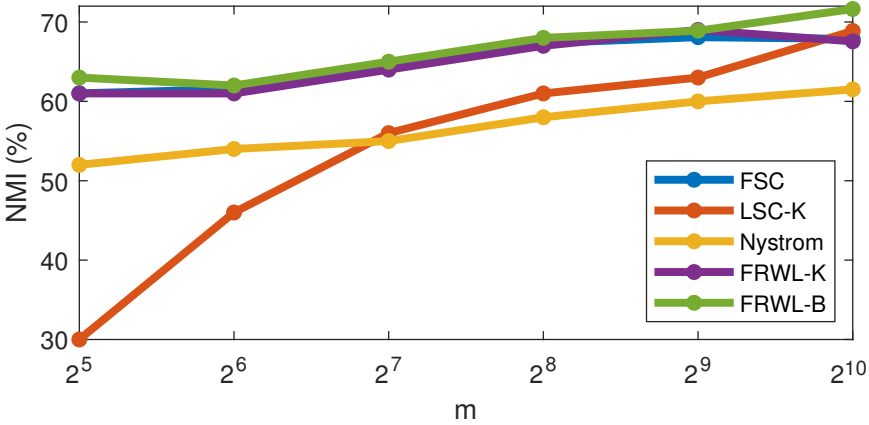
Dataset	Nystrom	LSC-K	FSC	FRWL-K	FRWL-B
USPS20	65.73	67.79	63.39	63.57	**72.57**
Coil20	71.26	77.78	78.05	78.14	**82.33**
Palm	91.83	87.82	91.41	90.77	**91.88**
Coil100	68.51	75.38	77.29	77.46	**78.11**
USPS	61.51	68.87	67.87	67.56	**71.64**
Mnist	24.03	62.51	59.01	59.85	**67.28**

Experiments on Real-world Datasets

Table 4: Running time on 6 data sets(s)

Dataset	Nystrom	LSC-K	FSC	FRWL-K	FRWL-B
USPS20	1.69	0.26	0.41	0.11	0.30
Coil20	1.53	0.26	0.99	0.11	0.31
Palm	1.18	0.63	0.83	0.40	0.56
Coil100	2.03	3.29	9.02	3.20	3.49
USPS	2.40	0.86	2.54	0.78	1.19
Mnist	6.44	11.39	67.39	14.72	13.70

Experiments on Real-world Datasets



Parameter sensitivity study of FRWL on USPS

Conclusion

Fast Spectral Clustering based on the Random Walk Laplacian (FRWL) can explicitly balance the popularity of anchors and the independence of data points, which is important for clustering of boundary points (especially when there are many neighbor anchors).

Thanks for your attention

Presenter: Cheng-Long Wang