K-Multiple-Means: A Multiple-Means Clustering Method with Specified K Clusters

Feiping Nie
feipingnie@gmail.com

Cheng-Long Wang
ch.l.w.reason@gmail.com

Xuelong Li
li@nwpu.edu.cn

School of Computer Science and Center for OPTIMAL,
Northwestern Polytechnical University, Xi'an, China

Background

Why Multiple Means?

K-means: each point is assigned to its nearest prototype (yellow "$\star$").

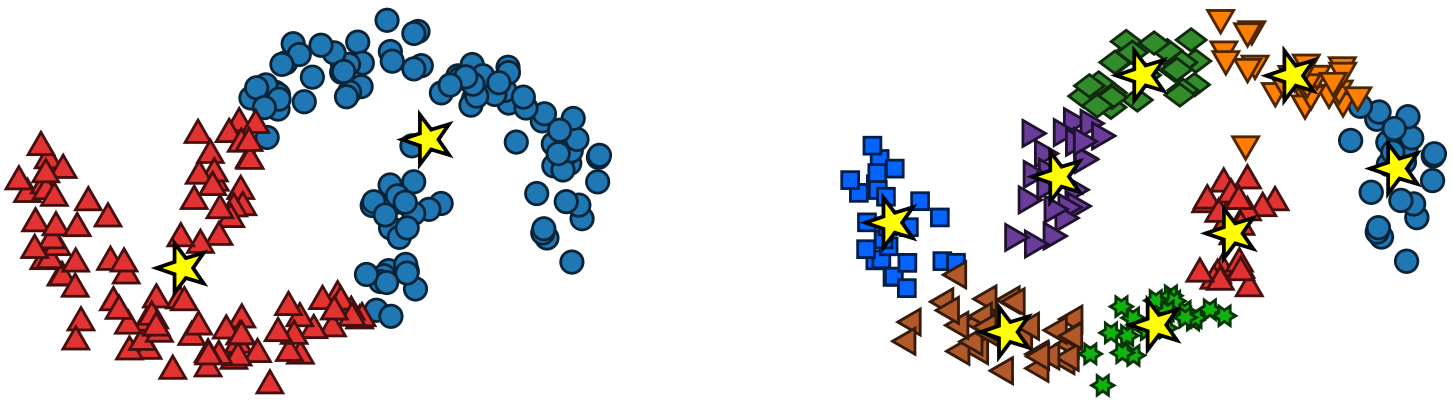
The squared error criterion of the K-means-type algorithms prohibits the algorithms to capture the non-convex patterns

$$f(U, V) = \sum_{i=1}^{n} \sum_{l=1}^{k} u_{il}^{r} \|x_i - v_l\|_2^2$$

$$s.t. \;\; U \geq 0, U\mathbf{1} = \mathbf{1}, U \in R^{n \times k}, V \in R^{k \times d}.$$

Why Multiple Means?

- Kernel-based clustering or spectral clustering, although capable of producing nonlinear separating hyperplanes, can not give prototypes for the clusters in the data space.
- The methods based on multiple means can not only capture non-convex patterns but also give multiple prototype of the clusters.



single-prototype multi-prototypes

Figure 1: Each point is assigned to its nearest prototype (yellow "⋆").

How to find Multiple-Means?

- Most multi-prototypes clustering algorithms consist of a splitting stage and a merging stage based on agglomerative strategies.
- Those algorithms use agglomerative strategies which often encounter difficulties regarding the selection of merge or split points and have very high time complexity.

Comparison of the Multi-Prototypes Clustering Algorithms

	Split Stage	Merge Stage
Tao's	Hierarchical subtractive clustering	The centers will be assigned to the same cluster when the density of the regions between the two centers is greater than 1/4 of the density of the two sub-clusters.
Liu et al. 's	Squared-error clustering	The prototypes who coexist in a high-density region are grouped into one cluster.
Luo et al. 's	Minimum spanning tree	1) The prototypes whose distance is smaller than the user-specified threshold are roughly merged. 2) Further merge step will be conducted based on the data distribution between two clusters prototypes.
Ben et al. 's	1) Fuzzy C-means 2) Iteratively 2-partition the subclusters based on the intra-cluster non-consistency value	The subclusters with the largest inter-cluster overlap are iteratively merged until a pre-determined cluster number is achieved.
Liang et al. 's	1) Fast Global Fuzzy K-means 2) Best-M Plot	The grouping multicenter (GMC) algorithms based on the degree of overlap between two clusters is used to group the cluster centers to represent k clusters.

How to find Multiple-Means?

The key to obtaining a specified number of clusters is to judge the connection relationships between multi-prototypes.

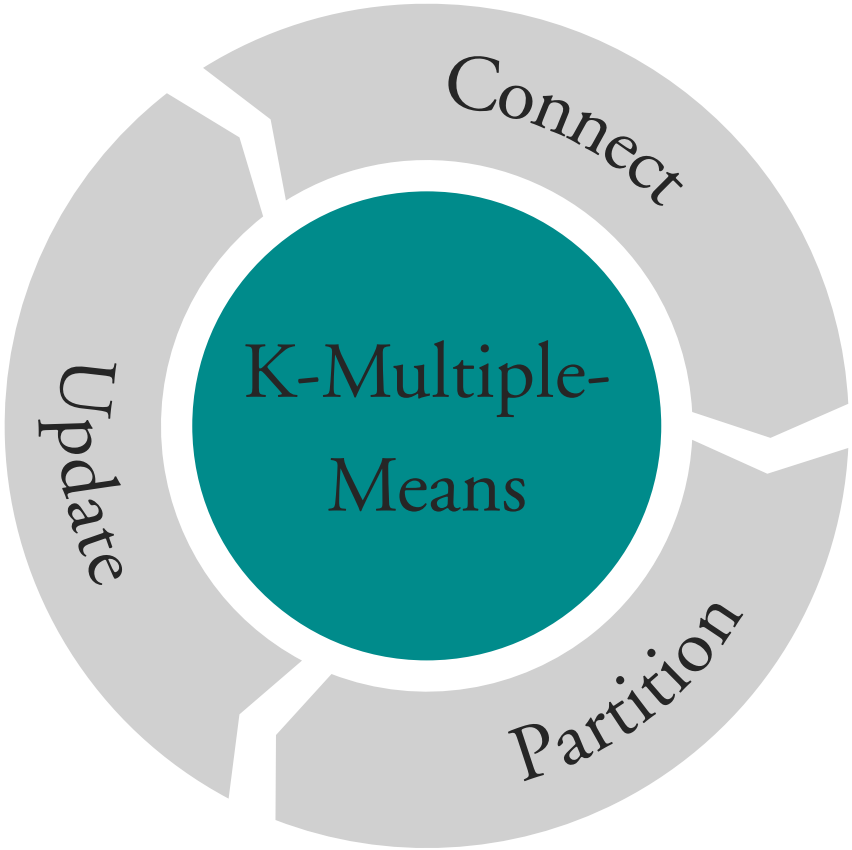
Split or Merge ?

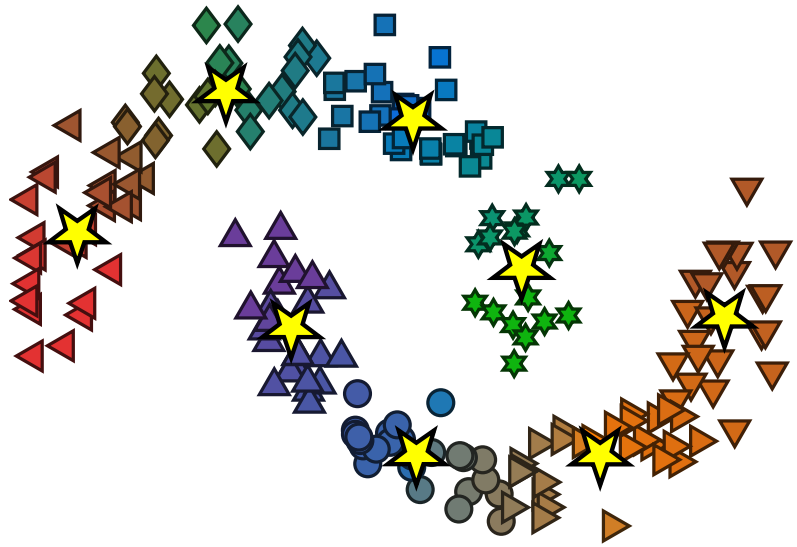
How to find Multiple-Means?

- Each object is allowed to have memberships in its neighboring sub-clusters rather than having a distinct membership in one single sub-cluster.
- The partition should be based on both the distribution of multiple prototypes and the distribution of data points.
- The algorithm can update the assignment iteratively, no matter it is the sub-cluster assignment for each data point or the cluster assignment for each prototype.

K-Multiple-Means

K-Multiple-Means(KMM)





- Connect: Neighboring Prototypes
- Partition: Clustering based on rank constraint
- Update: Weighted means of corresponding points

Neighboring Prototypes Assignment

- Suppose we need to partition n data points into k clusters with totally m prototypes (weighted means of the corresponding points).
- For the i-th data point $\mathbf{x}_i$, the j-th prototype $\mathbf{a}_j$ can be connected to $\mathbf{x}_i$ as a neighboring prototype with probability s_{ij}.
- The smaller the distance, such as $\|\mathbf{x}_i - \mathbf{a}_j\|_2^2$, between $\mathbf{x}_i$ and $\mathbf{a}_j$ is, the greater the corresponding connection probability s_{ij} is.
- The sparsity of the connection of data points to multiple prototypes should also be controled by the regularization term.

Neighboring Prototypes Assignment

The assignment problem of n data points' neighboring prototypes based on the weighted squared error criterion can be written as

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij}\left\|\mathbf{x}_i - \mathbf{a}_j\right\|_2^2 + \gamma\left\|\mathbf{S}\right\|_F^2 \\ &s.t.\ \mathbf{S} \geq 0, \mathbf{S1} = \mathbf{1} \end{aligned} \tag{2.1}$$

- When $\gamma = 0$, only the nearest prototype can be connected to $\mathbf{x}_i$ with probability $s_{ij} = 1$.
- When γ is large enough, all m prototypes can be connected to $\mathbf{x}_i$ with the same probability $\frac{1}{m}$.

Neighboring Prototypes Assignment

- For each data point $\mathbf{x_i}$, the assignment of neighbors is independent.
- Denote $d_{ij}^x = \|\mathbf{x}_i - \mathbf{a}_j\|_2^2$ and denote $\mathbf{d}_i^x$ as a vector with the j-th element as d_{ij}^x (same for $\mathbf{s}_i$). The assignment of neighboring prototypes for $\mathbf{x}_i$ can be written in vector form as

$$\begin{aligned} &\min_{\mathbf{s}_i} \left\| \mathbf{s}_i - \frac{\mathbf{d}_i^x}{2\gamma} \right\|_2^2 \\ &s.t.\ \mathbf{s}_i \geq 0, \mathbf{s}_i^T \mathbf{1} = 1. \end{aligned} \tag{2.2}$$

- Consider the locality of data, γ can be controlled by the number of neighbor prototypes $\tilde{k}$ which is specified manually. Thus, the problem can be solved with a closed-form solution.

Prototypes Update

- When $\mathbf{S}$ is updated, each prototype can be relocated to the mean of all data points assigned to it respectively. For j-th prototype, $\mathbf{a}_j$ can be updated by

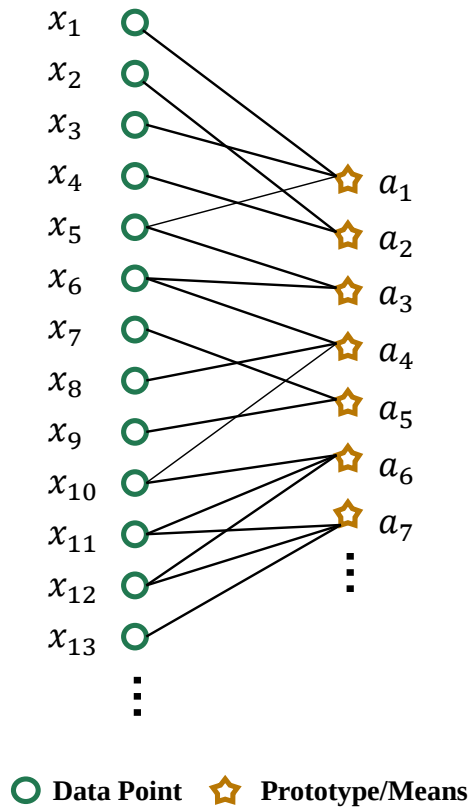
$$\mathbf{a}_j = \frac{\sum_{i=1}^{n} s_{ij}\mathbf{x}_i}{\sum_{i=1}^{n} s_{ij}}. \tag{2.3}$$

- This process can be iteratively performed to obtain an optimal multi-mean neighbor assignment until the assignment is not updated

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{A}} \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij} \left\|\mathbf{x}_i - \mathbf{a}_j\right\|_2^2 + \gamma \left\|\mathbf{S}\right\|_F^2 \\ &s.t.\ \mathbf{S} \geq 0, \mathbf{S1} = \mathbf{1}, \mathbf{A} \in \mathbb{R}^{m\times d}. \end{aligned} \tag{2.4}$$

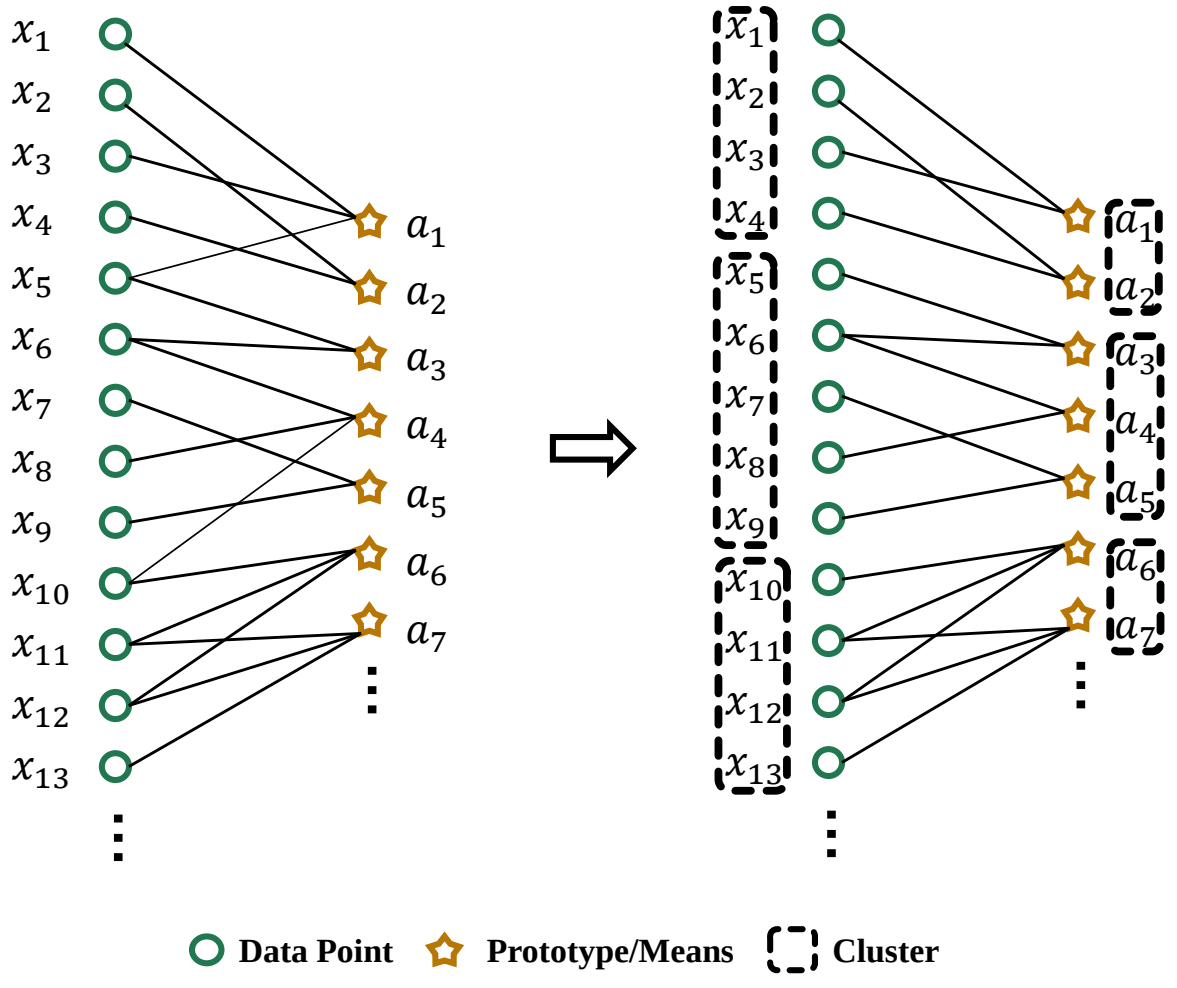
A K-Multiple-Means Problem

- However, the neighbor assignment with Eq.(2.4) connects n data points and m prototypes as just one connected component in most cases.
- An appropriate constraint $\mathbf{S} \in \Omega$ that the bipartite graph associated with $\mathbf{S}$ has exactly k connected components should be imposed on the objective function.



A K-Multiple-Means Problem

- In each iteration, the bipartite graph corresponding to $\mathbf{S}$ should be partitioned to k connected components.



A K-Multiple-Means Problem

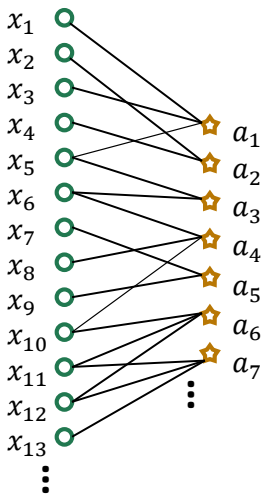
The K-Multiple-Means problem can be writted as:

$$\min_{\mathbf{S},\mathbf{A}} \sum_{i}^{n} \sum_{j}^{m} s_{ij} \left\| \mathbf{x}_i - \mathbf{a}_j \right\|_2^2 + \gamma \left\| \mathbf{S} \right\|_F^2 \\ s.t.\ \mathbf{S} \geq 0, \mathbf{S1} = \mathbf{1}, \mathbf{S} \in \Omega, \mathbf{A} \in \mathbb{R}^{m \times d} . \tag{2.5}$$

Reformulation of Eq.(2.5)

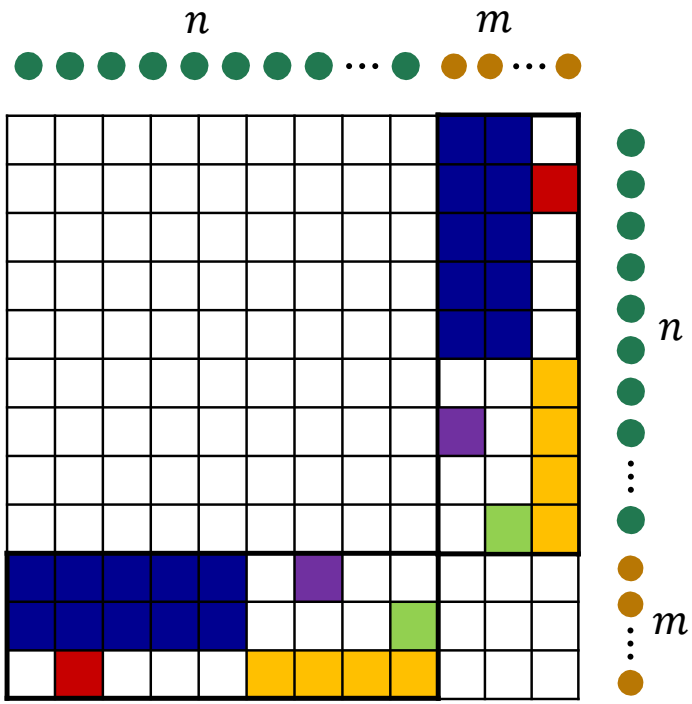
- The constraint $\mathbf{S} \in \Omega$ in Eq.(2.5) is very difficult to tackle. We will reformulate it to make it solvable.
- Denote the affinity matrix associated with the bipartite graph

$$\mathbf{P} = \begin{bmatrix} & \mathbf{S} \\ \mathbf{S^T} & \end{bmatrix}.$$



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

Affinity matrix



Reformulation of Eq.(2.5)

- Denote the normalized Laplacian matrix $\mathbf{L_S}$ associated with $\mathbf{S}$ as $\mathbf{L_S = I - D_S^{-\frac{1}{2}} P D_S^{-\frac{1}{2}}}$, where $\mathbf{D_S}$ is defined as a diagonal matrix where the i-th diagonal element is $d_{ii} = \sum_j s_{ij}$. We have the following theorem:

Theorem 1

The multiplicity k of the eigenvalue 0 of the normalized Laplacian matrix $\mathbf{L_S}$ is equal to the number of connected components in the bipartite graph associated with $\mathbf{S}$.

Reformulation of Eq.(2.5)

- According to the theorem, Eq.(2.5) is equivalent to the following problem:

$$\min_{\mathbf{S},\mathbf{A}} \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij}\left\|\mathbf{x}_i - \mathbf{a}_j\right\|_2^2 + \gamma\left\|\mathbf{S}\right\|_F^2$$
$$s.t.\ \mathbf{S} \geq 0, \mathbf{S1} = \mathbf{1}, rank(\mathbf{L_S}) = (n+m) - k, \mathbf{A} \in \mathbb{R}^{m\times d}. \tag{2.6}$$

- However, the rank constraint in Eq.(2.6) is also difficult to handle. We need further reformulation.

Reformulation of Eq.(2.5)

- Suppose $\sigma_i(\mathbf{L_S})$ is the k-th smallest eigenvalue of $\mathbf{L_S}$, we know $\sigma_i(\mathbf{L_S}) \geq 0$ since $\mathbf{L_S}$ is positive semi-definite. It can be seen that the problem (2.6) is equivalent to the following problem for a large enough value of λ:

$$\min_{\mathbf{S},\mathbf{A}} \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij}\left\|\mathbf{x}_i - \mathbf{a}_j\right\|_2^2 + \gamma\left\|\mathbf{S}\right\|_F^2 + \lambda \sum_{i=1}^{k} \sigma_i(\mathbf{L_S}) \tag{2.7}$$
$$s.t.\ \mathbf{S} \geq 0, \mathbf{S1} = \mathbf{1}, \mathbf{A} \in \mathbb{R}^{m\times d}.$$

- However, the second term in Eq.(2.7) is also difficult to handle. We need further reformulation.

Reformulation of Eq.(2.5)

- According to the Ky Fan's Theorem, we have

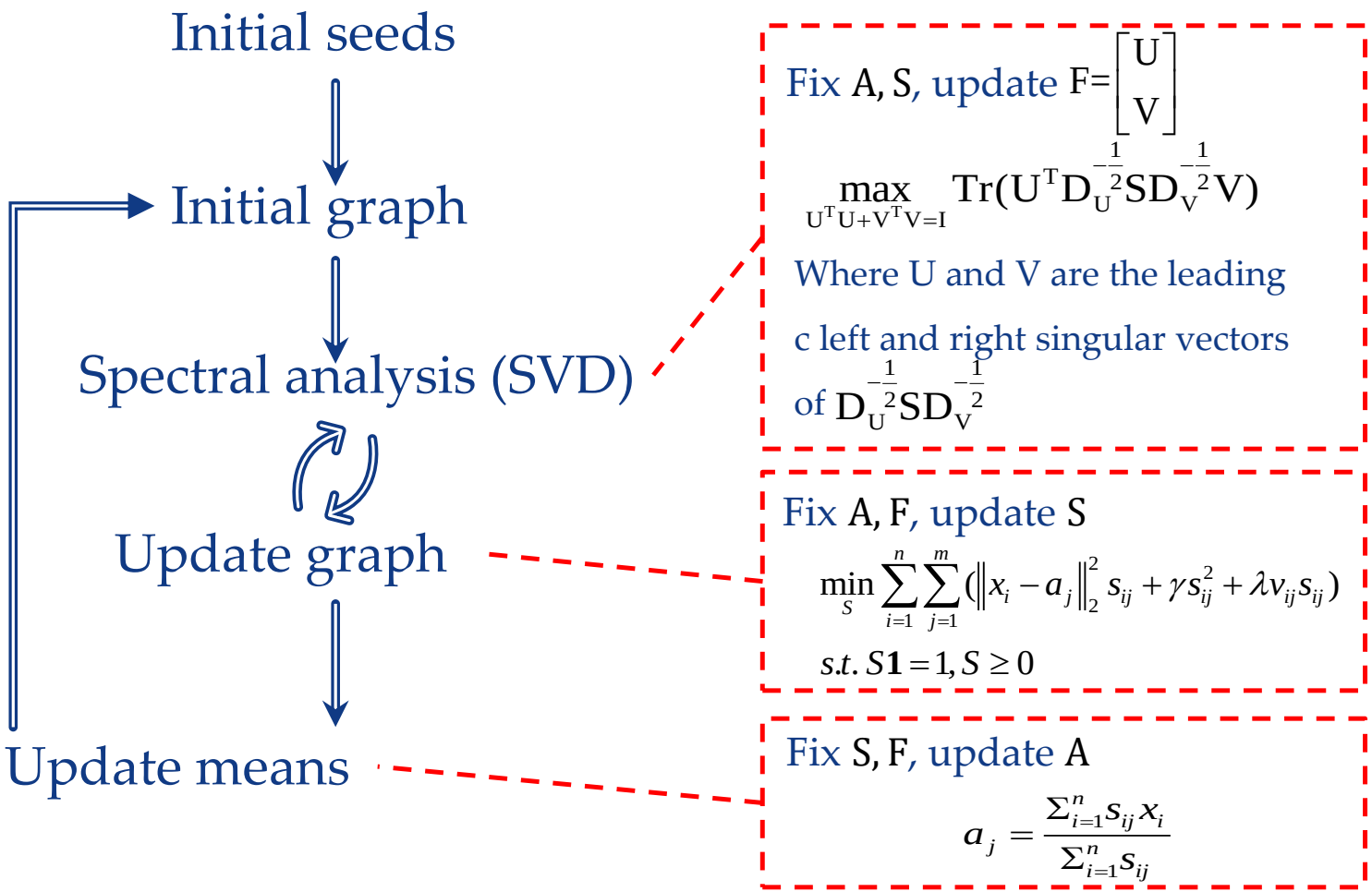
$$\sum_{i=1}^{k} \sigma_i(\mathbf{L_S}) = \min_{\mathbf{F} \in \mathbb{R}^{(n+m)\times k}, \mathbf{F^T F} = \mathbf{I}} \mathbf{Tr}(\mathbf{F^T L_S F}) \tag{2.8}$$

- Therefore, the problem (2.7) is further equivalent to the following problem:

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{A},\mathbf{F}} \sum_{i}^{n} \sum_{j}^{m} s_{ij} \left\| \mathbf{x}_i - \mathbf{a}_j \right\|_2^2 + \gamma \left\| \mathbf{S} \right\|_F^2 + \lambda Tr(\mathbf{F^T L_S F}) \\ &s.t.\ \mathbf{S} \geq 0, \mathbf{S1} = \mathbf{1}, \mathbf{A} \in \mathbb{R}^{m \times d}, \mathbf{F} \in \mathbb{R}^{(n+m)\times k}, \mathbf{F^T F} = \mathbf{I}. \end{aligned} \tag{2.9}$$

- Compared with the original problem (2.5), the problem (2.9) is **much easier** to solve.

Iterative Optimization



Connection to K-means Clustering

It looks the problem of KMM is far away from the problem of K-means Clustering. However, we have the following theorem:

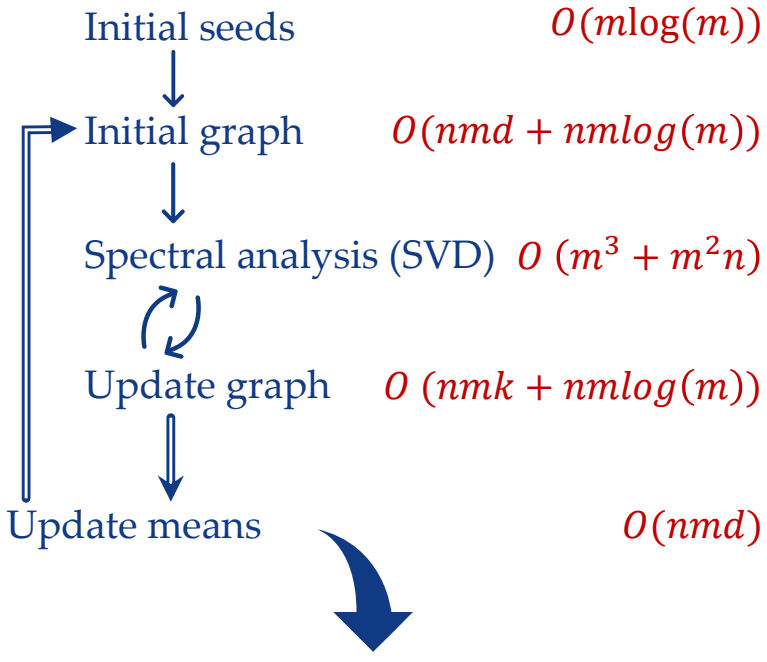
Theorem 2

When $\gamma \to \infty$,

$$\min_{\mathbf{S}\geq\mathbf{0},\mathbf{S1}=\mathbf{1},\mathbf{S}\in\Omega,\mathbf{A}} \sum_{i}^{n}\sum_{j}^{m} s_{ij}\left\|\mathbf{x}_i-\mathbf{a}_j\right\|_2^2+\gamma\left\|\mathbf{S}\right\|_F^2$$

is closely related to the problem of K-means, where the constraint Ω denotes that the bipartite graph $\mathcal{P}=(\mathbf{X},\mathbf{A},\mathbf{S})$ has exactly k connected components.

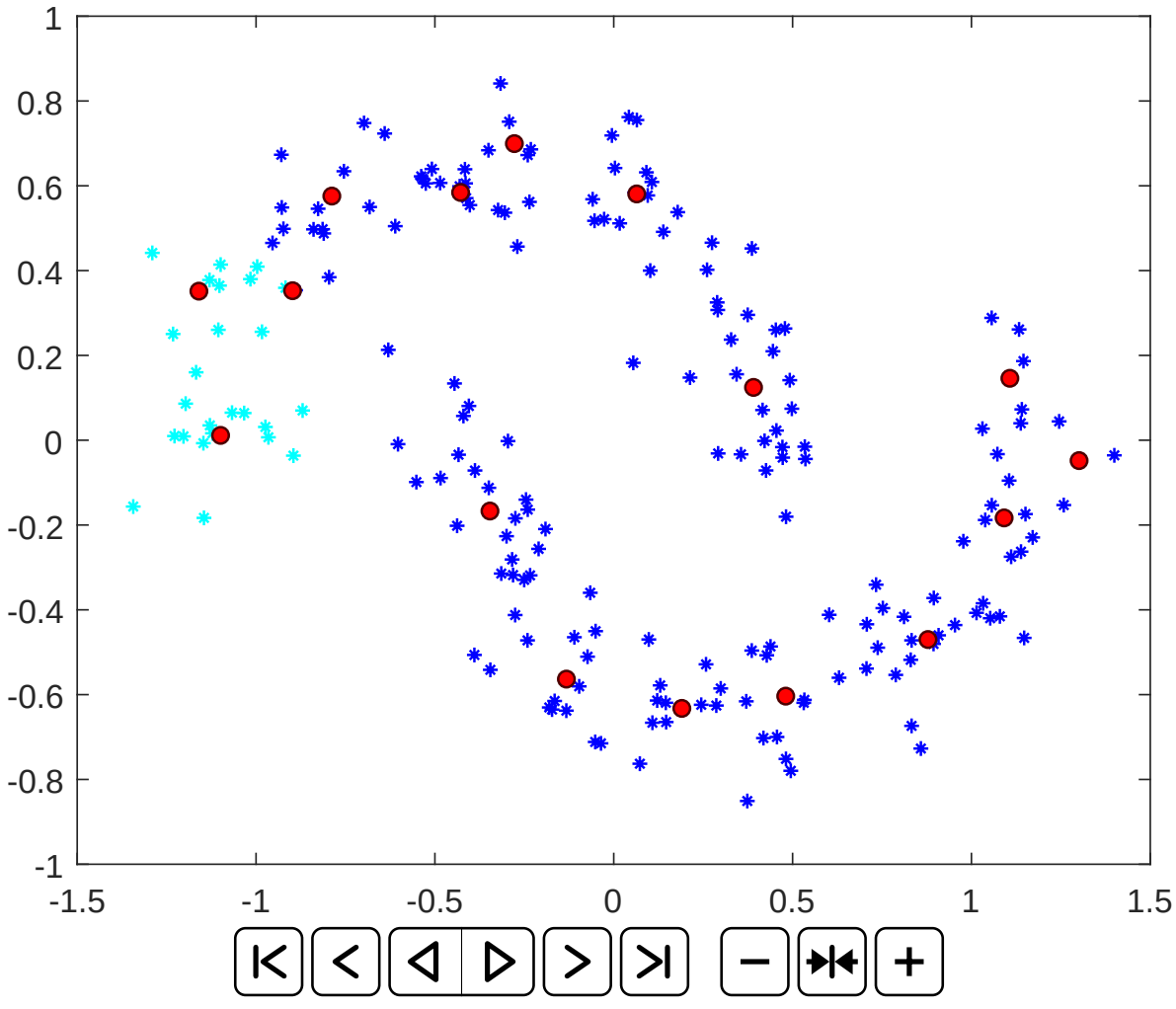
Time Complexity



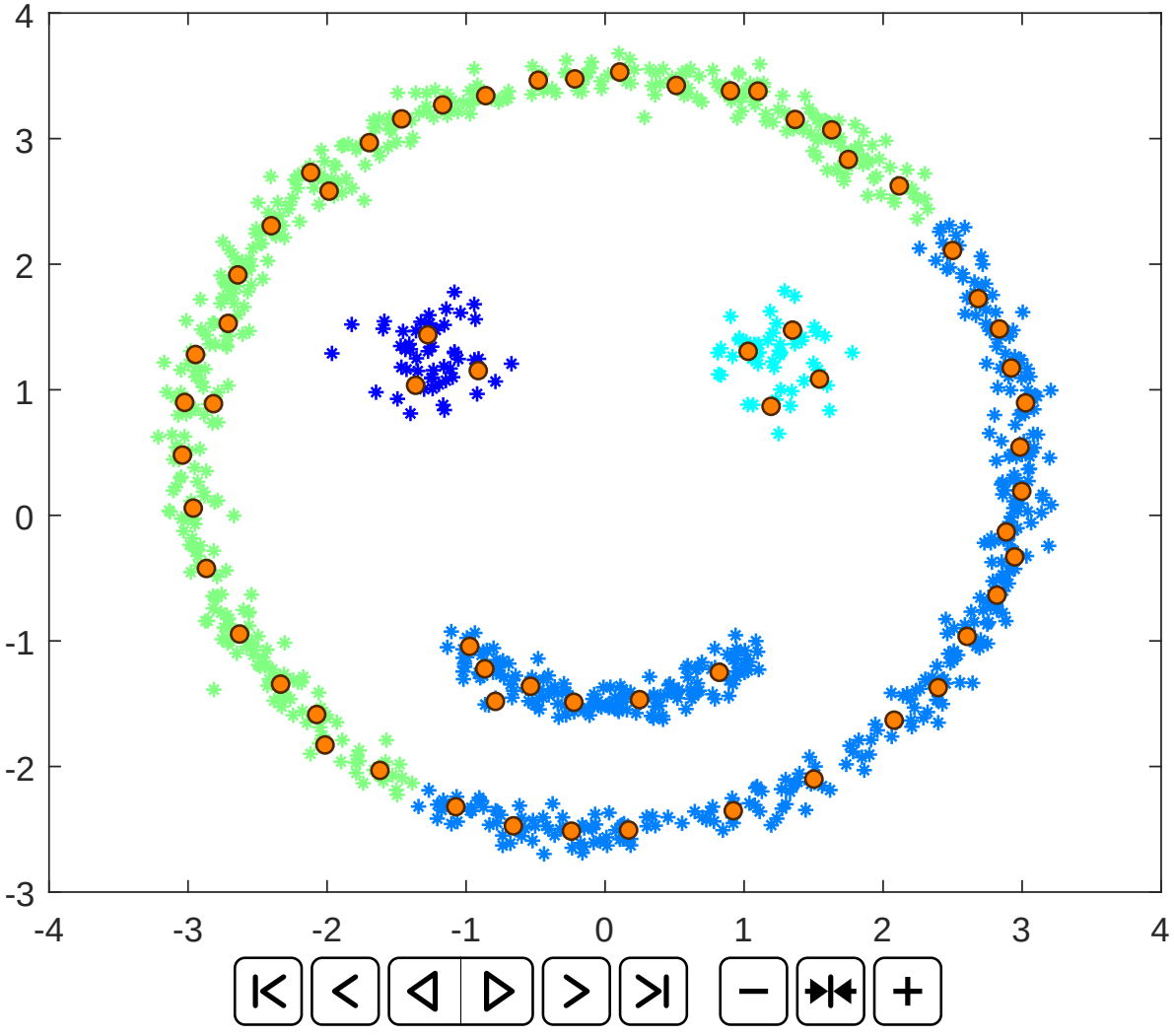
KMM scales linearly with n.

A sparse **S** can be learned to make KMM more efficiency.

Demo i



Demo ii



Experiments on Real-world Datasets

Clustering Performance Comparison on Real-world Datasets (%)

	Metric	K-means	SSC	KKmeans	RSFKC	CLR	MEAP	K-MEAP	KMM
Wine	ACC	94.94(±0.51)	66.85	96.06(±0.32)	95.50(±3.72)	93.25	94.94	48.31	**97.19 (±1.41)**
	NMI	83.23(±1.53)	40.32	85.81(±0.16)	84.88(±4.57)	77.29	83.18	5.22	**86.13 (±3.86)**
	Purity	94.94(±0.51)	66.85	96.06(±0.32)	95.50(±1.71)	93.25	94.94	48.31	**95.76 (±1.41)**
Ecoli	ACC	62.79(±6.21)	59.82	34.52(±1.16)	58.03(±9.76)	52.38	42.55	74.10	**78.85 (±4.46)**
	NMI	53.44(±3.10)	54.80	25.92(±1.85)	51.64(±16.65)	53.08	44.12	58.77	**69.48 (±4.86)**
	Purity	79.76(±3.06)	82.33	61.30(±3.00)	79.46(±11.45)	79.76	42.55	80.41	**82.37 (±3.95)**
Binalpha	ACC	64.88(±3.34)	66.82	28.26(±0.74)	59.11(±9.87)	67.40	40.99	62.94	**68.87(±7.00)**
	NMI	62.81(±1.87)	70.01	20.99(±0.38)	61.95(±13.25)	71.05	41.03	60.96	**72.94(±7.05)**
	Purity	72.33(±2.82)	76.00	35.54(±0.50)	71.19(±11.96)	**78.00**	45.41	69.84	76.59(±6.37)
Palm	ACC	63.65(±3.45)	59.78	68.70(±0.83)	71.13(±6.80)	68.65	71.55	40.20	**76.40 (±2.21)**
	NMI	87.55(±1.08)	79.98	89.06(±0.68)	89.82(±8.51)	90.27	90.60	71.23	**92.30 (±0.94)**
	Purity	71.80(±2.81)	62.90	74.60(±0.46)	76.11(±7.44)	79.45	77.80	45.70	**81.75 (±1.66)**
Abalone	ACC	14.62(±0.88)	13.96	14.79(±0.26)	19.12(±1.88)	14.96	19.70	16.51	**20.20(±1.02)**
	NMI	15.09(±0.29)	14.37	14.76(±0.14)	06.52(±3.32)	15.07	07.53	15.52	**16.03(±1.75)**
	Purity	27.36(±0.63)	**27.68**	26.43(±0.34)	19.89(±2.08)	27.67	19.70	27.31	25.20(±1.33)
Htru2	ACC	91.85(±2.10)	92.22	59.29(±1.20)	92.17(±2.55)	-	-	-	**95.49 (±2.21)**
	NMI	30.30(±1.01)	34.90	7.97(±0.56)	27.02(±2.26)	-	-	-	**40.12 (±1.55)**
	Purity	91.89(±1.32)	93.35	90.84(±0.78)	92.17(±3.58)	-	-	-	**95.49 (±1.92)**

Experiments on Real-world Datasets

Statistics of Real Benchmark Datasets

Datasets	Sample	Features	Clusters
Wine	178	13	3
Ecoli	336	7	8
BinAlpha	1854	256	10
Palm	2000	256	100
Abalone	4177	8	28
HTRU2	17898	8	2

Run Time of Multi-Means Clustering Algorithms (s)

Datasets	MEAP	K-MEAP	KMM
Wine	0.37	4.47	**0.22**
Ecoli	1.37	25.54	**0.34**
BinAlpha	164.06	1879.65	**12.52**
Palm	183.57	1904.91	**9.01**
Abalone	673.40	6804.30	**42.10**
HTRU2	-	-	227.75

Conclusion

- KMM models the clustering problem into a **bipartite graph partitioning** problem with the constrained Laplacian rank so that the problem can be formalized as an optimization problem.
- The **theoretical analysis** of the connection between our method and K-means clustering is shown.
- The time complexity of KMM is **linearly** with the data size.
- Empirical results on synthetic and real datasets show KMM **outperforms** existing methods.
- KMM can be applied to **many fields** such as vector quantization, cluster analysis, feature learning, nearest-neighbor search, data compression, etc.

Q & A (added after oral presentation)

Q & A

Q1: Does the number of prototypes m means that each cluster has m prototypes?

A1: No. The number of prototypes for k clusters adds up to a total of m. Some clusters have fewer prototypes and some clusters have more prototypes. The partition of m prototypes into k clusters is based on the partition of the bipartite graph.

Q & A

Q2: Why not clustering directly with spectral clustering?

A2: Because spectral clustering cannot give the prototypes of data points. In prototype-based clustering, apart from the partition of data points, the prototypes are also important. For example, people can use the prototypes obtained by k-means for vector quantization. KMM can give the multiple prototypes of non-convex clustering.

Q & A

Q3:It seems that KMM is a black box model and the algorithm is not fast.

A3: In fact, if you read the paper carefully you will know KMM is a white box model which has an explicit objective function. The effect of parameter settings on experimental results can also be expected. KMM is linear with respect to n. It is fast, but of course not very fast. There is still a lot of work to do, KMM is only the first step.

Q & A

Q4: Guidance for parameter?

A4:

Too large m will slow down the algorithm. Too small m may not fit the data distribution and cause inappropriate clustering results. In the paper, we set $m = \sqrt{n \times k}$, which is only an intermediate value. We can set m smaller but not too small.

The selection of the number of neighbors should also pay attention to. One case where the algorithm fails is that too small number of neighbors causes the bipartite graph to be over-segmented, and the follow-up cannot be completed. Generally, we can choose a small number of neighbors to learn the local structure of the data. If it is too large, the algorithm cannot learn the local structure of the data.

Q & A

Q5: The performance of KMM seems limited ?

A5: Yes. Actually the superiority of KMM on clustering may be limited which is only a heuristics algorithm. The innovation of KMM is that KMM can learn multiple prototypes of clusters with better performance than the previous methods. Prototype vectors can be used for vector quantization. More work to improve the clustering performance of KMM remains to be studied.

Thank you!

Welcome to visit our project!

https://github.com/CHLWR/KDD2019_K-Multiple-Means

Presenter: Cheng-Long Wang

ch.l.w.reason@gmail.com