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

**Conference Paper** · July 2019

DOI: 10.1145/3292500.3330846

Feiping Nie
University of Texas at Arlington

Cheng-Long Wang
Northwestern Polytechnical University

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

Feiping Nie
School of Computer Science
and Center for OPTIMAL
Northwestern Polytechnical
University
Xi’an, China
feipingnie@gmail.com

Cheng-Long Wang
School of Computer Science
and Center for OPTIMAL
Northwestern Polytechnical
University
Xi’an, China
ch.l.w.reason@gmail.com

Xuelong Li
School of Computer Science
and Center for OPTIMAL
Northwestern Polytechnical
University
Xi’an, China
li@nwpu.edu.cn

## ABSTRACT

In this paper, we make an extension of K-means for the clustering of multiple means. The popular K-means clustering uses only one center to model each class of data. However, the assumption on the shape of the clusters prohibits it to capture the non-convex patterns. Moreover, many categories consist of multiple subclasses which obviously cannot be represented by a single prototype. We propose a K-Multiple-Means (KMM) method to group the data points with multiple sub-cluster means into specified $k$ clusters. Unlike the methods which use the agglomerative strategies, the proposed method formalizes the multiple-means clustering problem as an optimization problem and updates the partitions of $m$ sub-cluster means and $k$ clusters by an alternating optimization strategy. Notably, the partition of the original data with multiple-means representation is modeled as a bipartite graph partitioning problem with the constrained Laplacian rank. We also show the theoretical analysis of the connection between our method and the K-means clustering. Meanwhile, KMM is linear scaled with respect to $n$. Experimental results on several synthetic and well-known real-world data sets are conducted to show the effectiveness of the proposed algorithm.

## CCS CONCEPTS

• **Theory of computation → Unsupervised learning and clustering**;

## KEYWORDS

Clustering; K-means; Multiple means; Graph Laplacian

**ACM Reference Format:**
Feiping Nie, Cheng-Long Wang, and Xuelong Li. 2019. K-Multiple-Means: A Multiple-Means Clustering Method with Specified K Clusters. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330846

*KDD ’19, August 4–8, 2019, Anchorage, AK, USA*

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330846

## 1 INTRODUCTION

Clustering is one of the most fundamental topics in data mining. Given a set of unlabeled objects, the task of clustering is to group the objects so that the objects with high similarity are grouped into the same group. An enormous variety of clustering algorithms [8, 18, 23, 25, 27, 31, 33] have been proposed over the past few decades. One of the most popular clustering algorithms is K-means [23], which aims to find a partition of the data into $k$ clusters such that the sum of the squared errors (SSE) of each point to the mean of the corresponding cluster is minimized. K-means uses an alternating minimization method to iteratively update the data assignments and the means of $k$ clusters until the assignments no longer change. In many real situations, issues such as overlapping intensities of different categories affect the performance of the algorithm. One variant of K-means is Fuzzy C-means [30], which computes the fuzzy membership of the data points to each cluster rather than a hard one. The Fuzzy C-means algorithm is more robust for ambiguity than K-means. The K-means-type (hard or fuzzy) algorithms have attracted a lot of attention in the community of data scientists and many impressive results have been reported [1, 2, 4, 6, 10, 16, 17, 29, 34].

Despite the simpleness and efficiency, the squared error criterion of the K-means-type algorithms tends to work well in hyperspherical clusters, which prohibits the algorithms to capture the non-convex patterns. Moreover, in many applications, each category consists of multiple subclasses which obviously cannot be represented by a single prototype. One option is to use the nonlinear clustering methods, such as kernel-based clustering and spectral clustering [9, 12, 24, 27, 39].The kernel clustering methods map the data into some feature space for a linear partition. Similarly, the spectral clustering methods dose a low-dimension embedding of the similarity matrix of data points [33]. Those methods aim to produce nonlinear separating hyper-surfaces between clusters whereas the design of the appropriate kernel or the construction of the data graph is not easy for every partition problem [5, 28].

Another line of research focuses on multiple prototypes (also known as class centers or class means) representation [15, 20, 22, 32, 35, 36]. Having more than one representative prototypes per cluster allows those algorithm to adjust well to the geometry of non-spherical shapes. In general, the multi-prototype clustering algorithms first split the data into many small subclasses and then iteratively merge them into a given number of clusters by some similarity measures. However, most of those algorithms use agglomerative strategies which often encounter difficulties regarding the selection of merge or split points. Wang et al. proposed a Multi-Exemplar Affinity Propagation (MEAP) algorithm to model the category of more complex structure. However, this algorithm uses a user-defined similarity matrix to calculate the messages, and the clustering results depend on the quality of the input matrix.

A notable technique in partitioning clustering literature is Constrained Laplacian Rank (CLR) [28]. The CLR method learns a graph with exactly $k$ connected components such that the optimal clustering results can be achieved simultaneously. Inspired by the previous work on the multiple prototypes representation and graph construction, we propose a K-Multiple-Means method to group the data points with multiple sub-cluster means into specified $k$ clusters. The proposed method formalizes the problem as an optimization problem and solves it by an alternating optimization strategy. Mainly, our method models the partition problem of data points with multiple-mean representation into a bipartite graph partitioning problem with the constrained Laplacian rank. In each iteration, the similarities between the data points and the sub-cluster means are updated following with the partition of the bipartite graph, and then the means of sub-clusters are relocated. The main contributions of this paper can be summarized as follows:

- We propose a multiple-means extension of K-means, i.e. K-Multiple-Means to solve the clustering problem of data points with multiple-means. Our approach 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.
- An efficient alternating optimization strategy with complexity analysis to solve the K-Multiple-Means is provided.
- We show the theoretical analysis of the connection between our method and K-means clustering.
- Experimental results show that KMM achieve better performance than the existing multiple-means methods.

The remainder of this paper is organized as follows: Section 2 briefly introduces the related works. In Section 3, we describe the proposed K-Multiple-Means methods and the theory analysis on the connection between our approach and the K-means clustering. The optimization of it is developed in Section 4. Section 5 reports the experimental results, respectively. Section 6 concludes this paper.

**Notations:** Throughout the paper, all the matrices are denoted by uppercase letters. For matrix $A$, the $i$-th row (with transpose) and the $(i, j)$-th element of $A$ are denoted by $a_i$ and $a_{ij}$, respectively; the transpose of A is denoted by $A^T$ and $Tr(A)$ denotes the trace of $A$. The L2-norm of vector $v$ is denoted by $\|v\|_2$, the Frobenius norm of matrix $A$ is denoted by $\|A\|_F$. An identity matrix is denoted by $I$, and $\mathbf{1}$ denotes a column vector with all the elements are one. For vector $v$ and matrix $A$, $v \geq 0$ and $A \geq 0$ mean all the elements of $v$ and $A$ are equal to or larger than zero.

# 2 RELATED WORKS

## 2.1 K-means-type Algorithms

Denote the data matrix by $X = [x_1, \ldots, x_n]^T \in R^{n \times d}$, the K-means-type algorithms [4, 23] try to find an optimal partition of dataset into $k$ clusters that minimize the following objective function

$$\begin{aligned} & f(U, V) = \sum_{i=1}^{n} \sum_{l=1}^{k} u_{il}^{r} \|x_i - v_l\|_2^2 \\ & s.t. \;\; U \geq 0, U\mathbf{1} = \mathbf{1}, \end{aligned} \tag{1}$$

where $U \in R^{n \times k}$, $V \in R^{k \times d}$. The membership degree of $u_{il}$ denotes the grade of membership of the $i$-th point in the $l$-th cluster; $v_l$ denotes the mean (center) of the $l$-th cluster, and $r \in [1, +\infty)$ is the fuzzy index. An alternating optimization strategy is used to solve the problem with iteratively updating the partition and the means of clusters.

For $r > 1$, if $x_i \neq v_l$ for all $i$ and $l$, $U, V$ are updated by

$$\begin{aligned} \hat{u}_{il} &= \frac{1}{\sum_{j=1}^{k} \left( \frac{\|x_i - v_l\|_2^2}{\|x_i - v_j\|_2^2} \right)^{\frac{2}{r-1}}}, 1 \leq i \leq n, 1 \leq l \leq k \\ \hat{v}_l &= \frac{\sum_{i=1}^{n} u_{il}^{r} x_i}{\sum_{i=1}^{n} u_{il}^{r}}, 1 \leq l \leq k \end{aligned} \tag{2}$$

respectively.

The iterative procedure will stop when $\max_{il}\{|u_{il}^{t+1} - u_{il}^{t}|\} \leq \epsilon$, where $t$ denotes the iteration step and $\epsilon$ is a termination criterion between 0 and 1. The fuzzy K-means algorithm produces a fuzzy partition matrix $U$. The point $x_i$ is assigned to the $l$-th cluster if $u_{il} = max_{1 \leq j \leq k}\{u_{ij}\}$.

For $r = 1$, it can be shown that the fuzzy K-means becomes the classical K-means.

The iterative procedure will stop when the means do not change.

## 2.2 Multi-Prototypes Clustering Algorithms

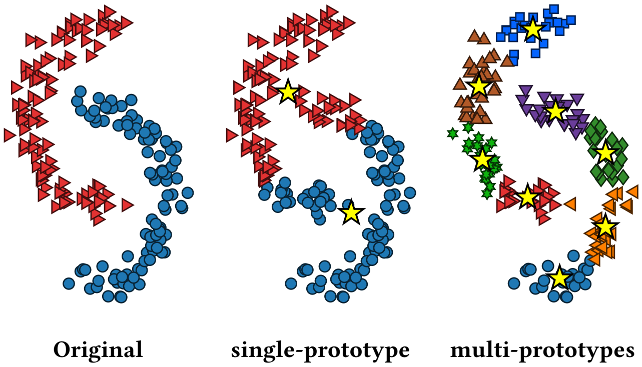


**Figure 1: The illustration of the difference between single-prototype representation and multi-prototypes representation. Each point is assigned to its nearest prototype (yellow "★").**

The squared error criterion of the K-means-type algorithms tends to work well in hyper-spherical clusters, which prohibits the algorithms to capture the non-convex patterns. One simple but effective alternative is the multi-prototypes representation, which models a cluster via multiple prototypes. Figure 1 shows an illustrative example of the difference between single-prototype

**Table 1: Comparison of the Multi-Prototypes Clustering Algorithms**

| | Split Stage | Merge Stage |
|---|---|---|
| Tao [32] | 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. [20] | Squared-error clustering | The prototypes who coexist in a high-density region are grouped into one cluster. |
| Luo et al. [22] | Minimum spanning tree | 1) The prototypes whose distance is smaller than the user-specified threshold are roughly merged.<br>2) Further merge step will be conducted based on the data distribution between two clusters prototypes. |
| Ben et al. [3] | 1) Fuzzy C-means<br>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. [19] | 1) Fast Global Fuzzy K-means<br>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. |

representation and multi-prototypes representation. Since multi-prototypes are used to represent clusters, the non-spherical clusters can be correctly detected.

The work on clustering based on multi-prototypes representation is relative less. Most multi-prototypes clustering algorithms [3, 19, 20, 22, 32] consist of a splitting stage and a merging stage based on agglomerative strategies. A comparison of 5 multi-prototypes clustering algorithms is presented in Table 1.

Whether it is a fuzzy clustering algorithm or a hard clustering algorithm, the key to obtaining a specified number of clusters is to judge the connection relationships between multiple centers. And it is easy to know that the degree of overlap between two sub-clusters belonging to the same cluster is likely to be significant. It is not enough clustering only based on the distances between multi-prototypes, but also considering the data distribution of each sub-cluster. Most of those algorithms use agglomerative strategies often encounter difficulties regarding the selection of merge or split points. The previous analysis shows that the design of ideal partitioning strategies for data points and multi-prototypes should be based on the following assumptions:

- 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 is based on both the distribution of multi-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.

In the next section, we will propose a more reasonable clustering approach based on multi-prototypes representation.

## 3 K-MULTIPLE-MEANS

Denote $A = [a_1, a_2, \ldots, a_m]^T \in R^{m \times d}$ as the prototype matrix. For the $i$-th data point $x_i$, the $j$-th prototype $a_j$ can be connected to $x_i$ as a neighboring prototype with probability $s_{ij}$. Usually, the smaller the distance $\left\|x_i - a_j\right\|_2^2$ between $x_i$ and $a_j$ is, the greater the corresponding connection probability $s_{ij}$ is. So the assignment problem of $n$ data points' neighboring prototypes based on the weighted squared error criterion can be written as

$$\begin{aligned} &\min_S \sum_{i=1}^{n} \sum_{j=1}^{m} s_{ij} \left\|x_i - a_j\right\|_2^2 + \gamma \left\|S\right\|_F^2 \\ &s.t.\ S \geq 0, S\mathbf{1} = \mathbf{1}, \end{aligned} \tag{3}$$

where $s_{ij}$ is the $(i, j)$-th element of matrix $S$. The second term in problem (3) is a regularization term. The regularization parameter $\gamma$ is used to control the sparsity of the connection of data points to multi-prototypes. When $\gamma = 0$, the problem (3) has a trivial solution, only the nearest prototype can be connected to $x_i$ with probability $s_{ij} = 1$ and all the other prototypes cannot be connected to $x_i$, which means that the partition is a hard partition. When $\gamma$ is large enough, all $m$ prototypes can be connected to $x_i$ with the same probability $\frac{1}{m}$.

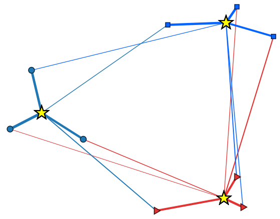


**Figure 2: An example of the assignment of neighboring prototypes. Each data point is connected with two nearest prototypes. The width of the line indicates the connection probability of each data point to the corresponding prototype.**

For each data point $x_i$, the assignment of neighbors is independent. So we can update the assignment of neighboring prototypes individually for each $x_i$. Denote $d_{ij}^x = \left\|x_i - a_j\right\|_2^2$ and denote $d_i^x$ as a vector with the $j$-th element as $d_{ij}^{x}$ (same for $s_i$), the assignment of neighboring prototypes for $x_i$ can be written in vector form as

$$\min_{s_i} \left\| s_i - \frac{d_i^x}{2\gamma} \right\|_2^2 \quad s.t.\ s_i \geq 0, s_i^T \mathbf{1} = 1. \tag{4}$$

This problem can be solved with a closed form solution [27]. Following [27], denote $d_{i1}^{th}, d_{i2}^{th}, \ldots, d_{im}^{th}$ as the distance of $m$ prototypes to $i$-th data point ordered from small to large. $\gamma$ can be set to $\gamma = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\tilde{k}}{2}d_{i,\tilde{k}+1}^{th} - \frac{1}{2}\sum_{j=1}^{\tilde{k}} d_{ij}^{th}\right)$, which is controlled by the number of neighbor prototypes $\tilde{k}$. Thus the probability that the $\tilde{j}$-th nearest prototype closest to $i$-th data point is connected to it is $s_{i\tilde{j}} = \frac{d_{i,\tilde{k}+1}^{th} - d_{i\tilde{j}}^{th}}{\tilde{k}d_{i,\tilde{k}+1}^{th} - \sum_{j=1}^{\tilde{k}} d_{i\tilde{j}}^{th}}$, where $\tilde{j} \leq \tilde{k}$. When $\tilde{j} > \tilde{k}$, $s_{i\tilde{j}} = 0$.

When $S$ is updated, each prototype can be relocated to the mean of all data points assigned to it respectively. For $j$-th prototype, $a_j$ can be updated by

$$a_j = \frac{\sum_{i=1}^{n} s_{ij} x_i}{\sum_{i=1}^{n} s_{ij}}. \tag{5}$$

This process can be iteratively performed by Eq.(6) to obtain an optimal multiple-mean neighbor assignment until the assignment is not updated.

$$\min_{A,S} \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij} \left\| x_i - a_j \right\|_2^2 + \gamma \left\| S \right\|_F^2 \quad s.t.\ S \geq 0, S\mathbf{1} = \mathbf{1}, A \in R^{m \times d}. \tag{6}$$

An example of the assignment of neighboring prototypes is presented in Figure 2. In most cases, the neighbor assignment with Eq.(6) connects $n$ data points and $m$ prototypes as just one connected component. In order to achieve the ideal neighbors assignment, a new appropriate constraint should be imposed on the objective function. Denote the constraint that $S$ has exactly $k$ connected components by $S \in \Omega$, the problem (6) becomes

$$\min_{A,S} \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij} \left\| x_i - a_j \right\|_2^2 + \gamma \left\| S \right\|_F^2 \quad s.t.\ S \geq 0, S\mathbf{1} = \mathbf{1}, S \in \Omega, A \in R^{m \times d}. \tag{7}$$

The problem (7) is not easy to solve. Because constraint $\Omega$ depends on $S$ and $\Omega$ is hard to tackle. In the next section, we will propose a efficient algorithm to solve the challenging problem.

## 4 OPTIMIZATION STRATEGY

Denote matrix $P = \begin{bmatrix} & S \\ S^T & \end{bmatrix}$ and the normalized Laplacian matrix $\tilde{L}_S$ associated with $S$ as $\tilde{L}_S = I - D^{-\frac{1}{2}}PD^{-\frac{1}{2}}$, where $D \in R^{(n+m)\times(n+m)}$ is defined as a diagonal matrix where the $i$-th diagonal element is $d_{ii} = \sum_j p_{ij}$. Chung [7] pointed out that if the similarity matrix $S$ is nonnegative, the normalized Laplacian matrix $\tilde{L}_S$ has an important property as follows:

Theorem 4.1. *The multiplicity $k$ of the eigenvalue 0 of the normalized Laplacian matrix $\tilde{L}_S$ is equal to the number of connected components in the bipartite graph associated with $S$.*

The Theorem 4.1 shows that if $rank(\tilde{L}_S) = (n+m) - k$, the bipartite graph associated with $S$ has $k$ connected components, i.e., the $n$ data points and $m$ prototypes are grouped into $k$ clusters. Motivated by Theorem 4.1, we add an additional constraint $rank(\tilde{L}_S) = (n+m) - k$ into the problem (6) to achieve the ideal prototypes assignment with specified $k$ clusters. Thus, our clustering model is to solve

$$\min_{A,S} \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij} \left\| x_i - a_j \right\|_2^2 + \gamma \left\| S \right\|_F^2 \quad s.t.\ S \geq 0, S\mathbf{1} = \mathbf{1}, A \in R^{m \times d}, rank(\tilde{L}_S) = (n+m) - k. \tag{8}$$

Since the rank constraint $rank(\tilde{L}_S) = (n+m) - k$ is hard to tackle, it's necessary to relax the constraint for solving. Denote $\sigma_i(\tilde{L}_S)$ as the $i$-th smallest eigenvalue of $\tilde{L}_S$. Note that $\sigma_i(\tilde{L}_S) \geq 0$ because $L_S$ is positive semi-definite. The optimal solution of $S$ with rank constraint can be achieved by solving the following problem

$$\min_{A,S} \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij} \left\| x_i - a_j \right\|_2^2 + \gamma \left\| S \right\|_F^2 + \lambda \sum_{i=1}^{k} \sigma_i(\tilde{L}_S). \quad s.t.\ S \geq 0, S\mathbf{1} = \mathbf{1}, A \in R^{m \times d}. \tag{9}$$

When $\lambda$ is large enough, the optimal solution of $S$ to the problem (9) will make $\sum_{i=1}^{k} \sigma_i(\tilde{L}_S)$ to be zero, and thus the rank constraint could be satisfied.

According to the Ky Fan's Theorem [11], we have

$$\sum_{i=1}^{k} \sigma_i(\tilde{L}_S) = \min_{F \in R^{(n+m)\times k}, F^T F = I} Tr(F^T \tilde{L}_S F). \tag{10}$$

Thus, the problem (9) can be further written as :

$$\min_{A,S,F} \sum_{i}\sum_{j} s_{ij} \left\| x_i - a_j \right\|_2^2 + \gamma \left\| S \right\|_F^2 + \lambda Tr(F^T \tilde{L}_S F) \quad s.t.\ S \geq 0, S\mathbf{1} = \mathbf{1}, A \in R^{m \times d}, F \in R^{(n+m)\times k}, F^T F = I, \tag{11}$$

where $\tilde{L}_S = I - D^{-\frac{1}{2}} \begin{bmatrix} & S \\ S^T & \end{bmatrix} D^{-\frac{1}{2}}$.

The problem (11) can be solved by an alternating optimization method which updates $S$, $F$ and $A$ iteratively.

### 4.1 Fix $A$ and Update $S, F$

When $A$ is fixed, the algorithm learns an $S$ which has exactly $k$ connected components, see Figure 3. The problem (11) becomes

$$\min_{S,F} \sum_{i}\sum_{j} s_{ij} \left\| x_i - a_j \right\|_2^2 + \gamma \left\| S \right\|_F^2 + \lambda Tr(F^T \tilde{L}_S F) \quad s.t.\ S \geq 0, S\mathbf{1} = \mathbf{1}, F \in R^{(n+m)\times k}, F^T F = I, \tag{12}$$

Similarly, the problem (12) can also be solved by an alternating optimization approach.

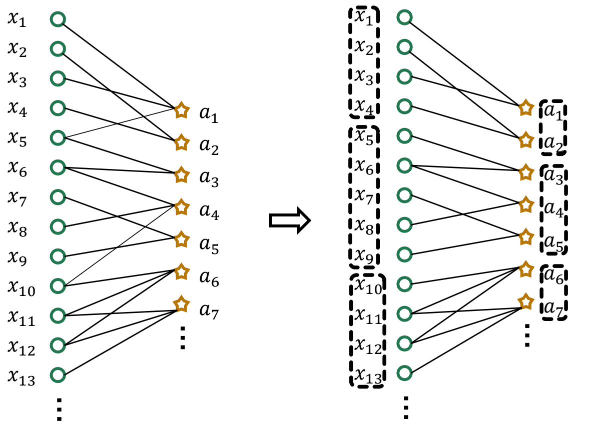


**Figure 3: Illustration of the optimal bipartite graph with constraint. Left: the initial bipartite graph whose nodes are connected as only one connected component. Right: The structured bipartite graph with a specified number of connected components. The points in the same dashed box belong to the same cluster.**

When $S$ is fixed, since $\tilde{L}_S = I - D^{-\frac{1}{2}} \begin{bmatrix} & S \\ S^T & \end{bmatrix} D^{-\frac{1}{2}}$, the problem (12) becomes

$$\max_{F\in\mathbb{R}^{(n+m)\times k}, F^TF=I} Tr(F^T D^{-\frac{1}{2}} \begin{bmatrix} & S \\ S^T & \end{bmatrix} D^{-\frac{1}{2}} F). \tag{13}$$

We rewrite $F$ and $D$ as the block matrices

$$F = \begin{bmatrix} U \\ V \end{bmatrix}, \; D = \begin{bmatrix} D_U & \\ & D_V \end{bmatrix},$$

where $U \in R^{n\times k}, V \in R^{m\times k}, D_U \in R^{n\times n}, D_V \in R^{m\times m}$. The problem (13) can be further rewritten as

$$\max_{U^TU+V^TV=I} Tr(U^T D_U^{-\frac{1}{2}} S D_V^{-\frac{1}{2}} V). \tag{14}$$

The problem (14) can be solved according to the Lemma 4.2 [26].

Lemma 4.2. *Suppose $A \in \mathbb{R}^{n\times m}$, $X \in \mathbb{R}^{n\times k}$, $Y \in \mathbb{R}^{m\times k}$. The optimal solutions to the problem*

$$\max_{X^TX+Y^TY=I} Tr(X^TAY)$$

*are $X = \frac{\sqrt{2}}{2}U_1$, $Y = \frac{\sqrt{2}}{2}V_1$ ,where $U_1, V_1$ are the leading $k$ left and right singular vectors of $A$, respectively.*

When $F$ is fixed, the problem (12) becomes

$$\min_S \sum_i^n \sum_j^m s_{ij}\|x_i - a_j\|_2^2 + \gamma\|S\|_F^2 + \lambda Tr(F^T\tilde{L}_SF) \\ s.t.\ S \geq 0, S\mathbf{1} = \mathbf{1}. \tag{15}$$

Recall that $\tilde{L}_S = I - D^{-\frac{1}{2}} \begin{bmatrix} & S \\ S^T & \end{bmatrix} D^{-\frac{1}{2}}$ and $D_S$ also depends on $S$, it looks also difficult to solve.

Fortunately, we have the following relationship:

$$Tr(F^T\tilde{L}_SF) = \frac{1}{2}\sum_{i=1}^{(n+m)}\sum_{j=1}^{(n+m)} p_{ij}\left\|\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}}\right\|_2^2. \tag{16}$$

According to the structure of $P$, Eq.(16) can be rewritten as

$$Tr(F^T\tilde{L}_SF) = \sum_{i=1}^{n}\sum_{j=1}^{m} s_{ij}\left\|\frac{f_i}{\sqrt{d_i}} - \frac{f_{(n+j)}}{\sqrt{d_{(n+j)}}}\right\|_2^2. \tag{17}$$

Denote $v_{ij} = \left\|\frac{f_i}{\sqrt{d_i}} - \frac{f_{(n+j)}}{\sqrt{d_{(n+j)}}}\right\|_2^2$, the problem (15) can be rewritten as

$$\min_S \sum_i^n \sum_j^m (s_{ij}\left\|x_i - a_j\right\|_2^2 + \gamma s_{ij}^2 + \lambda s_{ij}v_{ij}) \\ s.t.\ S \geq 0, S\mathbf{1} = \mathbf{1}. \tag{18}$$

Note that the problem (18) is independent between different $i$, so we can solve the following problem individually for each $i$:

$$\min_{s_i} \sum_j^m (s_{ij}\left\|x_i - a_j\right\|_2^2 + \gamma s_{ij}^2 + \lambda s_{ij}v_{ij}) \\ s.t.\ s_i \geq 0, s_i^T\mathbf{1} = 1. \tag{19}$$

Denote $\tilde{d}_i \in \mathbb{R}^{m\times 1}$ as a vector with the $j$-th element as $\tilde{d}_{ij} = d_{ij}^x + \lambda v_{ij}$, where $d_{ij}^x = \left\|x_i - a_j\right\|_2^2$, then the problem (19) can be written in vector form as

$$\min_{s_{ij}\geq 0, s_i^T 1=1} \left\|s_i + \frac{1}{2\gamma}\tilde{d}_i\right\|_2^2, \tag{20}$$

which can be solved with a closed form solution [27]. The iterative sub-procedure will stop when the rank constraint of $\tilde{L}_S$ is satisfied, i.e., $\sum_{i=1}^k \sigma_i\left(\tilde{L}_S\right) = 0$ and $\sum_{i=1}^{k+1} \sigma_i\left(\tilde{L}_S\right) > 0$.

## 4.2 Fix $S$, $F$ and Update $A$

When $S$ and $F$ is fixed, each sub-cluster representative is relocated to the weighted mean of all data points assigned to it. The $j$-th prototype can be updated by

$$a_j = \frac{\sum_{i=1}^n s_{ij}x_i}{\sum_{i=1}^n s_{ij}}. \tag{21}$$

The algorithm converges when the assignments no longer change. To make these update rules clear, we summarize the algorithm to solve Eq.(11) in Algorithm 1.

# 5 THEORETICAL ANALYSIS

In this section, we show the theoretical analysis of the connection between our method and the K-means clustering and provide the computational complexity analysis.

## 5.1 Connection to K-means Clustering

When $\gamma \to \infty$, we have the following theorem:

Theorem 5.1. *When $\gamma \longrightarrow \infty$ ,*

$$\min_{S\geq 0, S1=1, A, S\in\Omega} \sum_i^n \sum_j^m s_{ij}\left\|x_i - a_j\right\|_2^2 + \gamma\left\|S\right\|_F^2 \tag{22}$$

```
Algorithm 1: K-Multiple-Means
Input  : Data matrix $X \in R^{n\times d}$, cluster number $k$,
         subcluster number $m$, parameter $\gamma$, a large enough $\lambda$
Output: $k$ clusters
Initialize multiple-means $A$ (e.g. by picking $m$ samples at random).
repeat
    1.Calculate $S$ by the optimal solution to the problem (4).
    while not converge do
        1. Update $F = \begin{bmatrix} U \\ V \end{bmatrix}$, where $U$ and $V$ are $\frac{\sqrt{2}}{2}$ of the
           leading $k$ left and right singular vectors of
           $\tilde{S} = D_U^{-\frac{1}{2}} S D_V^{-\frac{1}{2}}$ respectively and $D = \begin{bmatrix} D_U & \\ & D_V \end{bmatrix}$;
        2. For each $i$, update the $i$-th row of $S$ by solving the
           problem (20), where $\tilde{d}_i \in \mathbb{R}^{m\times 1}$ as a vector with
           the $j$-th element as
           $\tilde{d}_{ij} = \|x_i - a_j\|_2^2 + \lambda \left\| \frac{f_i}{\sqrt{d_i}} - \frac{f_{(n+j)}}{\sqrt{d_{(n+j)}}} \right\|_2^2 .$
    end
    2.For each $j$, update the $j$-th row of $A$ with Eq.(21);
until converge;
Directly achieve the $k$ clusters based on the connectivity of
the bipartite graph.
```

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

Proof. When $S$ satisfies the constraint (which means that $n$ data points and $m$ prototypes are grouped into $k$ clusters), $\lambda Tr(F^T \tilde{L}_S F)$ will become to 0. Denote the $i$-th component of $S$ by $S_i \in R^{n_i \times m_i}$, where $n_i$ is the number of data points in the component and the $m_i$ is the number of prototypes in the component. When $\gamma \longrightarrow \infty$, solving problem (22) is to solve the following problem for each $i$:

$$\min_{S_i \geq 0, S_i \mathbf{1} = 1, A_i} \|S_i\|_F^2 . \tag{23}$$

The optimal solution to the problem (23) is that all the element of $S_i$ are equal to $\frac{1}{m_i}$. Therefore, the optimal solution $S$ to the problem (22) should be the following form when $\gamma \longrightarrow \infty$:

$$s_{ij} = \begin{cases} \frac{1}{m_k} & x_i, a_j \text{ are in the same component } k \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

When the number of data points and prototypes in each component is fixed, denote the partition by $\mathcal{V}$. The problem (22) becomes

$$\min_{A,Z \in R^{n\times m}, Z \in \mathcal{V}} \sum_i^n \frac{1}{\sum_h z_{ih}} \sum_j^m z_{ij} \|x_i - a_j\|_2^2 , \tag{25}$$

where $z_{ij} = \begin{cases} 1, & x_i \text{ is connected with } a_j \\ 0, & x_i \text{ is not connected with } a_j \end{cases}$.

The optimal solution of $a_j$ is $a_j^* = \frac{1}{\sum_l z_{lj}} \sum_l x_l z_{lj}$. Then the problem (25) can be written as

$$\min_{A,Z \in R^{n\times m}, Z \in \mathcal{V}} \sum_i^n \frac{1}{\sum_h z_{ih}} \sum_j^m z_{ij} \left\| x_i - \frac{1}{\sum_l z_{lj}} \sum_l x_l z_{lj} \right\|_2^2 . \tag{26}$$

It's notable that for the prototypes belonging to the same cluster, the mean values are equal. Denote the indicator matrix by $Y \in R^{n\times k}$, if $x_i$ belongs to the $l$-th cluster, $y_{il} = 1$; otherwise, $y_{il} = 0$. Note also that if $x_i$ and $a_j$ belongs to the same cluster, $z_{ij} = 1$; otherwise, $z_{ij} = 0$. Thus the problem (26) can be written as

$$\min_{Y \in R^{n\times k}, Y \in \mathcal{V}} \sum_i^n \sum_j^k y_{ij} \left\| x_i - \frac{1}{\sum_i y_{ij}} \sum_i x_i y_{ij} \right\|_2^2 . \tag{27}$$

This problem (27) is closely related to

$$\min_{Y \in R^{n\times k}, Y \in Ind} \sum_i^n \sum_j^k y_{ij} \left\| x_i - \frac{1}{\sum_i y_{ij}} \sum_i x_i y_{ij} \right\|_2^2 . \tag{28}$$

Obviously, the problem (28) is equivalent to

$$\min_{Y \in R^{n\times k}, Y \in Ind, U \in R^{k\times d}} \sum_{i=1}^n \sum_{j=1}^k \|x_i - u_j\|_2^2 \, y_{ij}, \tag{29}$$

The problem (29) is the problem of $K$-means, which completes the proof. □

## 5.2 Computational Complexity

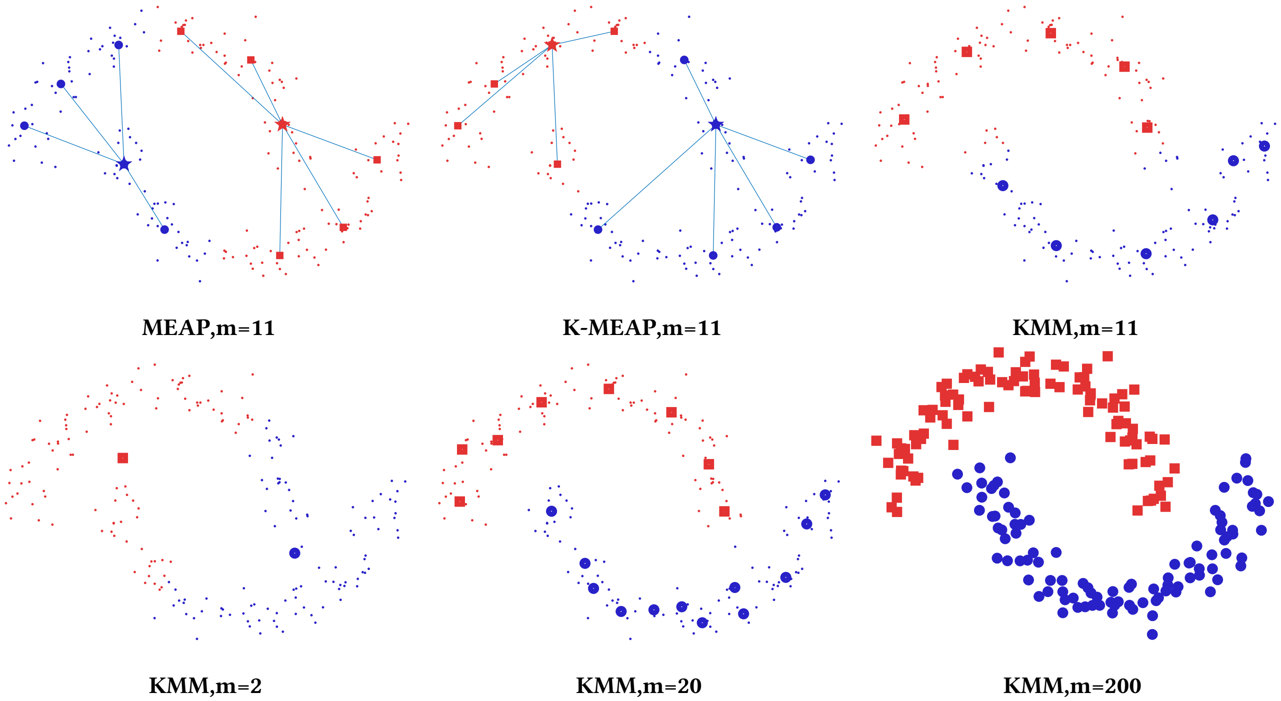
In this subsection, we consider the computational complexity of Algorithm 1. The cost of constructing the initial k-nn graph is $O(nmd + nmlog(m))$. To update $S$ with Eq.(12), we need to run a sub-alternative procedure. To update $F$ with Eq.(14), we need $O(m^3 + m^2 n)$ to perform SVD decomposition on $\tilde{S} = D_U^{-\frac{1}{2}} S D_V^{-\frac{1}{2}}$. And then we need $O(nmk + nmlog(m))$ to get the closed-form solution of $S$ with Eq.(20). Since $log(m)$and $m^3$ are usually small, the computational complexity of the sub-iteration procedure is $(O(nmd + nmc + m^2 n)t_1)$, where $t_1$ is the number of iterations of the sub-alternating system. To update $A$ with Eq.(21), we need $O(nmd)$.

Overall, we need $O(n((md + mc + m^2)t_1 + md)t)$, where $t$ is the number of iterations. It's notable that the KMM is linear scaled with respect to $n$.

In order to make the algorithm more efficiency, we can learn a sparse $S$, which only needs to update the $\tilde{k}$ nearest similarities for each data point in $S$ and set other similarities in $S$ to zero. Thus the time complexity of updating $S$ can be reduced significantly. Meanwhile, since $S$ is sparse, the step of updating $F$ only needs to compute the top $\tilde{k}$ eigenvectors on a very sparse matrix. The algorithm will be faster.

# 6 EXPERIMENTS

In this section, we investigate the performance of K-Multiple-Means on both synthetic data and real benchmark datasets. For simplicity, we denote our algorithm as KMM in the following context.


MEAP,m=11

K-MEAP,m=11

KMM,m=11

KMM,m=2

KMM,m=20

KMM,m=200

**Figure 4: The Difference of between MEAP, K-MEAP and KMM on Non-spherical Data**

## 6.1 Experiments on Synthetic Data

In this subsection, two methods based on multi-exemplar representation are performed for comparison, one of which is a multi-exemplar affinity propagation (MEAP [35]) algorithm and the other is K-MEAP [36] which is a variant of MEAP and can achieve specified $k$ clusters.

The experiment on two-moon dataset is conducted to illustrate the difference in clustering results between MEAP, K-MEAP and KMM on non-spherical clusters respectively. There are two clusters of data distributed in the moon shape. Each cluster has a volume of 100 samples and the noise percentage is set to be 0.1. The similarity matrix for MEAP and K-MEAP is constructed the same as in [35] and the parameters are set as follows: $T_{unchange} = 100, T_{max} = 1000, u = 0.9$ as in [35]. Both MEAP and K-MEAP automatically determine the number of exemplars in each cluster. For KMM, $m$ needs to be specified manually.

Fig. 4 shows the comparison results. Different clusters are distinguished by different colors. Each cluster obtained by MEAP/K-MEAP contains one super-exemplar (marked with stars) and an automatically determined number of exemplars (marked with squares and circles) assigned to that super-exemplar. Due to the loss of information of the original data distribution in the merge phase, MEAP and K-MEAP fail to learn the optimal data structure. The clusters obtained by KMM are represented by multi-prototypes (marked with squares and circles). It is easy to see that when $m = c$, the result of KMM is a solution of spherical K-means clustering problem. When $m$ becomes large, non-spherical clusters can be effectively represented by multi-prototypes.

**Table 2: 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 |

## 6.2 Experiments on Real Benchmark Datasets

We conduct experiments on six real-world datasets: Abalone, Ecoli, HTRU2, Palm, BinAlpha, Wine. A detailed summarization of these datasets is in Table 2.

*6.2.1 Methods and Settings.* The compared approaches and their parameter settings are summarized as follows:

1) **K-means**: The K-means++ algorithm [1] is used for cluster center initialization of Lloyd's algorithm [21].

2) **Spectral Clustering**: The widely used Selftuning spectral clustering (SSC) [38] is performed. We choose a Matlab version[1] of it to constructed the similarity matrix and set the number of neighbors to 5 for the similarity matrix construction.

3) **Kernel-based Clustering**: The Mercer Kernel K-means ( KKmeans) [13] method is compared here. We choose a Matlab

[1] http://www.cs.ucsb.edu/~wychen/sc

**Table 3: 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)** |

version[2] of it implemented by Gonen et al. [14]. The similarity matrix was constructed in the same way as NCut.

4) **Fuzzy Clustering**: The Robust and Sparse Fuzzy K-Means Clustering (RSFKC) [37] algorithm is compared, which extends the standard Fuzzy K-means algorithm by incorporating a robust function and have suitable sparseness. The parameters were tuned by a grid search method as in [37].

5) **Exemplar-based Clustering**: The MEAP and K-MEAP clustering methods are compared here. The construction of the similarity matrix and the setting of parameter were the same as described previously.

6) **Graph-based Clustering**: The Constrained Laplacian Rank Frobenius norm clustering method (CLR) [28] is compared, in which a Constrained Laplacian Rank method is used to learn a graph with exactly $k$ connected components. The number of neighbors is set to be 5 for the similarity matrix, which was constructed recommended in [28].

For MEAP and K-MEAP, since the algorithms may fall into a fail mode [35], we use the grid search method to select the value of $u$ from 0.1 to 0.9 in the step of 0.05 and report the best result.

For KMM, as discussed in Section 3.1, the regularization parameter $\gamma$ can be set by tuning the number of nearest neighbor $\tilde{k}$. We determined the value of $\lambda$ in a heuristic way to accelerate the procedure: first set $\lambda = \gamma$, then in each iteration, if the number of zero eigenvalues in $\tilde{L}_S$ was larger than $k$, we divided $\lambda$ by two; if smaller we multiplied $\lambda$ by two; otherwise we stopped the iteration.

In the experiments, we used $\tilde{k} = 5$. The number of sub-clusters $m$ was set to the greatest integer less than or equal to $\sqrt{n \times k}$, which is the median of the parameter adjustment range. The number of clusters was set to be the ground truth. The standard clustering Accuracy (ACC), Normalized Mutual Information (NMI) and Purity metrics were used to measure the clustering performance.

**Table 4: 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 |

6.2.2 *Results.* Note that the results of the K-means-type clustering algorithm vary on different initialization. To reduce the influence of statistical variation, we repeat each method 100 times with random initialization and report the average clustering accuracy and standard deviation. The experimental results are reported in Table 3. The results of MEAP, Kmeap, and CLR on Htru2 is missing because the size of Htru2 is too large to be learned by the graph-based methods. As can be seen from the two tables, KMM is consistently the best algorithm using three evaluation metrics.

## 6.3 Computation Time

Our newly proposed KMM is not only effective but also efficient. We record the run time of three multiple-means clustering algorithms (MEAP, K-MEAP and KMM) when conducted performance comparison experiments and report the mean of results in Table 4. All the codes in the experiments are implemented in MATLAB

[2] https://github.com/mehmetgonen/lmkkmeans

R2018a, and run on a Windows 10 machine with 3.30 GHz i5-4590 CPU, 16 GB main memory. Obviously, KMM is faster than the other two multiple-means clustering methods.

## 7 CONCLUSION

In this paper, we propose the K-Multiple-Means method to group the data points with multiple sub-cluster means into the specified $k$ clusters. The proposed method formalizes the multiple-means clustering problem as an optimization problem and updates the partitions of $m$ sub-cluster means and $k$ clusters by an alternating optimization strategy. In each iteration, the data points with multiple-means are grouped based on the partition of a bipartite graph associated with the similarity matrix. The theoretical analysis of the connection between our method and K-means clustering is shown. Experimental extensions have been conducted to demonstrate the effectiveness of our algorithm.

## ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China grant under number 61772427 and 61751202.

## REFERENCES

[1] David Arthur and Sergei Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.* Society for Industrial and Applied Mathematics, 1027–1035.

[2] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. 2005. Clustering with Bregman divergences. *Journal of machine learning research* 6, Oct (2005), 1705–1749.

[3] Shenglan Ben, Zhong Jin, and Jingyu Yang. 2011. Guided fuzzy clustering with multi-prototypes. In *Neural Networks (IJCNN), The 2011 International Joint Conference on.* IEEE, 2430–2436.

[4] James C Bezdek, Robert Ehrlich, and William Full. 1984. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences* 10, 2-3 (1984), 191–203.

[5] Francesco Camastra and Alessandro Verri. 2005. A novel kernel method for clustering. *IEEE transactions on pattern analysis and machine intelligence* 27, 5 (2005), 801–805.

[6] Robert L Cannon, Jitendra V Dave, and James C Bezdek. 1986. Efficient implementation of the fuzzy c-means clustering algorithms. *IEEE transactions on pattern analysis and machine intelligence* 2 (1986), 248–255.

[7] Fan RK Chung. 1997. *Spectral graph theory.* Number 92. American Mathematical Soc.

[8] Inderjit S Dhillon. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 269–274.

[9] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. 2004. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 551–556.

[10] Chris Ding and Xiaofeng He. 2004. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning.* ACM, 29.

[11] Ky Fan. 1949. On a theorem of Weyl concerning eigenvalues of linear transformations I. *Proceedings of the National Academy of Sciences* 35, 11 (1949), 652–655.

[12] Maurizio Filippone, Francesco Camastra, Francesco Masulli, and Stefano Rovetta. 2008. A survey of kernel and spectral methods for clustering. *Pattern recognition* 41, 1 (2008), 176–190.

[13] M. Girolami. 2002. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks* 13, 3 (May 2002), 780–784. https://doi.org/10.1109/TNN.2002.1000150

[14] Mehmet Gönen and Adam A Margolin. 2014. Localized data fusion for kernel k-means clustering with application to cancer biology. In *Advances in Neural Information Processing Systems.* 1305–1313.

[15] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. 1998. CURE: an efficient clustering algorithm for large databases. In *ACM Sigmod Record*, Vol. 27. ACM, 73–84.

[16] Greg Hamerly and Charles Elkan. 2004. Learning the k in k-means. In *Advances in neural information processing systems.* 281–288.

[17] Anil K. Jain and Richard C. Dubes. 1988. *Algorithms for Clustering Data.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[18] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. 1999. Data clustering: a review. *ACM computing surveys (CSUR)* 31, 3 (1999), 264–323.

[19] Jiye Liang, Liang Bai, Chuangyin Dang, and Fuyuan Cao. 2012. The $K$-Means-Type Algorithms Versus Imbalanced Data Distributions. *IEEE Transactions on Fuzzy Systems* 20, 4 (2012), 728–745.

[20] Manhua Liu, Xudong Jiang, and Alex C Kot. 2009. A multi-prototype clustering algorithm. *Pattern Recognition* 42, 5 (2009), 689–698.

[21] Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE transactions on information theory* 28, 2 (1982), 129–137.

[22] Ting Luo, Caiming Zhong, Hong Li, and Xia Sun. 2010. A multi-prototype clustering algorithm based on minimum spanning tree. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on*, Vol. 4. IEEE, 1602–1607.

[23] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.

[24] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On Spectral Clustering: Analysis and an Algorithm. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01).* MIT Press, Cambridge, MA, USA, 849–856.

[25] Feiping Nie, Lai Tian, and Xuelong Li. 2018. Multiview Clustering via Adaptively Weighted Procrustes. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '18).* ACM, New York, NY, USA, 2022–2030.

[26] Feiping Nie, Xiaoqian Wang, Cheng Deng, and Heng Huang. 2017. Learning A Structured Optimal Bipartite Graph for Co-Clustering. In *Advances in Neural Information Processing Systems.* 4132–4141.

[27] Feiping Nie, Xiaoqian Wang, and Heng Huang. 2014. Clustering and projected clustering with adaptive neighbors. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 977–986.

[28] Feiping Nie, Xiaoqian Wang, Michael I. Jordan, and Heng Huang. 2016. The Constrained Laplacian Rank Algorithm for Graph-based Clustering. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16).* AAAI Press, 1969–1976.

[29] Nikhil R Pal and James C Bezdek. 1995. On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy systems* 3, 3 (1995), 370–379.

[30] Enrique H Ruspini. 1969. A new approach to clustering. *Information and control* 15, 1 (1969), 22–32.

[31] Jianbo Shi and Jitendra Malik. 2000. Normalized Cuts and Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 22, 8 (Aug. 2000), 888–905.

[32] Chin-Wang Tao. 2002. Unsupervised fuzzy clustering with multi-center clusters. *Fuzzy Sets and Systems* 128, 3 (2002), 305–322.

[33] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17, 4 (2007), 395–416.

[34] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. 2001. Constrained k-means clustering with background knowledge. In *ICML*, Vol. 1. 577–584.

[35] Chang-Dong Wang, Jian-Huang Lai, Ching Y Suen, and Jun-Yong Zhu. 2013. Multi-exemplar affinity propagation. *IEEE transactions on pattern analysis and machine intelligence* 35, 9 (2013), 2223–2237.

[36] Yangtao Wang and Lihui Chen. 2016. K-MEAP: Multiple Exemplars Affinity Propagation With Specified $K$ Clusters. *IEEE transactions on neural networks and learning systems* 27, 12 (2016), 2670–2682.

[37] Jinglin Xu, Junwei Han, Kai Xiong, and Feiping Nie. 2016. Robust and Sparse Fuzzy K-means Clustering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI'16).* AAAI Press, 2224–2230.

[38] Lihi Zelnik-Manor and Pietro Perona. 2005. Self-tuning spectral clustering. In *Advances in neural information processing systems.* 1601–1608.

[39] Hongyuan Zha, Xiaofeng He, Chris Ding, Ming Gu, and Horst D Simon. 2002. Spectral relaxation for k-means clustering. In *Advances in neural information processing systems.* 1057–1064.