

A multi-view K-multiple-means clustering method

ZHANG Nini^{1,2}, GE Hongwei^{1,2}

(1. School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China;

2. Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 214122, China)

Abstract: The K-multiple-means (KMM) retains the simple and efficient advantages of the K-means algorithm by setting multiple subclasses, and improves its effect on non-convex data sets. And aiming at the problem that it cannot be applied to the Internet on a multi-view data set, a multi-view K-multiple-means (MKMM) clustering method is proposed in this paper. The new algorithm introduces view weight parameter, reserves the design of setting multiple subclasses, makes the number of clusters as constraint and obtains clusters by solving optimization problem. The new algorithm is compared with some popular multi-view clustering algorithms. The effectiveness of the new algorithm is proved through the analysis of the experimental results.

Key words: K-multiple-means (KMM) clustering; weight parameters; multi-view K-multiple-means (MKMM) method

0 Introduction

Clustering is a common technology for pattern recognition and is widely applied to machine learning task such as image segmentation and user portrait. K-means^[1] is the most classical method among a large number of existing clustering algorithms. And it is widely used due to its high efficiency and intuitive principle. However, the K-means algorithm does not perform well on non-spherical datasets. In order to improve the defect of K-means, many variants algorithm of K-means algorithm have been proposed^[2-5]. Setting multiple subclasses for a class is a strategy to improve the K-means algorithm. This design is more in line with the practical application scenarios, and performs better on non-convex datasets. K-multiple-means (KMM), a multiple-means clustering method with specified K ^[6] proposed by Nie et al. at ACM SIGKDD in 2019, is a typical example. KMM algorithm is concerned because of its excellent performance^[5-7].

KMM is a traditional clustering method to study samples through a set of characteristics. When the samples has multiple groups of features, multi-view clustering can divide the samples by integrating and

processing multiple groups of features. In the era of big data, data with multiple sets of characteristics is very common in real scenarios^[7]. For example, in the problem of understanding multimedia content, multimedia content contains both the video signal from the camera and the audio signal from the microphone. In some image recognition tasks, image features include colour feature, textures feature and shape feature. The proliferation of multi-view data makes many scholars have a strong interest in multi-view learning. The data on different views of multi-view data is heterogeneous but potentially related. In other word, in multi-view data, each individual view has a specific attribute for a specific knowledge discovery task, but the different views often contain complementary information. Therefore, how to use this information to reveal the potential value of multi-perspective data is very important in big data research. In terms of unsupervised learning, the clustering method based on single view can not solve problems by using multi-view information as effectively as the multi-view clustering in many cases. Multi-view clustering needs to judge relationship among samples in each view respectively and completes clustering task by using the complementary

Received date: 2021-07-05

Foundation items: National Youth Natural Science Foundation of China (No. 61806006); Innovation Program for Graduate of Jiangsu Province (No. KYLX160-781); Project Supported by Jiangsu University Superior Discipline Construction Project

Corresponding author: ZHANG Nini (2423776510@qq.com)

and consensus information of multiple views. It is difficult to obtain good result by integrating different views to a single view and then using the advanced traditional clustering algorithm to cluster. Because each view has its specific attributes, in the process of feature fusion, a particular view may have higher weight than other views, resulting in clustering relies on only one of the views.

Although KMM algorithm has excellent performance in traditional clustering, similar to other single-view clustering methods, it cannot use multiple groups of feature in multi-view data to complete the clustering task effectively. In this paper, a multi-view K-multiple-means (MKMM) clustering method is proposed. In the new algorithm, weight parameters of view are introduced, a new objective function is proposed, and the optimum allocation of weight parameters and clustering results of data are obtained through alternating optimization strategy.

For a better clustering performance on multi-view data, it's an effective strategy that modify good traditional clustering method to adapt to multi-view datasets. Recently, KMM and MKMM has excellent performance on traditional data sets. This paper proposed a new multi-view clustering based on KMM and revisited KMM in this section.

The key idea of KMM algorithm is to set multiple prototypes for each cluster and make the location of clustering prototype and the partitioning of data to prototype an optimization problem.

Denote $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbf{R}^{n \times d}$ as data sample matrix, $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_m]^T \in \mathbf{R}^{m \times d}$ as prototype matrix. The connected probability of i th sample \mathbf{x}_i and j th prototype \mathbf{a}_j is s_{ij} . In most instances, the smaller distance $\|\mathbf{x}_i - \mathbf{a}_j\|_2^2$ between \mathbf{x}_i and \mathbf{a}_j is, the bigger value of s_{ij} is. Therefore, the location of prototypes and assignment of data samples to neighboring prototypes can be written as

$$\min_{\mathbf{S}} \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i - \mathbf{a}_j\|_2^2 + \gamma \|\mathbf{S}\|_F^2, \quad (1)$$

s. t. $\mathbf{S} \geq \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{S} \in \Omega.$

The position of prototypes will change as \mathbf{S} changes. When \mathbf{S} updated, each prototype can be relocated. This process can be iteratively performed by

$$\min_{\mathbf{A}, \mathbf{S}} \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i - \mathbf{a}_j\|_2^2 + \gamma \|\mathbf{S}\|_F^2, \quad (2)$$

s. t. $\mathbf{S} \geq \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{S} \in \Omega, \mathbf{A} \in \mathbf{R}^{m \times d}.$

The clustering result can be obtained by solving problem (2).

KMM performs well in traditional clustering, so the multi-view clustering based on KMM has a good foundation for multi-view data clustering.

1 Proposed method

1.1 Design of objective function

As mentioned above, KMM sets multiple prototypes in each class for better performance (Fig. 1), and solving problem (1) can obtain the assignment of data to neighboring prototypes. Supposing that the data set has n_v views, this paper integrates information from all views by introducing weight parameters $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{n_v}]$ to extend KMM algorithm to multi-view data. Then, problem (1) becomes

$$\min_{\mathbf{S}} \sum_{v=1}^{n_v} \mathbf{w}_v^2 \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2 + \gamma \|\mathbf{S}\|_F^2, \quad (3)$$

s. t. $\mathbf{S} \geq \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1},$

where $\mathbf{x}_i^{(v)}$ is the i th sample on v th view; $\mathbf{a}_j^{(v)}$ is the j th sample on v th view, the explanation of matrix \mathbf{S} and parameters γ is the same as the the explanation of Eq. (1).

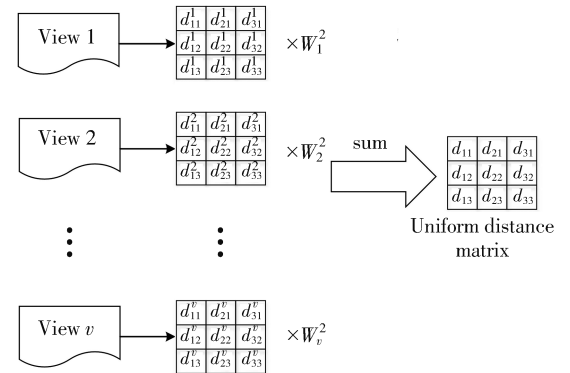


Fig. 1 Integration of multi-view schematic

Similar to the single view, the task assigned to a similar prototype for each data sample are independent of the other data samples in the multi-view. However, when data is partitioned into similar prototypes, the information of different views will influence each other and jointly determine the partition of sample. Therefore, the assignment of each sample \mathbf{x}_i is presented as

$$\min_{s_i} \left\| s_i - \frac{\mathbf{d}_i^x}{2\gamma} \right\|_2^2, \quad \text{s. t. } s_j \geq \mathbf{0}, \mathbf{s}_i^T \mathbf{1} = \mathbf{1}. \quad (4)$$

Denote $\mathbf{d}_i^{x(v)} = \|\mathbf{x}_i^{(v)} - \mathbf{a}_i^{(v)}\|_2^2$, \mathbf{d}_i^x is the weighted sum of $\mathbf{d}_i^{x(v)}$, consisting of j elements $\mathbf{d}_i^j =$

$$\sum_v w_v^2 \|\mathbf{x}_i^{(v)} - \mathbf{a}_i^{(v)}\|_2^2.$$

Problem (4) can be solved based on the method proposed in Ref. [10]. Denote $\mathbf{d}_{ij}^{\text{th}}$ as the j th distance of m prototypes to i th data point ordered from small to large. γ can be set as $\frac{1}{n} \sum_{i=1}^n \left(\frac{\tilde{k}}{2} \mathbf{d}_{i(\tilde{k}+1)}^{\text{th}} - \frac{1}{2} \sum_{j=1}^{\tilde{k}} \mathbf{d}_{ij}^{\text{th}} \right)$, \tilde{k} represents the number of nearest prototypes connected to the sample. If the j th prototype is not among the nearest \tilde{k} prototypes to i th sample, the value of s_{ij} is $\mathbf{0}$. Otherwise, supposing j th prototype is \tilde{j} th prototype nearest to i th sample, the value of s_{ij} is $\frac{\mathbf{d}_{i(\tilde{k}+1)}^{\text{th}} - \mathbf{d}_{ij}^{\text{th}}}{\tilde{k} \mathbf{d}_{i(\tilde{k}+1)}^{\text{th}} - \sum_{j=1}^{\tilde{k}} \mathbf{d}_{ij}^{\text{th}}}$.

After the matrix \mathbf{S} is updated, each prototype will be relocated according to the average of new subclass. This process is performed iteratively until the partition of sample is no longer changed which can be expressed as

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{S}} \sum_{v=1}^{n_v} w_v^2 \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2 + \gamma \|\mathbf{S}\|_F^2, \\ \text{s. t. } \mathbf{S} \geq \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{A} \in \mathbf{R}^{m \times d}. \end{aligned} \quad (5)$$

As with single-view clustering, in most cases, connecting samples to prototypes belonging to the same cluster will result in a connected graph. In order to make the partition of samples more reasonable, considering that the matrix \mathbf{S} should have k connected components, MKMM introduces a new constraint $\mathbf{S} \in \Omega$ to the objective function (5) to form a new objective, that is

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{S}} \sum_{v=1}^{n_v} w_v^2 \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2 + \gamma \|\mathbf{S}\|_F^2, \\ \text{s. t. } \mathbf{S} \geq \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{S} \in \Omega, \mathbf{A} \in \mathbf{R}^{m \times d}. \end{aligned} \quad (6)$$

In order to get the best view weight parameters distribution, the view may initially be assigned a weight average. When the partition between the prototype and the samples is updated again, the view weights will be updated and changed accordingly to obtain the best weight. Eq. (6) should be turned as

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{S}, \mathbf{W}} \sum_{v=1}^{n_v} w_v^2 \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2 + \gamma \|\mathbf{S}\|_F^2, \\ \text{s. t. } \mathbf{S} \geq \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{S} \in \Omega, \mathbf{A} \in \mathbf{R}^{m \times d}, \\ \mathbf{W} \geq \mathbf{0}, \mathbf{W}\mathbf{1} = \mathbf{1}. \end{aligned} \quad (7)$$

1.2 Optimization strategy

Denote $\mathbf{P} = \begin{bmatrix} \mathbf{S} \\ \mathbf{S}^T \end{bmatrix}$, $\tilde{\mathbf{L}}_s = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{P} \mathbf{D}^{-\frac{1}{2}}$ is the

normalized Laplacian matrix $\tilde{\mathbf{L}}_s$ associated with \mathbf{S} , where matrix $\mathbf{D} \in \mathbf{R}^{(n+m) \times (n+m)}$ is the degree matrix of \mathbf{P} . According to the discussion in preliminary, in order to partition prototypes into clusters, the restriction $\text{rank}(\tilde{\mathbf{L}}_s) = (n+m) - k$ is replaced with the restriction $\mathbf{S} \in \Omega$ in Eq. (7), then

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{S}, \mathbf{W}} \sum_{v=1}^{n_v} w_v^2 \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2 + \gamma \|\mathbf{S}\|_F^2, \\ \text{s. t. } \mathbf{S} \geq \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1}, \text{rank}(\tilde{\mathbf{L}}_s) = (n+m) - k, \\ \mathbf{A} \in \mathbf{R}^{m \times d}, \mathbf{W} \geq \mathbf{0}, \mathbf{W}\mathbf{1} = \mathbf{1}. \end{aligned} \quad (8)$$

In order to solve the problem (8), relax the restriction in problem (8) and turn the problem (8) into

$$\begin{aligned} \min_{\mathbf{A}, \mathbf{S}, \mathbf{W}, \mathbf{F}} \sum_{v=1}^{n_v} w_v^2 \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2 + \gamma \|\mathbf{S}\|_F^2 + \\ \lambda \text{Tr}(\mathbf{F}^T \tilde{\mathbf{L}}_s \mathbf{F}), \text{ s. t. } \mathbf{S} \geq \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1}, \mathbf{A} \in \mathbf{R}^{m \times d}, \\ \mathbf{F} \in \mathbf{R}^{(n+m) \times k}, \mathbf{F}^T \mathbf{F} = \mathbf{I}, \mathbf{W} \geq \mathbf{0}, \mathbf{W}\mathbf{1} = \mathbf{1}. \end{aligned} \quad (9)$$

Eq. (9) can be solved by update \mathbf{A} , \mathbf{S} , \mathbf{F} , \mathbf{W} iteratively. Fix \mathbf{A} firstly, then update \mathbf{S} , \mathbf{F} , \mathbf{W} . Problem (9) can be turned into

$$\begin{aligned} \min_{\mathbf{S}, \mathbf{W}, \mathbf{F}} \sum_{v=1}^{n_v} w_v^2 \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2 + \gamma \|\mathbf{S}\|_F^2 + \\ \lambda \text{Tr}(\mathbf{F}^T \tilde{\mathbf{L}}_s \mathbf{F}), \text{ s. t. } \mathbf{S} \geq \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1}, \\ \mathbf{F} \in \mathbf{R}^{(n+m) \times k}, \mathbf{F}^T \mathbf{F} = \mathbf{I}, \mathbf{W} \geq \mathbf{0}, \mathbf{W}\mathbf{1} = \mathbf{1}. \end{aligned} \quad (10)$$

For solving problem (10), fix \mathbf{S} , \mathbf{W} and update \mathbf{F} . The optimization of \mathbf{F} is the same as KMM. Rewriting \mathbf{F} as $\begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix}$, \mathbf{D} as $\begin{bmatrix} \mathbf{D}_U & \\ & \mathbf{D}_V \end{bmatrix}$, \mathbf{U} , \mathbf{V} are the first k left singular vectors and right singular vectors of $\frac{\sqrt{2}}{2}$ times $\mathbf{D}_U^{-\frac{1}{2}} \mathbf{S} \mathbf{D}_V^{-\frac{1}{2}}$, respectively.

When \mathbf{F} is fixed, fix \mathbf{W} , update \mathbf{S} and change the problem (10) as

$$\begin{aligned} \min_{\mathbf{S}} \sum_{v=1}^{n_v} w_v^2 \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2 + \gamma \|\mathbf{S}\|_F^2 + \\ \lambda \text{Tr}(\mathbf{F}^T \tilde{\mathbf{L}}_s \mathbf{F}), \text{ s. t. } \mathbf{S} \geq \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1}. \end{aligned} \quad (11)$$

According to the definition of $\tilde{\mathbf{L}}_s$, Eq. (12) can be obtained as

$$\text{Tr}(\mathbf{F}^T \tilde{\mathbf{L}}_s \mathbf{F}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m s_{ij} \left\| \frac{\mathbf{f}_i}{\sqrt{d_i}} - \frac{\mathbf{f}_{n+j}}{\sqrt{d_{n+j}}} \right\|_2^2. \quad (12)$$

Denote $\mathbf{v}_{ij} = \left\| \frac{\mathbf{f}_i}{\sqrt{d_i}} - \frac{\mathbf{f}_{n+j}}{\sqrt{d_{n+j}}} \right\|_2^2$, problem (12) can be

written as

$$\min_{\mathbf{S}} \sum_{v=1}^{n_v} \mathbf{w}_v^2 \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2 + \lambda s_{ij} \mathbf{v}_{ij} + \gamma \|\mathbf{S}\|_F^2, \quad (13)$$

s. t. $\mathbf{S} \geq \mathbf{0}, \mathbf{S}\mathbf{1} = \mathbf{1}.$

Because the partition of each sample to the prototype is independent of each other, the update of \mathbf{S} can be regarded as the update of each s_i . Denote $\tilde{\mathbf{d}}_i^x = \sum_{v=1}^{n_v} \mathbf{w}_v^2 \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2$, $\tilde{\mathbf{d}}_{ij} = \tilde{\mathbf{d}}_i^x + \lambda \mathbf{v}_{ij}$, problem (13) can be written as

$$\min_{s_i} \left\| s_i - \frac{1}{2\gamma} \tilde{\mathbf{d}}_i \right\|_2^2, \quad \text{s. t. } s_i \geq 0, s_i^T \mathbf{1} = \mathbf{1}. \quad (14)$$

The solution of problem (14) is similar to that of problem (4).

When \mathbf{S}, \mathbf{F} fixed, it needs to update \mathbf{W} , which is used to solve the problem (14), expressed as

$$\min_{\mathbf{W}} \sum_{v=1}^{n_v} \mathbf{w}_v^2 \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2, \quad (15)$$

s. t. $\mathbf{W} \geq \mathbf{0}, \mathbf{W}\mathbf{1} = \mathbf{1}.$

Denote $\mathbf{y}_v = \sum_{i=1}^n \sum_{j=1}^m s_{ij} \|\mathbf{x}_i^{(v)} - \mathbf{a}_j^{(v)}\|_2^2$, problem (15) can be written as

$$\min_{\mathbf{W}} \sum_{v=1}^{n_v} \mathbf{w}_v^2 \mathbf{y}_v, \quad \text{s. t. } \mathbf{W} \geq \mathbf{0}, \mathbf{W}\mathbf{1} = \mathbf{1}. \quad (16)$$

Denote $\mathbf{Y} \in \mathbf{R}^{n_v \times n_v}$, problem (16) can be regarded as

$$\min_{\mathbf{W}} \mathbf{W}^T \mathbf{Y} \mathbf{W}, \quad \text{s. t. } \mathbf{W} \geq \mathbf{0}, \mathbf{W}\mathbf{1} = \mathbf{1}. \quad (17)$$

Removing the constraint $\mathbf{W} \geq \mathbf{0}$, the Lagrange function of Eq. (17) can be expressed as

$$\mathbf{L}(\mathbf{w}, \eta) = \mathbf{W}^T \mathbf{Y} \mathbf{W} - \eta (\mathbf{W}^T \mathbf{1} - \mathbf{1}), \quad (18)$$

where η is Lagrange multiplier. According to KKT condition, the value of $\frac{\partial \mathbf{L}(\mathbf{w}, \eta)}{\partial \mathbf{w}}$ is 0. According to $\mathbf{W}^T \mathbf{1} = \mathbf{1}$, \mathbf{W} can be explained by

$$\mathbf{w}_v = \frac{1}{\mathbf{y}_v} \times \left(\sum_{r=1}^{n_v} \frac{1}{\mathbf{y}_r} \right)^{-1}. \quad (19)$$

When $\tilde{\mathbf{L}}_s$ satisfies $\sum_{i=1}^k \sigma_i(\tilde{\mathbf{L}}_s) = \mathbf{0}$ and $\sum_{i=1}^{k+1} \sigma_i(\tilde{\mathbf{L}}_s) > \mathbf{0}$, stop updating $\mathbf{S}, \mathbf{F}, \mathbf{W}$ iteratively. When $\mathbf{S}, \mathbf{F}, \mathbf{W}$ updated, update prototype matrix \mathbf{A} by

$$\mathbf{a}_j^{(v)} = \sum_{i=1}^n \frac{s_{ij} \mathbf{x}_i^{(v)}}{s_{ij}}. \quad (20)$$

1.3 MKMM algorithm steps

In summary, the main steps of the MKMM

algorithm are described as:

Input: multi-view data set $\mathbf{X}^{n_v} = \{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \dots, \mathbf{X}^{(n_v)}\}$, number of clusters k , number of subclass m , parameter γ, λ .

Output: clustering results $\mathbf{C} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_k\}$.

Step 1: calculate matrix \mathbf{S} ;

Step 2: update $\mathbf{F} = \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix}$, where \mathbf{U}, \mathbf{V} are the first k left singular vectors and right singular vectors of $\frac{\sqrt{2}}{2} \mathbf{D}_U^{-\frac{1}{2}} \mathbf{S} \mathbf{D}_V^{-\frac{1}{2}}$;

Step 3: solving problem (14) to update matrix \mathbf{S} .

Step 4: update weight parameter \mathbf{W} according to Eq. (20). Repeat steps 2, 3, 4 until converge.

Step 5: calculate the position of each prototype and update the matrix \mathbf{A} until the position of the prototype no longer changes according to the matrix \mathbf{S} ;

Step 6: the k clusters are obtained according to the bipartite graph composed of samples and prototypes.

1.4 Computational complexity

The cost of updating \mathbf{F} is $O(m^3 + m^2 n)$ and mainly produced by calculating the singular value decomposition of $\tilde{\mathbf{S}}$. The time complexity required to update \mathbf{S} is $O(nmk + nm \log m)$. The cost of updating the view weight parameter is $O(n_v)$. Supposing the number of iterations to update \mathbf{S} and \mathbf{F} is t_1 . Normally, the values of m^3 and $\log m$ are small, so the time complexity of updating \mathbf{S} and \mathbf{F} iteratively is $O((nmk + nmc + m^2 n + n_v) t_1)$. The time complexity required to update \mathbf{A} is $O(nmd)$.

In summary, supposing that \mathbf{A} is updated t_2 times, and the total time complexity of the KMM algorithm is

$$O((nmk + nmc + m^2 n + n_v) t_1 + nmd) t_2 + n \log n + nd).$$

2 Experiments

2.1 Clustering metrics

The measure of clustering generally uses one or more evaluation indicators as judgment criteria to evaluate and analyze the clustering results, so as to determine the quality of the clustering algorithm. This article evaluates the algorithm through *Accuracy*, *Normalized Mutual Information (NMI)*^[11] and *Purity* metrics. The three metrics are introduced, respectively.

Supposing \mathbf{C} is the real label of dataset, \mathbf{W} is the label obtained by the algorithm. The definition of

Accuracy is

$$Accuracy(\mathbf{W}, \mathbf{C}) = \frac{\sum_{i=1}^n \delta(c_i, \text{map } \mathbf{w}_i)}{n}, \quad (21)$$

where map is the best mapping function, which can transform the label obtained by the algorithm and the real label into one to one mapping relationship. δ is the indicator function.

The definition of NMI is

$$NMI(\mathbf{W}, \mathbf{C}) = \frac{\mathbf{I}(\mathbf{W}, \mathbf{C})}{\sqrt{\mathbf{H}(\mathbf{W})\mathbf{H}(\mathbf{C})}}, \quad (22)$$

where \mathbf{I} represents mutual information; \mathbf{H} represents information entropy.

Purity is defined as

$$Purity(\mathbf{W}, \mathbf{C}) = \frac{\sum_k \max_j |\omega_k \cap c_j|}{n}. \quad (23)$$

The larger the values of the Accuracy, NMI and Purity are, the better the clustering performance. The value range of the metrics is $[0, 1]$.

2.2 Datasets

The Caltech101-7 in Table 1 is a dataset of 7 classes extracted from Caltech101. The Caltech101 is a dataset containing 101 classes of images created by the California University in 2003. Each class contains 40 to 80 pictures, each with assize of 300×200 pixels. The seven classes extracted by Caltech101-7 are human faces, motorcycles, dollar, garfield, snoopy, stop signs and Windsor chairs. Caltech101-7 extracted six features from the above seven classes images containing gabor, wavelet cenhist, hog, gist and lbp.

The Caltech101-20 in Table 1 is a dataset of 20 classes extracted from the Caltech101. The 20 classes are human face, leopard, motorcycle, binoculars, brain, camera, car sidewall, dollar, ferry, garfield, hedgehog, pagoda, rhino, snoopy, stapler, parking indicator, water lily, Windsor chair, wrench, and yinyang. The six features extracted from the images by Caltech101-20 are the same as Caltech101-20.

Table 1 Real benchmark dataset

Datasets	Samples	Views	Clusters
Caltech101-7	1 474	6	7
Caltech101-20	2 386	6	20
Yale32	165	3	15
Wikipedia articles	693	2	10

The Yale32 in Table 1 is a face dataset of Yale University. There are 15 people in the dataset. Each

person has 11 pictures with different poses, expressions and lighting. There are 165 pictures in total, and the pixels of the pictures are 32×32 pixels.

The Wikipedia Articles in Table 1 is some documents collected from the featured article in Wikipedia. This is a continuously growing data set. This article uses a dataset composed of 2 669 articles in 29 classes collected in October 2009. Because some of the classes are sparse, only 10 classes are retained, and dataset are pruned to retain 693 samples. The dataset has two sets of features, one set is taken from the text information in the document and the other set is taken from the text.

2.3 Competitors

In order to evaluate the MKMM algorithm proposed in this paper, some algorithms are selected.

Liu et al. proposed MKKM-MR^[12] algorithm. This algorithm designed a novel, effective matrix-induced regularization to reduce such redundancy and enhance the diversity of the selected kernels. The algorithm needed to set the regularization parameters in advance when the algorithm was executed. According to the recommendations in the article, this article sets the parameter range from -15 to 20, sets the step size as 1 for comparison experiments, and takes the average of 26 results of algorithm.

Zhao et al. proposed SCMK1^[13] algorithm. This algorithm learned similarity information from data and integrated three subtask of traditional spectral clustering into a unified framework. The parameters were tuned as suggested^[13].

Zhao et al. proposed SCMK2^[14] algorithm. This algorithm proposed a model to simultaneously learn cluster indicator matrix and similarity information in kernel spaces in a principles way.

Wang et al. proposed MVC_LFA^[15] algorithm. This algorithm proposed to maximally align the consensus partition with the weighted base partitions. According to the recommendations, the balance parameters were set according to the values in set $\{2^{-15}, 2^{-14}, \dots, 2^{15}\}$, and the average value of 31 experiments was displayed.

2.4 Performance

Results of the MKMM algorithm and the competitors on the four multi-view datasets were evaluated by Accuracy, NMI and Purity metric, which was shown in Tables 2–4.

Table 2 Accuracy values of five algorithms

Datasets	MKKM-MR	SCMK1	SCMK2	MVC_LFA	MKMM
Caltech101-7	0.405 4	0.510 0	0.550 0	0.370 4	0.631 3
Caltech101-20	0.377 7	0.540 0	0.560 0	0.391 7	0.578 7
Yale32	0.554 3	0.632 5	0.569 7	0.566 2	0.856 3
Wikipedia articles	0.546 1	0.532 5	0.554 1	0.539 8	0.594 0

Table 3 NMI values of five algorithms

Datasets	MKKM-MR	SCMK1	SCMK2	MVC_LFA	MKMM
Caltech101-7	0.395 8	0.400 0	0.490 0	0.391 1	0.461 4
Caltech101-20	0.518 5	0.620 0	0.490 0	0.518 5	0.492 0
Yale32	0.581 4	0.610 4	0.565 2	0.572 9	0.903 6
Wikipedia articles	0.5	0.472 6	0.514 6	0.480 4	0.558 1

Table 4 Purity values of five algorithms

Datasets	MKKM-MR	SCMK1	SCMK2	MVC_LFA	MKMM
Caltech101-7	0.819 1	0.656 7	0.600 0	0.815 3	0.824 0
Caltech101-20	0.738 4	0.694 5	0.657 3	0.740 5	0.688 0
Yale32	0.562 2	0.673 9	0.600 0	0.568 7	0.883 2
Wikipedia articles	0.583 4	0.583 0	0.624 8	0.583 9	0.630 4

In order to eliminate the influence of the initial prototypes selection of the MKMM algorithm, the average value of 30 experiments performed by the algorithm on the dataset is used for display. From the Tables 2 – 4, it can be seen that the MKMM algorithm has outstanding performance in some datasets compared with other popular multi-view clustering algorithms. Although the MKMM algorithm performs not the best on other datasets, it still achieves a good clustering effect. MKMM performs particularly well on Yale32, probably because the same person in different facial expressions can be taken as in distinct subclasses. MKMM may be more advantageous on similar datasets to Yale32.

3 Conclusions

Considering that the KMM algorithm cannot solve the problem of multi-view clustering, this paper proposes MKMM algorithm. The algorithm introduces view weight parameter, designs a new objective function and effectively uses multiple features to achieve better clustering results. However, affected by the initial point selection like the KMM algorithm, so that the clustering results of the MKMM algorithm are unstable. Therefore, how to make the MKMM algorithm select prototypes on the multi-view dataset scientifically and improve the

clustering effect while stabilizing the clustering performance will be the next research work.

References

- [1] MacQueen J. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967: 281-297.
- [2] Liu M, Jiang X, Kot A C. A multi-prototype clustering algorithm. Pattern Recognition, 2009, 42(5): 689-698.
- [3] Luo T, Zhong C, Li H, et al. A multi-prototype clustering algorithm based on minimum spanning tree. In: Proceedings of the 7th International Conference on Fuzzy Systems and Knowledge Discovery, 2010: 1602-1607.
- [4] Wang C D, Lai J H, Zhu J Y, et al. Multi-exemplar affinity propagation. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(9): 2223-2237.
- [5] Wang Y, Chen L. K-MEAP: multiple exemplars affinity propagation with specified K clusters. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(12): 2670-2682.
- [6] Nie F, Wang C L, Li X. K-multiple-means: A multiple-means clustering method with specified k clusters. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019: 959-967.
- [7] Yang Y, Wang H. Multi-view clustering: a survey. Big Data Mining and Analytics, 2018, 1(2): 83-107.
- [8] Chung F R K, Graham F C. Spectral graph theory. American Mathematical Society, 1997.
- [9] Fan K. On a theorem of weyl concerning eigenvalues of linear transformations II. In: Proceedings of the National

- Academy of Sciences, 1950, 41(1): 652.
- [10] Nie F, Wang X, Huang H. Clustering and projected clustering with adaptive neighbors. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York: ACM, 2014: 977-986.
- [11] Strehl A, Ghosh J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 2002, 3(3): 583-617.
- [12] Liu X, Dou Y, Yin J, et al. Multiple kernel k-means clustering with matrix-induced regularization. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016: 1888-1894.
- [13] Kang Z, Peng C, Cheng Q, et al. Unified spectral clustering with optimal graph. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2018.
- [14] Kang Z, Peng C, Cheng Q. Twin learning for similarity and clustering: a unified kernel approach. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017: 2080-2086.
- [15] Wang S, Liu X, Zhu E, et al. Multi-view clustering via late fusion alignment maximization. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019: 3778-3784.

多视图 K-多均值聚类算法

张倪妮^{1,2}, 葛洪伟^{1,2}

(1. 江南大学 人工智能与计算机学院, 江苏 无锡 214122;

2. 江南大学 江苏省模式识别与计算智能实验室, 江苏 无锡 214122)

摘要: 多均值聚类算法假设每个类拥有多个子类, 通过求解优化问题的方式来求解每个样本子类的划分和最终类簇的划分。该算法弥补了 K-均值算法在非球数据集上的劣势, 取得了较好的聚类效果, 但是该算法无法被运用到多视图数据集上。本文提出了一种多视图 K-多均值聚类算法, 保留了 K-多均值设置多个子类的设计, 引入了视图权重参数, 将目标聚类数作为限制条件, 通过求解最优问题获得最终的类簇。将本文提出的算法与流行的多视图聚类算法进行对比实验, 证明了本文算法的优越性。

关键词: K-多均值聚类; 权重参数; 多视图 K-多均值算法

引用格式: ZHANG Nini, GE Hongwei. A multi-view K-multiple-means clustering method. *Journal of Measurement Science and Instrumentation*, 2021, 12(4): 405-411. DOI: 10.3969/j.issn.1674-8042.2021.04.004