Probabilistic K-means Clustering via Nonlinear Programming

Yujian Li*, Bowen Liu, Zhaoying Liu, and Ting Zhang

Abstract—K-means is a classical clustering algorithm with wide applications. However, soft K-means, or fuzzy c-means at \( m = 1 \), remains unsolved since 1981. To address this challenging open problem, we propose a novel clustering model, i.e. Probabilistic K-Means (PKM), which is also a nonlinear programming model constrained on linear equalities and linear inequalities. In theory, we can solve the model by active gradient projection, while inefficiently. Thus, we further propose maximum-step active gradient projection and fast maximum-step active gradient projection to solve it more efficiently. By experiments, we evaluate the performance of PKM and how well the proposed methods solve it in five aspects: initialization robustness, clustering performance, descending stability, iteration number, and convergence speed.

Index Terms—K-means, soft K-means, probabilistic K-means, fuzzy c-means, nonlinear programming, active gradient projection, maximum-step active gradient projection, fast maximum-step active gradient projection.

1 Introduction

Clustering analysis is an unsupervised machine learning method. It has been widely used in image and video processing [1]-[4], speech processing [5], biology [6], medicine [7], sociology [8], and so on. The task of clustering is to find some similarities from datasets [9], [10], and then to classify samples with the similarities into clusters (i.e. classes or categories). Clustering methods mainly include: partition-based clustering [11]-[13], hierarchical-based clustering [14]-[16], density-based clustering [10], [17], [18], graph-based clustering [19]-[21], grid-based clustering [22]-[24], model-based clustering [25]-[27] and subspace clustering [21], [24], [28]-[31], etc. Note that these methods may have intersections.

Partition-based clustering algorithms, such as K-means [11], K-medoids [12], and EM-K-means [32], divide a dataset into several disjoint subsets with a similarity criterion. In general, they first choose initial cluster centers randomly or manually, then adjust categories of samples and update the cluster centers according to the nearest neighbor principle until convergence. Hierarchical-based clustering algorithms fall into two types: bottom-up and top-down. The bottom-up clustering algorithms, such as CURE [14] and BIRCH [15], start with each sample treated as a separate cluster, and repeatedly merge two or more clusters until satisfying certain conditions, e.g. only one cluster left. The top-down clustering algorithms, like MMDCA [16], start with a single cluster of all samples, and repeatedly split one cluster into two or more dissimilar clusters until meeting some criteria.

- Y. Li is with Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China. As corresponding author, he is with School of Artificial Intelligence, Guilin University of Electronic Technology, Guilin 541004, China.
  E-mails: liyujian@guet.edu.cn
- B. Liu, Z. Liu and T. Zhang are with Faculty of Information Technology, Beijing University of Technology, Beijing, China, 100124.
  E-mail: liuba2017@gmails.bjut.edu.cn, zhaoying.liu@bjut.edu.cn, zhangting@bjut.edu.cn

...
2 PKM Clustering Model

Given a set of data points \( X = \{ x_i | x_i \in \mathbb{R}^D, 1 \leq i \leq L \} \), let us divide it into \( K \) clusters. Suppose \( \omega_j \) is the \( j \)-th cluster.

Then, \( X = \bigcup_{j=1}^{K} \omega_j \), \( \forall 1 \leq i \neq j \leq K, \omega_i \cap \omega_j = \emptyset \). If \( L_j \) denotes the number of elements in \( \omega_j \) with the center of \( c_j \), and \( L = \sum_{i=1}^{K} L_j \), then the standard K-means clustering model is described as minimizing a hard objective function:

\[
J = \sum_{i=1}^{K} \sum_{x_i \in \omega_j} \| x_i - c_j \|^2
\]

where

\[
c_j = \frac{1}{L_j} \sum_{x_i \in \omega_j} x_i
\]

(2)

With membership degree \( w_{ij} \) assigned to the \( i \)-th data point in the \( j \)-th cluster for \( 1 \leq i \leq L \) and \( 1 \leq j \leq K \), the hard K-means model is extended to a soft model, namely the FCM model. FCM minimizes the objective function below:

\[
J = \sum_{j=1}^{K} \sum_{i=1}^{L} w_{ij} \| x_i - c_j \|^2
\]

s.t. \( \sum_{j=1}^{K} w_{ij} = 1, w_{ij} \geq 0 \)

where \( m \) ranges from 1 to infinity.

When \( m > 1 \), \( w_{ij} \) and \( c_j \) can be computed alternately as follows [13]

\[
\begin{align*}
  w_{ij} &= \frac{\| x_i - c_j \|^{-2}}{\sum_{k=1}^{K} \| x_i - c_k \|^{-2m}} \\
  c_j &= \frac{\sum_{i=1}^{L} w_{ij} x_i}{\sum_{i=1}^{L} w_{ij}}
\end{align*}
\]

(4)

However, FCM at \( m = 1 \) is an challenging open problem that remains unsolved since 1981. It is also called “soft K-means”. Replacing membership degree \( w_{ij} \) by probability \( p_{ij} \), we get the soft K-means clustering model as follows,

\[
J = \sum_{j=1}^{K} \sum_{i=1}^{L} p_{ij} \| x_i - c_j \|^2
\]

s.t. \( \sum_{j=1}^{K} p_{ij} = 1, p_{ij} \geq 0 \)

Note that probability \( p_{ij} \) takes a value in \([0, 1]\). This means that data point \( x_i \) may belong to class \( j \) with probability \( p_{ij} \), rather than to only one class in hard K-means. The higher the probability \( p_{ij} \), the more likely that \( x_i \) is in class \( j \). The view of probability is more in line with Bayesian decision theory [48].

For the soft K-means, we can construct a Lagrangian function,

\[
J = \sum_{j=1}^{K} \sum_{i=1}^{L} p_{ij} \| x_i - c_j \|^2 + \sum_{i=1}^{L} \lambda_i (\sum_{j=1}^{K} p_{ij} - 1)
\]

(6)

Letting \( \frac{\partial J}{\partial c_j} = \sum_{j=1}^{K} \sum_{i=1}^{L} 2 p_{ij} (c_j - x_i) = 0 \), we obtain

\[
c_j = \frac{\sum_{i=1}^{L} p_{ij} x_i}{\sum_{i=1}^{L} p_{ij}}
\]

(7)
Substituting (7) into (5), we get an equivalent model, namely, PKM

$$J = \sum_{j=1}^{K} \sum_{i=1}^{L} p_{ij} \left\| x_i - \frac{\sum_{i=1}^{L} p_{ij} x_i}{\sum_{i=1}^{L} p_{ij}} \right\|^2$$

\hspace{1cm} (8)

s.t. \( \sum_{j=1}^{K} p_{ij} = 1, p_{ij} \geq 0 \)

If setting \( K = 2 \) and \( L = 2 \) with \( x_1 = (1,1)^T \), \( x_2 = (2,2)^T \), we can rewrite the objective function as follows,

$$J(p_{11}, p_{12}, p_{21}, p_{22}) = \sum_{j=1}^{2} \sum_{i=1}^{2} p_{ij} \left\| x_i - \frac{\sum_{i=1}^{2} p_{ij} x_i}{\sum_{i=1}^{2} p_{ij}} \right\|^2$$

$$= \left[ p_{11} \left\| \frac{p_{21} x_1 - p_{21} x_2}{p_{11} + p_{21}} \right\|^2 + p_{21} \left\| \frac{p_{11} x_2 - p_{11} x_1}{p_{11} + p_{21}} \right\|^2 \right]$$

$$+ \left[ p_{12} \left\| \frac{p_{22} x_1 - p_{22} x_2}{p_{12} + p_{22}} \right\|^2 + p_{22} \left\| \frac{p_{12} x_2 - p_{12} x_1}{p_{12} + p_{22}} \right\|^2 \right]$$

$$= \frac{p_{11} p_{21} + p_{21} p_{22}}{(p_{11} + p_{21})(p_{12} + p_{22})} \left\| x_1 - x_2 \right\|^2$$

\hspace{1cm} (9)

where \( p_{12} = 1 - p_{11} \) (0 \( \leq \) \( p_{11} \leq \) 1) and \( p_{22} = 1 - p_{21} \) (0 \( \leq \) \( p_{21} \leq \) 1).

Therefore, \( J(p_{11}, p_{12}, p_{21}, p_{22}) = J(p_{11}, p_{21}) \) is indeed a function of \( p_{11} \) and \( p_{21} \), as shown in Fig. 1. Note that Fig. 1(a) is a surface of \( J(p_{11}, p_{21}) \), with its vertical view in Fig. 1(b). In Fig. 1(a), \( p_{11} \) is the probability of \( x_1 \) in cluster 1, and \( p_{21} \) is the probability of \( x_2 \) in cluster 1. Obviously, the function \( J(p_{11}, p_{21}) \) has countless probability pairs that take the same maximum value of 1, satisfying \( p_{11} = p_{21} \), e.g., (0.5, 0.5). But it has only two probability pairs that take the same minimum value of 0, namely (0, 1) or (1, 0). Because of symmetricty, the two pairs produce the same cluster assignation: \( x_1 \) in one class, and \( x_2 \) in the other.

3 Solutions to PKM

According to [6], PKM is a nonlinear programming problem constrained on linear equalities and linear inequalities. In theory, the problem can be solved by AGP, but requiring optimization of step length and fast computation of projection matrices.

In this section, we first calculate the gradient of PKM's objective function, and then propose the AGP method and its two improvements: the MSAGP method and the FM-SAGP method, to solve the PKM model.

3.1 Gradient Calculation

For convenience, we define \( F_k \) as

$$F_k = \left\| x_k - \frac{\sum_{i=1}^{L} p_{ij} x_i}{\sum_{i=1}^{L} p_{ij}} \right\|^2$$

\hspace{1cm} (10)

Using the chain rule on (8), we obtain

$$\frac{\partial J}{\partial p_{ij}} = F_i + \sum_{k=1}^{L} p_{kj} \frac{\partial F_k}{\partial p_{ij}}$$

\hspace{1cm} (11)

According to (10), we can further derive (12)-(15) as follows,

$$\frac{\partial F_k}{\partial p_{ij}} = \left( \sum_{i=1}^{L} p_{ij} x_i \right)^T \left( \sum_{i=1}^{L} p_{ij} x_i \right) - \left( \frac{\sum_{i=1}^{L} p_{ij} x_i}{\sum_{i=1}^{L} p_{ij}} \right)^T \left( \frac{\sum_{i=1}^{L} p_{ij} x_i}{\sum_{i=1}^{L} p_{ij}} \right)$$

\hspace{1cm} (12)

$$\frac{\partial F_k}{\partial p_{ij}} = -2 \frac{x_i^T x_i \sum_{i=1}^{L} p_{ij} x_i - \frac{\sum_{i=1}^{L} p_{ij} x_i}{\sum_{i=1}^{L} p_{ij}} \left( \sum_{i=1}^{L} p_{ij} x_i^T \right) \left( \sum_{i=1}^{L} p_{ij} x_i \right)}{\left( \sum_{i=1}^{L} p_{ij} \right)^2}$$

\hspace{1cm} (13)

$$\frac{\partial F_k}{\partial p_{ij}} = \left( \frac{\sum_{i=1}^{L} p_{ij} x_i}{\sum_{i=1}^{L} p_{ij}} \right)^T \left( \frac{\sum_{i=1}^{L} p_{ij} x_i}{\sum_{i=1}^{L} p_{ij}} \right) - \left( \sum_{i=1}^{L} p_{ij} x_i \right)^T \left( \sum_{i=1}^{L} p_{ij} x_i \right)$$

\hspace{1cm} (14)

$$\frac{\partial F_k}{\partial p_{ij}} = -2 \frac{x_i^T x_i \sum_{i=1}^{L} p_{ij} x_i - \frac{\sum_{i=1}^{L} p_{ij} x_i}{\sum_{i=1}^{L} p_{ij}} \left( \sum_{i=1}^{L} p_{ij} x_i^T \right) \left( \sum_{i=1}^{L} p_{ij} x_i \right)}{\left( \sum_{i=1}^{L} p_{ij} \right)^2}$$

\hspace{1cm} (15)
Substituting \( \text{(15)} \) into \( \text{(11)} \), we finally get,

\[
\frac{\partial J}{\partial p_{ij}} = \left| x_i - \frac{L}{\sum_{i=1}^L p_{ij}} \right|^2
\]

\[
-\frac{2}{L} \sum_{i=1}^L p_{ij} \left( x_k - \frac{L}{\sum_{i=1}^L p_{ij}} \right) \left( x_i - \frac{L}{\sum_{i=1}^L p_{ij}} \right) \]

\[
\frac{\partial J}{\partial p_{ij}} = \left| x_i - \frac{L}{\sum_{i=1}^L p_{ij}} \right|^T \left( x_i - \frac{L}{\sum_{i=1}^L p_{ij}} \right) \]

\[
-\frac{2}{L} \sum_{i=1}^L p_{ij} \left( x_k - \frac{L}{\sum_{i=1}^L p_{ij}} \right) \left( x_i - \frac{L}{\sum_{i=1}^L p_{ij}} \right) \]

\[
\frac{\partial J}{\partial p_{ij}} = \left| x_i - \frac{L}{\sum_{i=1}^L p_{ij}} \right|^T \left( x_i - \frac{L}{\sum_{i=1}^L p_{ij}} \right) \]

and the gradient

\[
\nabla J = \left[ \frac{\partial J}{\partial p_{i1}} \ldots \frac{\partial J}{\partial p_{iK}} \ldots \frac{\partial J}{\partial p_{L1}} \ldots \frac{\partial J}{\partial p_{LK}} \right]
\]

3.2 The AGP Method

In the constraints of PKM, there are \( L \) equalities and \( KL \) inequalities,

\[
\forall 1 \leq i \leq L, \sum_{j=1}^K p_{ij} = 1 \quad (19)
\]

\[
\forall 1 \leq i \leq L, 1 \leq j \leq K, p_{ij} \geq 0 \quad (20)
\]

where probability matrix \((p_{ij})_{L \times K}\) can be vectorized as

\[
P = \begin{bmatrix} p_{i1} & \ldots & p_{iK} & \ldots & p_{ij} & \ldots & p_{L1} & \ldots & p_{LK} \end{bmatrix}^T
\]

Let \( I_{LK \times LK} \) be the identity matrix of size \( LK \times LK \). Define two matrices, inequality matrix \( A \) and equality matrix \( E \),

\[
A = I_{LK \times LK}
\]

\[
E = \begin{bmatrix}
1 & \ldots & 1 & 0 & \ldots & 0 & 0 & 0 \\
0 & \ldots & 0 & 1 & \ldots & 0 & 0 & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \ldots & 0 & 0 & \ldots & 1 & \ldots & 1
\end{bmatrix}_{L \times LK}
\]

\[
AP \geq 0, EP = 1
\]

Let \( P^{(0)} \) stand for an initial probability vector, and \( P^{(k)} \) for the probability vector at iteration \( k \). Note that \( P^{(0)} \) is randomly initialized with the constraints of \( \text{(24)} \). At iteration \( k \), the rows of inequality matrix \( A \) are broken into two groups: one is active, the other is inactive. The active group is composed of all inequalities that must work exactly as an equality at \( P^{(k)} \), whereas the inactive group is composed of the left inequalities. If \( A_1^{(k)} \) and \( A_2^{(k)} \) respectively denote the active group and the inactive group, we have

\[
A_1^{(k)} P^{(k)} = 0
\]

\[
A_2^{(k)} P^{(k)} > 0
\]

Combining \( A_1^{(k)} \) with \( E \), we define an active matrix \( N^{(k)} \), namely,

\[
N^{(k)} = \begin{bmatrix} A_1^{(k)} \\ E \end{bmatrix}
\]

which determines the active constraints at iteration \( k \).

When \( N = N^{(k)} \) is not a square matrix, we construct two matrices \( G^{(k)} \) and \( Q^{(k)} \) as follows,

\[
G^{(k)} = N^T (NN^T)^{-1} N
\]

\[
Q^{(k)} = I_{LK \times LK} - G^{(k)}
\]

where \( G^{(k)} \) is called projection matrix, and \( Q^{(k)} \) is its orthogonal projection matrix \( \text{(46)} \).

At iteration \( k \), projecting the gradient of PKM’s objective function on the subspace of the active constraints, we obtain the projected gradient,

\[
d^{(k)} = -Q^{(k)} \nabla J (P^{(k)})
\]

If \( d^{(k)} = 0 \), we stop at a local minimum. Otherwise, we compute the probability vector at iteration \( k + 1 \),

\[
P^{(k+1)} = P^{(k)} + td^{(k)}
\]

where \( t \) is a step length. Usually, \( t \) takes a small value, e.g. \( t = 0.01 \) or \( t = 0.1 \).

When \( N = N^{(k)} \) is a square matrix, \( N \) must be invertible. Meanwhile, the projection matrix \( G^{(k)} = N^T (NN^T)^{-1} N = I_{LK \times LK} \), and \( Q^{(k)} = 0 \). Thus, the projected gradient \( d^{(k)} = -Q^{(k)} \nabla J (P^{(k)}) = 0 \), which cannot be chosen as a descent direction. Based on \( \text{(47)} \), we find the descent direction as follows.

1) Compute a new vector,

\[
q^{(k)} = (NN^T)^{-1} N \nabla J = (N^T)^{-1} \nabla J
\]

2) Break \( q^{(k)} \) into two parts \( q_1^{(k)} \) and \( q_2^{(k)} \), namely,

\[
q^{(k)} = \begin{bmatrix} q_1^{(k)} \\ q_2^{(k)} \end{bmatrix} = \begin{bmatrix} (q_1^{(k)})^T \\ (q_2^{(k)})^T \end{bmatrix}^T
\]

where the size of \( q_1^{(k)} \) is the number of rows of \( A_1^{(k)} \), and that of \( q_2^{(k)} \) is the number of rows of \( E \).

3) If all elements of \( q_1^{(k)} \) are greater than or equal to 0, i.e. \( q_1^{(k)} \geq 0 \), then stop. Otherwise, choose any one that is less than 0, delete the corresponding row of \( A_1^{(k)} \), and use \( \text{(27)} - \text{(30)} \) to compute \( d^{(k)} \).
Thus, matrices. Hence, (35) holds. 

3.3 The MSAGP Method

In the AGP method, the step length is manually chosen as a small number, leading to slow convergence. To address this issue, we present the MSAGP method by iteratively estimating the maximum step length in the feasible region. At iteration $k$, because $A_1^{(k)}P^{(k)} = 0$ and $A_2^{(k)}P^{(k)} > 0$, we can estimate the maximum step length 

$$t(k) = \frac{p(k)}{d(k)}$$

as follows.

1) Let $p_{ij}^{(k+1)} = p_{ij}^{(k)} + t_{ij}d_{ij}^{(k)} = 0$ for $p_{ij}^{(k)} > 0$,

2) Compute $t_{ij} = \frac{-p_{ij}^{(k)}}{d_{ij}}$ for $p_{ij}^{(k)} > 0$ and $d_{ij}^{(k)} < 0$,

3) $t_{max}^{(k)} = \min \{t_{ij}\}$.

In Fig. 2 we can see that step length $t(k)$ for AGP is shorter than $t_{max}^{(k)}$ for MSAGP at iteration $k$. Thus, MSAGP is expected to converge faster than AGP because it always finds a boundary point $P^{(k+1)}$ to update probability vector $P^{(k)}$. The faster convergence will be confirmed in Section 4.

3.4 The FMSAGP Method

Although MSAGP converges faster than AGP, it has to calculate a large number of matrix inversions. In fact, according to (28), the calculation of $G^{(k)} = N^T(NN^T)^{-1}N$ iteratively requires $(NN^T)^{-1}$ with $N = N^{(k)}$. To calculate $G^{(k)}$ more efficiently, it needs the following lemma.

**Lemma.** Suppose that $n$ is a row vector and $[n^T, N]^T$ is full row rank. If projection matrix $G_N = N^T(NN^T)^{-1}N$ and its orthogonal projection matrix $Q_N = I - G_N$, then

$$G(n^T, N)^T = G_N + Q_Nn^T(Q_Nn^T, Q_Nn^T)^{-1}nQ_N.$$  

(35)

**Proof.** Because $[n^T, N]^T$ is full row rank, by matrix theory [49 - 51], there must exist a matrix $G$ such that

$$
\begin{align*}
G(n^T, N)^T &= G + G_N, \\
GG_N &= G_NG = 0
\end{align*}
$$

(36)

Meanwhile, there must exist a matrix $Z$, such that

$$G = Z^Tn^T(Z^Tn^T, Z^Tn^T)^{-1}nZ$$

(37)

Combining (36) with (37), we obtain

$$Z^Tn^T(Z^Tn^T, Z^Tn^T)^{-1}nZG_N = 0$$

(38)

Thus, $nZG_N = 0$ must hold for all $n$. This leads to $ZG_N = 0$. That is to say, $Z$ is the orthogonal projection matrix of $G_N$, $Z = I - G_N = Q_N$. Moreover, both $G_N$ and $Q_N$ are symmetric matrices. Hence, (35) holds.

Using the above lemma, we can devise the FMSAGP method as follows,

1) Let $k = 0$, $N = N^{(0)} = E$, $G^{(0)} = N^T(NN^T)^{-1}N$, initialize $P^{(0)}$, go to 4;

2) Find an active row vector $n$ from $A_2^{(k-1)}$, such that $n^TP^{(k-1)}>0$, $n^TP^{(k)} = 0$;

3) Compute $G^{(k)}$ as

$$G^{(k)} = G^{(k-1)} + Q^{(k-1)}n^T(Q^{(k-1)}n^T, Q^{(k-1)}n^T)^{-1}nQ^{(k-1)}$$

(39)

4) Compute $Q^{(k)} = I - G^{(k)}$, and $d^{(k)} = -Q^{(k)}n$.

5) Compute step length $t^{(k)}$ by (34) and probability vector $P^{(k+1)}$ by (31);

6) Let $k = k + 1$, if $G^{(k)} \neq I_LK\times LK$, go to 2;

7) Construct $N = N^{(k)}$ by (27), and compute a new vector $q^{(k)} = (N^T)^{-1}n$ by (32);

8) Break $q^{(k)}$ into two parts $q_1^{(k)}$ and $q_2^{(k)}$ by (33);

9) If all elements of $q_1^{(k)}$ are greater than or equal to 0, stop;

10) Choose any element less than 0 from $q^{(k)}$, and delete the corresponding row of $A_1^{(k)}$;

11) Reconstruct $N = N^{(k)}$ by (27);

12) Compute $d^{(k)}$ by (28)-(30), $t^{(k)}$ by (34), and $P^{(k+1)}$ by (31);

13) Let $k = k + 1$, and go to 7).

4 EXPERIMENTS

In order to evaluate the performance of PKM and how well the methods of AGP, MSAGP, and FMSAGP solve it, we conduct a large number of experiments on fifteen datasets: one artificial dataset, one human face dataset, and thirteen UCI datasets. These datasets are detailed in Table 1.

| Artificial dataset | Class | Dimension |
|-------------------|------|-----------|
| Iris              | 150  | 3         |
| Parkinson         | 195  | 2         |
| Seeds             | 210  | 3         |
| Segmentations     | 210  | 7         |
| Glass             | 214  | 6         |
| Ionosphere        | 351  | 2         |
| Dermatologe       | 358  | 6         |
| Breast-cancer     | 683  | 2         |
| Natural           | 2000 | 9         |
| Yeast             | 2426 | 3         |
| Waveform          | 5000 | 3         |
| Satelltoe         | 6435 | 6         |
| Epileptic         | 11500| 5         |
| UCI               |      |           |
| Glass             | 214  | 6         |
| Yeast             | 2426 | 3         |
| Waveform          | 5000 | 3         |
| Satelltoe         | 6435 | 6         |
| Epileptic         | 11500| 5         |

The artificial dataset is generated by ourselves. As shown in Fig. 3, it is composed of 310 data points, with 150, 150, 5 and 5 respectively from 4 classes. The human face dataset is Yale-faces-B[1] which consists of 5850 images scaled down to 32x32 pixels.

The datasets are detailed in Table 1.

1. https://cervisia.org/machine_learning_data.php
4.1 Initialization Robustness

In this subsection, we evaluate initialization robustness of PKM, KM++ and FCM on the artificial dataset, with the result given in Table 2. The initialization robustness is defined as the number of correct times out of running an algorithm 1000 times initialized randomly. The correctness means the cluster assignation of a dataset is completely consistent with its original distribution. With the artificial dataset in Fig. 3, three inconsistent distributions are displayed as examples in Fig. 4. From Table 2, we can see that PKM have 954 correct times, more than 645 by KM++. However, FCM may have number of correct times that varies with parameter \( m \). For example, it has 687, 693, 780 and 691 correct times at \( m = 1.09, 1.1, 1.3 \) and 1.4, respectively. However, it has no correct times at \( m = 1.5, 1.09, 2.0 \) and 2.1. At \( m = 1.3 \), FCM achieves the maximum number of correct times, namely 780, still lower than PKM.

Overall, PKM has a correct percentage of 95.4%, better than KM++ (64.5%) and FCM (0 to 78.0%) in terms of initialization robustness.

4.2 Clustering Performance

In this subsection, we evaluate clustering performance of PKM, KM++ and FCM in terms of five measures: Standard Squared Error (SSE), Davies Bouldin Index (DBI) \([52]\), Normalized Mutual Information (NMI) \([53]\), Adjusted Rand Index (ARI) \([53]\), and V-measure (VM) \([54]\). The SSE is defined as

\[
SSE = \sum_{i=1}^{K} \sum_{x_i \in \omega_j} \| x_i - c_j \|^2
\]

(40)

where \( K \) is the number of clusters, and \( c_j \) is the center (mean) of the \( j \)-th cluster. DBI is the ratio of within-class distances to between-class distances. NMI is a normalization of the mutual information score to scale the results between 0 (no mutual information) and 1 (perfect correlation). ARI is an adjusted similarity measure between two clusters. V-measure is a harmonic mean between homogeneity and completeness, where homogeneity means that each cluster is a subset of a single class, and completeness means that each class is a subset of a single cluster.

The above five measures fall into two types: internal evaluation and external evaluation. Internal evaluation, including SSE and DBI, can work without ground truth labels. But external evaluation, such as NMI, ARI and VM, must work with ground truth labels. The lower the internal evaluation, the better the clustering result. However, the higher the external evaluation, the better the clustering result.

We test PKM (solved by FMSAGP), KM++ and FCM \((m = 1.3)\) on Yale-faces-B and 13 UCI datasets, with the 5-time average results reported in Table 3. The best in each
TABLE 3
Internal Evaluation of PKM, KM++ and FCM \((m = 1.3)\).

| Datasets          | SSE     | DBI     |
|-------------------|---------|---------|
|                   | PKM     | KM++    | FCM     | PKM     | KM++    | FCM     |
| Yale-faces-B      | 8.51e9  | 8.52e9  | 8.42e9  | 1.6184  | 1.6128  | 1.5986  |
| Iris              | 78.942  | 110.89  | 110.9   | 0.3928  | 0.4801  | 0.4561  |
| Seeds             | 587.05  | 588.05  | 587.32  | 0.3980  | 0.4421  | 0.3962  |
| Breast-cancer     | 2419.3  | 2419.3  | 2419.3  | 0.3785  | 0.3786  | 0.3786  |
| Dermatology       | 11702   | 11702   | 11702   | 0.7180  | 0.7939  | 0.7609  |
| Ionosphere        | 2419.4  | 2419.4  | 2419.4  | 0.7567  | 0.7567  | 0.7579  |
| Parkinson         | 1.34e6  | 1.34e6  | 1.34e6  | 0.4932  | 0.4974  | 0.4974  |
| Glass             | 377.27  | 379.41  | 366.35  | 0.8505  | 0.7074  | 0.7909  |
| Natural           | 1888.3  | 1883.5  | 1879.2  | 1.7035  | 1.6587  | 1.6089  |
| Satellite         | 1.67e7  | 1.69e7  | 1.69e7  | 0.8524  | 0.8731  | 0.8957  |
| Yeast             | 713.89  | 713.88  | 714.07  | 1.1601  | 1.1582  | 1.1753  |
| Segmentation      | 1.01e6  | 9.85e5  | 1.99e6  | 0.7461  | 0.7530  | 1.0899  |
| Waveform          | 1.33e5  | 1.33e5  | 1.33e5  | 0.8663  | 0.9301  | 0.8665  |
| Epileptic         | 5.1e10  | 5.1e10  | 5.3e10  | 2.6051  | 2.6956  | 3.6407  |

case is highlighted in bold. Some big values are in scientific notation. For instance, 8.42e9 means \(8.42 \times 10^9\).

From Table 3, we have the following observations.

1) By SSE, PKM performs better than KM++ on 5 datasets, and as well as KM++ on 6 datasets. Meanwhile, it performs better than FCM on 6 datasets, and as well as FCM on 5 datasets.

2) By DBI, PKM performs better than KM++ on 9 datasets, and as well as KM++ on 1 datasets. Meanwhile, it performs better than FCM on 10 datasets.

From Table 4, we have the following observations.

1) By NMI, PKM performs better than KM++ on 5 datasets, and as well as KM++ on 5 datasets. Meanwhile, it outperforms FCM on 12 datasets.

2) By ARI, PKM perform better than KM++ on 5 datasets, and as well as KM++ on 4 datasets. Meanwhile, it outperforms FCM on 11 datasets.

3) By VM, PKM perform better than KM++ on 9 datasets, and as well as KM++ on 2 datasets. Meanwhile, it outperforms FCM on 11 datasets.

Overall, PKM outperforms KM++ and FCM in terms of SSE, DBI, NMI, ARI and VM.

4.3 Descent Stability of MSAGP

In this subsection, we evaluate the descent stability of MSAGP. Theoretically, AGP needs a small step length to make the value of its objective function stably descend at each iteration and gradually converge to a local minimum. If the step length is too large, the value may descend with oscillation, and even does not converge at all. MSAGP iteratively estimates a maximum step length for AGP in the feasible region of PKM so as to speed up its convergence with fewer iterations. However, will this estimate have any serious influence of oscillation on the convergence of AGP?

To demonstrate the influence, we select 9 UCI datasets from Table 1, including Satellite, Yale, Epileptic, Natural, Yeast, Waveform, Glass, Dermatology, and Breast-cancer. On these datasets, we use MSAGP to solve PKM, and illustrate the value of its objective function vs. iteration in Fig. 5. Obviously, we can see that MSAGP descends stably without oscillation.

![Figure 5: Descent stability of MSAGP on 9 UCI datasets](image)

(a) (b) (c) (d) (e) (f) (g) (h) (i)

4.4 Iteration Number of MSAGP and AGP

In this subsection, we compare iteration number for MSAGP and AGP \((t = 0.01 \text{ and } 0.1)\) to converge on 8 UCI datasets, including Segmentation, Parkinson, Dermatology, Breast-cancer, Seeds, Ionosphere, Glass and Iris. The results are illustrated in Figs. 6-7 and reported in Table 5, where we have the following observations.
### TABLE 4
External Evaluation of PKM, KM++ and FCM (m = 1.3).

| Datasets       | NMI   | ARI   | VM   |
|----------------|-------|-------|------|
|                | PKM   | KM++  | FCM  | PKM   | KM++  | FCM  | PKM   | KM++  | FCM  |
| Yale-faces-B   | 0.7959| 0.7952| 0.7722| 0.6631| 0.6123| 0.6386| 0.8264| 0.1744| 0.0059|
| Iris           | 0.7501| 0.6733| 0.6723| 0.7233| 0.5779| 0.5763| 0.7419| 0.7149| 0.7081|
| Seeds          | 0.7025| 0.7025| 0.6949| 0.7135| 0.7135| 0.7166| 0.7050| 0.6999| 0.6949|
| Breast-cancer  | 0.7546| 0.7478| 0.7478| 0.8521| 0.8464| 0.8464| 0.7545| 0.7478| 0.7478|
| Dermatology    | 0.1141| 0.1032| 0.1046| 0.0391| 0.0266| 0.0261| 0.1140| 0.1056| 0.1095|
| Ionosphere     | 0.1349| 0.1349| 0.1349| 0.0391| 0.0266| 0.0261| 0.1140| 0.1056| 0.1095|
| Parkinson      | 0.1124| 0.1217| 0.1217| 0.1876| 0.2046| 0.2046| 0.1153| 0.0607| 0.1262|
| Glass          | 0.3294| 0.4178| 0.3489| 0.2201| 0.2551| 0.2126| 0.3871| 0.3857| 0.3807|
| Natural        | 0.0523| 0.0536| 0.0521| 0.0247| 0.0261| 0.0273| 0.0518| 0.0529| 0.0495|
| Satellite      | 0.6117| 0.6117| 0.6117| 0.5292| 0.5293| 0.4289| 0.5790| 0.5544| 0.6128|
| Yeast          | 0.0050| 0.0050| 0.0050| 0.0117| 0.0117| 0.0109| 0.0048| 0.0045| 0.0045|
| Segmentation   | 0.5338| 0.5132| 0.4678| 0.3823| 0.3313| 0.3172| 0.4996| 0.5252| 0.5729|
| Waveform       | 0.3622| 0.3622| 0.3622| 0.2536| 0.2536| 0.2529| 0.3622| 0.3622| 0.3559|
| Epileptic      | 0.1228| 0.1675| 0.1033| 0.0245| 0.0263| 0.0225| 0.1226| 0.1291| 0.0058|

### TABLE 5
Iteration number for MSAGP and AGP to converge with objective function value on 8 UCI datasets.

| Datasets    | Iteration number | Objective function value |
|-------------|------------------|--------------------------|
|             | MSAGP            | AGP (t = 0.01)           | AGP (t = 0.1) |
| Segmentation| 1290             | 1338                     | 1291         | 986167.4343 | 986167.4343 | 986167.4343 |
| Parkinson   | 195              | 196                      | 196          | 134335.02   | 134335.02   | 134335.02   |
| Dermatology | 1801             | 6329                     | 2247         | 11715.1043  | 11715.1043  | 11715.1043  |
| Breast-cancer| 452              | 577                      | 462          | 19323.2     | 19323.2     | 19323.2     |
| Seeds       | 421              | 634                      | 534          | 587.32      | 587.32      | 587.32      |
| Glass       | 1079             | 190683                   | 115145       | 378.9       | 377.19      | 381.74      |
| Ionosphere  | 368              | 2070                     | 524          | 2419.3      | 2419.3      | 2419.3      |
| Iris        | 308              | 53022                    | 5873         | 78.95       | 78.95       | 78.95       |

1) MSAGP requires fewer iterations to converge than AGP on the eight datasets, especially much fewer on Dermatology (Fig. 6c), Ionosphere (Fig. 6f), Glass (Figs. 7a-c) and Iris (Figs. 7d-f).

2) When converging, MSAGP and AGP have the same objective function value on Parkinson, Breast-cancer, Seeds, Ionosphere and Iris. However, MSAGP has an objective function value slightly greater than AGP (t = 0.01) on Segmentation, Dermatology and Glass, while slightly smaller than AGP (t = 0.1) on Glass.

Overall, MSAGP can reach a competitive convergence with AGP in fewer iterations.

### 4.5 Convergence Speed of FMSAGP and MSAGP

In this subsection, we compare convergence speed of FMSAGP and MSAGP in running time on 9 UCI datasets: Parkinson, Iris, Seeds, Glass, Segmentation, Breast-cancer, Dermatology, Yeast and Waveform. The results are reported in Table 6, where we have the following observations.

1) FMSAGP runs faster than MSAGP on all the nine datasets.

2) The larger the dataset, the lower the speed-up (η) of FMSAGP to MSAGP. For example, the speed-up is 47.3% on Iris (150 samples), 22.3% on Glass (214 samples), 9.7% on Yeast (2426 samples) and 1.5% on Waveform (5000 samples).

![Fig. 6. Iteration number (in x-axis) for MSAGP and AGP (t = 0.01 and 0.1) to converge on 6 UCI datasets: Segmentation (a), Parkinson (b), Dermatology (c), Breast-cancer (d), Seeds (e) and Ionosphere (f). Y-axis means objective function value of PKM.](image)

### 5 Conclusion

In this paper, the most important contribution is our solution to the clustering problem of FCM at m = 1 by a
new model, i.e. PKM. Moreover, we have addressed the PKM model by three methods: AGP, MSAGP and FM-SAGP. Additionally, we have conducted a large number of experiments to evaluate how well these methods work in initialization robustness, clustering performance, descent stability, iteration number, and convergence speed. As future work, we will further improve our solution to PKM, and apply the basic idea to other relevant problems, such as $L_p$-norm K-means, kernel PKM, and even possibilistic c-means. Particularly, we will combine deep neural networks with PKM to build deep PKM models, and to develop more nonlinear programming models for machine learning.

**ACKNOWLEDGMENTS**

This work was supported by the National Natural Science Foundation of China under grant numbers: 61876010 and 61806013.

**REFERENCES**

[1] Purkait P, Chin T J, Sadri A, et al. Clustering with Hypergraphs: The Case for Large Hyperedges[1]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(9): 1697-1711.

[2] Otto C, Wang D, Jain A. Clustering Millions of Faces by Identity[2]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(2):289-303.

[3] Kim S, Yoo C D, Nowozin S, et al. Image Segmentation Using Higher-Order Correlation Clustering[3]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(9):1761-1774.

[4] Marin D, Tang M, Ayed I B, et al. Kernel Clustering: Density Biases and Solutions[4]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(1):136-147.

[5] Heymann J, Drude L, Haeberl-Umbach R. A Generic Neural Acoustic Beamforming Architecture for Robust Multi-Channel Speech Processing[5]. Computer Speech and Language, 2017, 46: 374-385.

[6] Ronan T, Qi Z, Naegel K M. Avoiding Common Pitfalls When Clustering Biological Data[6]. Science Signaling, 2016, 9(e432):r6.

[7] Taylor M J, Husain K, Gartner Z J, et al. A DNA-Based T Cell Receptor Reveals a Role for Receptor Clustering in Ligand Discrimination[7]. Cell, 2017, 169(1):108-119.

[8] Liu A A, Su Y T, Nie W Z, et al. Hierarchical Clustering Multi-Task Learning for Joint Human Action Grouping and Recognition[8]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 39(1):102-114.

[9] Li Y J. A Clustering Algorithm Based on Maximal $\theta$-Distant Subtrees. Pattern Recognition. 2007, 40(5):1425-1431.

[10] Frey B J, Dueck D. Clustering by Passing Messages Between Data Points[10]. Science, 2007, 315(5814):972-976.

[11] MacQueen J. Some Methods for Classification and Analysis of Multivariate Observations[C]. Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967:281-297.

[12] Kaufmann L, Rousseweu P J. Clustering by Means of Medoids[12]. Statistical Data Analysis Based on the L1-norm and Related Methods. North-Holland, 1987:405-416.

[13] Bezdek J C. Pattern Recognition with Fuzzy Objective Function Algorithms[M]. Plenum Press, 1981.

[14] Guha S, Rastogi R, Shim K, et al. CURE: An Efficient Clustering Algorithm for Large Databases[14]. Information Systems, 1998, 26(1):35-58.

[15] Madan D, Dana K J. Modified Balanced Iterative Reducing and Clustering Using Hierarchies (m-BIRCH) for Visual Clustering[15]. Pattern Analysis and Applications, 2016, 19(4):1023-1040.

[16] Johnson T, Singh S K. Divisive Hierarchical Bisecting MinCMax Clustering Algorithm[16]. Proceedings of the International Congress on Data Engineering and Communication Technology, 2017.

[17] Ester M, Kriegel H P, Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise[17]. International Conference on Knowledge Discovery and Data Mining, 1996:226-231.

[18] Rodriguez A, Laio A. Clustering by Fast Search and Find of Density Peaks[18]. Science, 2014, 344(6191):1492-1496.

[19] Szlam A, Bresson X. Total Variation and Cheeger Cuts[C]. International Conference on Machine Learning, 2010: 1039-1046.

[20] Elhamifar E, Vidal R. Sparse Subspace Clustering[19]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(7):1358-1371.

[21] Ester M, Kriegel H P, Xu X. A Density-Based Algorithm for High-Dimensional Data[J]. International Journal of Intelligent Systems, 1996:226-231.

[22] Marin D, Tang M, Ayed I B, et al. Hierarchical Clustering Multi-Task Learning for Joint Human Action Grouping and Recognition[22]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 39(1):102-114.

[23] Kadir S N, Goodman D F, Harris K D. High-Dimensional Cluster Analysis with The Masked Em Algorithm[J]. Neural Computation, 2014, 26(11):2379-2394.

[24] Elamifar E, Vidal R. Sparse Subspace Clustering[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2009:2790-2797.
