11th International Conference on Intelligent Control and Information Processing
Dali, China, December 3-7, 2021

Fast Particle Swarm Optimization for Balanced Clustering

1st Meng Zhang
Key Laboratory of Intelligent Information
Processing and Control of
Chongqing Municipal Institutions of Higher Education,
Chongqing Three Gorges University,
Chongqing, 40044 China
mengzhang2000116@163.com

2nd Yao Xiao
Key Laboratory of Intelligent Information
Processing and Control of
Chongqing Municipal Institutions of Higher Education,
Chongqing Three Gorges University,
Chongqing, 40044 China
yaoxiao227192@gmail.com

3rd Xiaoling Song*
Key Laboratory of Intelligent Information
Processing and Control of
Chongqing Municipal Institutions of Higher Education,
Chongqing Three Gorges University,
Chongqing, 40044 China
20191084@sanxiau.edu.cn

4th Xiangguang Dai
Key Laboratory of Intelligent Information
Processing and Control of
Chongqing Municipal Institutions of Higher Education,
Chongqing Three Gorges University,
Chongqing, 40044 China
daixiangguang@163.com

5th Nian Zhang
Department of Electrical and Computer Engineering
University of the District of Columbia,
Washington, D.C., 20008 USA
nzhang@udc.edu

2021 11th International Conference on Intelligent Control and Information Processing (ICICIP) | 978-1-6654-2515-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICICIP53388.2021.9642162

Abstract—There are balanced priorities in various engineering fields (e.g. medicine, statistics, artificial intelligence, and economics, etc.). Some clustering algorithms cannot maintain the natural balanced structure of data. This paper proposes a soft-balanced clustering framework, which can achieve a balanced clustering for each cluster. The model can be formulated d as a mixed-integer optimization problem. We transform the problem into several subproblems and utilize PSO to search the global solution. Experiments show that the proposed algorithm can achieve satisfactory clustering results than other clustering algorithms.

Index Terms—Clustering, Balanced clustering, Particle Swarm Optimization, Matrix Factorization

I. Introduction

Clustering [1] is an important research topic of machine learning, which has a wide range of applications in computer vision, image segmentation and salience or target detection [2]–[5]. The clustering algorithms aim to divide the unlabeled dataset into groups which consist of similar data points [6], such as K-means [7], Graph clustering (GC) [8] and Spectral clustering (SC) [9], [10]. Unfortunately, these algorithms may lead to poor solutions when each cluster comprises few data points. Therefore, some researchers proposed generative or discriminative algorithms [10]–[14]. Generative algorithms [11], [12] combine prior knowledge of data distribution and clustering model. Discriminative algorithms [10], [13], [14] directly parameterize the clustering boundary, so as to achieve more clear clustering. Malinen et al. [15] imposed a constraint of the cluster size into K-means, which leads to the equivalent cluster size for each cluster. Bradley et al. [16] proposed k constraints for the clustering problem, and each cluster should contain a minimum number of data points at least.

In the real-world, most datasets contain the property of the balanced structure. In other words, each category may contain an equal number of data points. For these datasets, we want to design an algorithm and reflect the balance structure of the clustering results. Due to the different optimization goals, balanced clustering algorithms can be divided into hard-balanced and soft-balanced algorithms [15]–[19]. Hard-balanced approaches [15], [16] consider the cluster size to be the main goal of the clustering algorithm. Soft-balanced algorithms [17]–[19] aim to reduce the clustering errors. Recently, Han et al. [20] introduced a method called global balanced

This work is supported by Foundation of Chongqing Municipal Key Laboratory of Institutions of Higher Education ([2017]3), Foundation of Chongqing Development and Reform Commission (2017[1007]), Scientific and Technological Research Program of Chongqing Municipal Education Commission (Grant Nos. KJQN201901218 and KJQN201901203), Natural Science Foundation of Chongqing (Grant No. cstc2019jcyj-bshX0101), Foundation of Chongqing Three Gorges University and National Science Foundation (NSF) grant #2011927 and DoD grant #W911NF1810475.
*Corresponding author: Xiaoling Song

clustering (GBC), which can adjust the balance degree by parameters. GBC can capture both local and global data, but it converges slowly and tends to fall into the local minima. Recently, the particle swarm optimization (PSO) [21], [22] was proposed as a genetic algorithm. Compared with other genetic algorithms, PSO is enriched with the features of the simple concept, easy implementation, and computational efficiency.

Combining GBC and PSO, we propose a new balanced clustering method that can quickly converge and try the best to find the global optimal solution. In this article, we propose Fast Particle Swarm Optimization for Balanced Clustering (FPSOBC), a fast balanced clustering method that regresses each unknown clustering label to the corresponding data sample. In other words, we decompose the label matrix into the original high-dimensional data matrix and the weighted matrix. Moreover, we try our best to reduce the error between the label matrix and the product of the decomposed matrices. To achieve the goal of balanced clustering, we imposed the balance constraint into the label matrix. In summary, the main contributions of our work include the following aspects.

- A soft-balanced clustering framework is proposed to achieve balanced clustering solution and reduce the clustering error.
- The framework can be formulated as a mixed-integer optimization problem (MIOP). To solve the MIOP effectively, the MIOP is divided into a convex problem and an integer optimization problem. We alternatively solve the two subproblems and change the solution by PSO until a global solution is achieved.
- Extensive experiments and evaluations on several datasets show that our proposed method achieves superior performance in comparison to other state-of-the-art methods.

II. Related Works

A. Augmented Lagrange Multipliers

Augmented Lagrangian methods [24] are usually utilized to solve optimization problems with equality constraints. For a convex optimization problem with linear equality constraints, we can rewrite it into the following form:

$$\begin{aligned} \min \quad & F(X) \\ s.t. \quad & h(X) = 0, \end{aligned} \tag{1}$$

The Lagrange Multiplier method for problem (1) is defined by

$$\mathcal{L}(X, \mu) = F(X) + \mu h(X). \tag{2}$$

For problem (2), we rewrite it by the Augmented Lagrange Multipliers (ALM) as follows:

$$\mathcal{L}(X, \Lambda, \mu) = F(X) + \Lambda h(X) + \frac{\mu}{2}\|h(X)\|_F^2, \tag{3}$$

where Λ is Lagrange multiplier and μ is the step size. The penalized Lagrangian method is used to transform a constrained problem into an unconstrained problem. The augmented Lagrange multiplier method can be shown in Algorithm 1.

Algorithm 1 Augmented Lagrange Multipliers (ALM)

Input: $\rho > 0$
Output: X
Initializing μ, and set $\Lambda = 0$
while not converge do
  1) Solve $\mathbf{X}^{(t+1)} = \arg\min_{\mathbf{X}} \mathcal{L}\left(\mathbf{X}^{(t)}, \Lambda^{(t)}, \mu^{(t)}\right)$ in Eq. (3);
  2) Update $\Lambda : \Lambda^{(t+1)} = \Lambda^{(t)} + \mu^{(t)} I\left(\mathbf{X}^{(t+1)}\right)$
  3) Update $\mu : \mu^{(t+1)} = \rho\mu^{(t)}$.
end while
Return X

B. Particle Swarm Optimization

The particle swarm optimization simulates the bird in the bird swarm by designing a kind of massless particle. The particle has only two attributes: speed and position. Speed and position denote the speed and direction of movement, respectively. Each particle searches the optimal solution separately in the search space, and records it as the current individual extremum, and shares the individual extremum with other particles in the whole particle swarm to find the optimal individual extremum as the current global optimal solution of the whole particle swarm. All particles in the particle swarm adjust their speed and location according to the current individual extremum they find and the current global optimal solution shared by the whole particle swarm.

Firstly, each particle searches the optimal solution separately in the searching space, and records it as the current individual extremum.

$$pBest_i^{\text{new}} = \begin{cases} pBest_i^{\text{old}} & ,\text{if } fit(pBest_i^{\text{old}}) < fit(x_i) \\ x_i & ,\text{otherwise,} \end{cases} \tag{4}$$

where $pBest_i$ is the historical optimal position of the particle i, and $fit(\cdot)$ is an evaluation function.

Secondly, the global optimal individual extremum is found according to the individual optimal extremum of each particle

$$gBest = min\{pBest_1, pBest_2, \cdots, pBest_i\}, \tag{5}$$

where $gBest$ is the global optimal location.

Thirdly, each particle adjusts the speed and location according to the current individual extremum and the current global optimal solution.

$$v_i = v_i + c_i \times rand() \times (pBest_i - x_i) + c_2 \times rand() \times (gBest_i - x_i), \tag{6}$$

and

$$x_i = x_i + v_i, \tag{7}$$

where v_i is the speed of particles i, $rand(\cdot)$ is a random function which can generate a random number between

0 and 1, x_i is the current position of the particle i and c_1 and c_2 are the learning factors. The particle swarm optimization can be shown as Algorithm 2.

Algorithm 2 Particle Swarm Optimization (PSO)

```
for each particle i do
    Initializing velocity v_i, and position x_i for each particle i
    Evaluate particle i and set pBest = x_i
end for
gBest = min{pBest_i}
while not stop do
    for i = 1 to N do
        Update velocity and position of particle i
        Evaluate particle i
        if fit(x_i)<fit(pBest_i) then
            pBest_i = x_i
        end if
        if fit(pBest_i)<fit(gBest) then
            gBest = pBest_i
        end if
    end for
end while
Return gBest
```

III. Fast Particle Swarm Optimization for Balanced Clustering

In this section, we propose Fast Particle Swarm Optimization for Balanced Clustering (FPSOBC). A fast balanced clustering method that regresses each unknown clustering label to the corresponding data sample. In order to achieve the goal of balanced clustering, we apply balance constraints to the label matrix.

A. Clustering optimization

Supposed that there are n samples for the dataset and each sample contains m-dimensional features. We hope to group the dataset denoted by $X = [x_1, \cdots, x_n] \in R^{n\times m}$ into c classes and find a label assignment matrix (indicator matrix) $B = [b_1, b_2, \cdots, b_n]^\top \in \{-1,1\}^{n\times c}$ for the clustering results. Thus, these assumptions can be mathematically denoted by

$$B \approx XW, \tag{8}$$

where W is a projection matrix, and B can be defined as

$$B_{ik} = \begin{cases} 1 & , x_i \in \text{cluster } k \\ -1 & , \text{otherwise.} \end{cases} \tag{9}$$

Generally, the Frobenius norm is utilized to measure the error between B and XW. Moreover, the L2-regularization of each projection vector is proposed to avoid the overfitting case. In summary, we can transform (8) into an optimization problem as follows:

$$\begin{aligned} \min_{W,B} \quad & \|B - XW\|_F^2 + \lambda \|W\|_F^2 \\ s.t. \quad & B \in \{-1,1\}^{n\times c}. \end{aligned} \tag{10}$$

The optimization problem can be solved by updating two variables alternatively.

B. Balanced Constraints

We supposed a vector $S = [s_1, s_2, \cdots, s_c]$ that the size of cluster i denoted by s_i. Thus, there are c balanced constraints, where $s_i = n/c, (i = 1, \cdots, c)$. We have

$$\begin{aligned} & \sum_{k=1}^{c} \left(s_k - \frac{n}{c}\right)^2 \\ =& \sum_{k=1}^{c} s_k^2 - 2\frac{n}{c}\sum_{k=1}^{c} s_k + \frac{n^2}{c^2} \\ =& \|S\|^2 - \frac{n^2}{c} \\ =& \|S\|^2 - const. \end{aligned} \tag{11}$$

Balance constraints can be described by the following problem:

$$\min_{S} \quad \|S\|_2^2. \tag{12}$$

It is obvious that S can be represented by the product of $B^\top$ and a column vector which entries are all equal to 1, we can transform S into the following form

$$S = B^\top 1. \tag{13}$$

By replacing S with (13), we can rewrite (12) as follows:

$$\min_{B} \quad tr(B^\top 1 1^\top B). \tag{14}$$

C. Problem Formulation

Combining (10) and (14) leads to our problem as follows:

$$\begin{aligned} \min_{B,W} \quad & \|B - XW\|_F^2 + \lambda\|W\|_F^2 + \beta tr(B^\top 1 1^\top B) \\ s.t. \quad & X \in R^{n\times m}, W \in R^{m\times c}, B \in \{-1,1\}^{n\times c} \end{aligned} \tag{15}$$

where α and β are hyper-parameters.

Normally, the objective function of problem (15) can be transformed into the following optimization problems:

$$\min_{W} \quad \|B - XW\|_F^2 + \lambda\|W\|_F^2 \tag{16}$$

and

$$\min_{B} \quad \|B - XW\|_F^2 + \beta tr(B^\top 1 1^\top B). \tag{17}$$

For problem (16), its solution can be

$$W = (X^\top X + \lambda I)^{-1} X^\top B = A X^\top B, \tag{18}$$

where $A = (X^\top X + \lambda I)^{-1}$. Substituting (18) into (15), we have

$$\min_{B} \quad \|B - XAX^\top B\|_F^2 + \lambda\|AX^\top B\|_F^2 + \beta tr(B^\top 1 1^\top B). \tag{19}$$

For problem (19), we can transform the first term as follows

$$\begin{aligned} & \|B - XAX^\top B\|_F^2 \\ =& tr((B - XAX^\top B)^\top (B - XAX^\top B)) \\ =& tr(B^\top I_n B - 2B^\top XA^\top X^\top B + B^\top XA^\top X^\top XAX^\top B), \end{aligned} \tag{20}$$

Similarly to the second term of (19), we have

$$\begin{aligned} &\|AX^\top B\|_F^2 \\ =&tr(B^\top XA^\top AX^\top B). \end{aligned} \tag{21}$$

In summary, problem (19) can be rewritten by

$$\begin{aligned} \min_B \quad & tr(B^\top MB) \\ s.t. \quad & B \in \{-1,1\}^{n\times c}. \end{aligned} \tag{22}$$

where $M = I_n - 2XA^\top X^\top + XA^\top X^\top XAX^\top + \lambda XA^\top AX^\top + \beta 11^\top$.

D. Optimization With ALM

Due to the discrete constraint, it is not easy to solve problem (22). Firstly, we propose an auxiliary matrix $Z \in R^{n\times c}$ to substitute B. Secondly, we utilize ALM and transform problem (22) as follows:

$$\mathcal{L}(Z,B,\mu,\Lambda) = tr(Z^\top MZ) + \frac{\mu}{2}\|B - Z + \frac{1}{\mu}\Lambda\|_F^2. \tag{23}$$

Finally, we mainly discuss how to solve problem (23). Problem (23) can be transformed into the following two optimization problems:

$$\min_Z \quad tr(Z^\top MZ) + \frac{\mu}{2}\|B - Z + \frac{1}{\mu}\Lambda\|_F^2 \tag{24}$$

and

$$\min_B \quad \frac{\mu}{2}\|B - Z + \frac{1}{\mu}\Lambda\|_F^2. \tag{25}$$

For problem (24), we have

$$Z = (\mu I_n + 2M)^{-1}(\mu B + \Lambda). \tag{26}$$

For problem (25), we transform it as follows:

$$\min_B \quad \|B - V\|_F^2, \tag{27}$$

where $V = Z - \frac{1}{\mu}\Lambda$ and

$$B_{ik} = \begin{cases} 1 & ,\text{if } k = argmax\{V_{ik}\}_{k=1}^c \\ -1 & ,\text{otherwise.} \end{cases} \tag{28}$$

Combining (18) and (28), we can alternately solve W and B until the local optimum solution searched [27], [28]. Our algorithm which is called Fast Balanced Clustering (FBC) can be shown as Algorithm 3.

Algorithm 3 Fast Balanced Clustering (FBC)

```
Input: X, c, λ and β
Output: B
  Initializing B ∈ {−1,1}^{n×c} and W ∈ R^{m×c}
  while not converge do
    Solve W by Eq.(18);
    while not converge do
      Solve Z by Eq.(26);
      Solve B by Eq.(28);
      Update Λ : Λ^(t+1) = Λ^(t) + μ^(t) (B − Z);
      Update μ : μ^(t+1) = ρμ^(t).
    end while
  end while
  Return B
```

E. Optimization With PSO

Since the optimization problem is a non-convex optimization problem, which makes our algorithm trapped in a local optima. To address this problem, we change the solution by PSO until a global solution is achieved. Our algorithm can be shown in Algorithm 4.

Algorithm 4 Fast Particle Swarm Optimization for Balanced Clustering (FPSOBC)

```
Input: X, c, λ and β
Output: B
  for each particle i do
    Initializing velocity V_i, and position B_i by FBC
  for each particle i
    Evaluate particle i and set pBest = B_i
  end for
  gBest = min{pBest_i}
  while not stop do
    for i = 1 to N do
      Update velocity and position of particle i
      Update position B_i by FBC
      Evaluate particle i
      if fit(X_i)<fit(pBest_i) then
        pBest_i = B_i
      end if
      if fit(pBest_i)<fit(gBest) then
        gBest = pBest_i
      end if
    end for
  end while
  B = gBest
  Return B
```

IV. Experimental examples

In this section, five methods (i.e. CKmeans, BKmeans, Kmeans, FBC and FPSOBC) are proposed to sustain the performance of clustering on the three different image datasets (i.e. Yale, Wine, Iris, Seeds) and we use Accuracy (ACC) and Normalized Mutual Information (NMI) to measure the clustering utility.

A. Compared Methods

We compare FBC and FPSOBC with classical balanced clustering algorithms Kmeans, CKmeans and non-balanced algorithm Kmeans.

- FBC: a soft-balanced clustering framework can achieve balanced clustering solution and reduce the clustering error, which we introduced in Section III-D.
- FPSOBC: a fast balanced clustering method that regresses each unknown clustering label to the corresponding data sample, which we introduced in Section III-E
- Kmeans:a traditional unsupervised clustering algorithm.

- CKmeans: a clustering algorithm with k constraints, and each cluster should contain a minimum number of data points at least.
- BKmeans: a K-mean method with constraint of the cluster size, which can lead to the equivalent cluster size for each cluster.

B. Compared Datasets

- Yale: The dataset contains 165 facial images, each measuring 32 × 32 pixels. The images are from 15 different people, each with 11 photos.
- Wine: The dataset contains 178 wine components, each wine has 13 different components. The wines can be divided into 3 categories. For balance clustering, we organize the dataset into balanced.
- Iris: The dataset contains 150 characteristics of iris flower, each iris flower has 4 characteristics. The iris flowers can be divided into 3 categories.
- Seeds: The dataset contains 210 geometrical properties of seeds, each seeds has 7 geometrical properties. The seeds can be divided into 3 categories.

For balanced clustering, we balanced unbalanced datasets in Table. I. In Fig. I, it shows that Properties of each dataset.

TABLE I
Properties of datasets

DATASET	Samples	Features	Classes
YALE	165	1024	15
Wine	144	13	3
Iris	150	4	3
Seeds	210	7	3

C. Clustering Evaluation Index

Clustering ACC can be written as

$$\mathrm{ACC}=\frac{\sum_{i=1}^{n}\delta\left(p_i,\operatorname{map}\left(q_i\right)\right)}{n}, \tag{29}$$

where $\delta(a,b)=1$ when $a=b$, or $\delta(a,b)=0$, function $\operatorname{map}(q_i)$ is a mapping function which can map clustering label q_i to ground truth labels, n is the sample number.

Clustering NMI can be written as

$$\mathrm{NMI}(\Omega,C)=\frac{I(\Omega;C)}{(H(\Omega)+H(C)/2)}, \tag{30}$$

where Mutual Information function $I()$ can be defined as follows:

$$\begin{aligned} I(\Omega;C)&=\sum_k\sum_j P\left(w_k\cap c_j\right)\log\frac{P\left(w_k\cap c_j\right)}{P\left(w_k\right)P\left(c_j\right)} \\ &=\sum_k\sum_j\frac{\left|w_k\cap c_j\right|}{N}\log\frac{N\left|w_k\cap c_j\right|}{\left|w_k\right|\left|c_j\right|}\end{aligned} \tag{31}$$

and entropy function $H()$ can be defined as follows:

$$\begin{aligned} H(\Omega)&=-\sum_k P\left(w_k\right)\log P\left(w_k\right) \\ &=-\sum_k\frac{\left|w_k\right|}{N}\log\frac{\left|w_k\right|}{N}\end{aligned} \tag{32}$$

where $P(w_k)$ and $P(c_j)$ can be seen as the probabilities of samples belonging to cluster w_k, category c_j, $P(w_k\cap c_j)$ is the joint probability that can be seen as belonging to both cluster w_k and category c_j.

Remarks: We can find codes of ACC and NMI on the homepage of Deng Cai (http://www.cad.zju.edu.cn/home/dengcai).

D. Parameter Settings

λ and β are two important parameters of the model (15) in this paper. It is necessary to select feasible parameters for clustering performance. Since the model is sensitive to penalty terms, they need to adjust carefully, but there is no theory and effective method to adjust λ and β. We get the best clustering results by exhaustively combining $\lambda\in\{10^{-2},10^{-1},1,10^{1},10^{2}\}$ and $\beta\in\{10^{-2},10^{-1},1,10^{1},10^{2}\}$. From Fig. 1 we can obtain: 1)The bigger λ and β bring the worse clustering performance. 2)For the Iris dataset, λ and β tend to 0 for better results. For the Wine and Yale datasets, λ and β tend to 1 for better results. For seeds dataset, λ and $\beta<0$ can always obtain a satisfactory result.

The ALM parameters are set as Tabel II.

TABLE II
parameter for ALM

μ	0.1 or 1
ρ	1.005
iters	50

In the optimization with PSO, we set the population to 50 and learning factors c1 and c2 to 2 for all datasets.

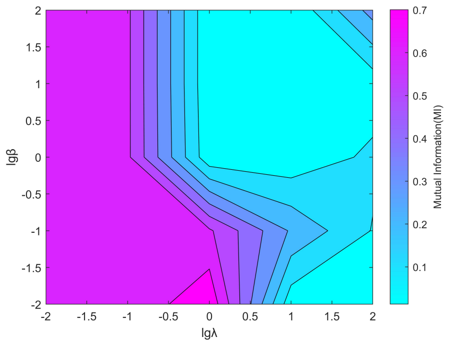
Since the variable optimization problem of this model is non-convex, we repeat each algorithm 10 times and present the minimum / minimum / average result.

E. Performance Comparison

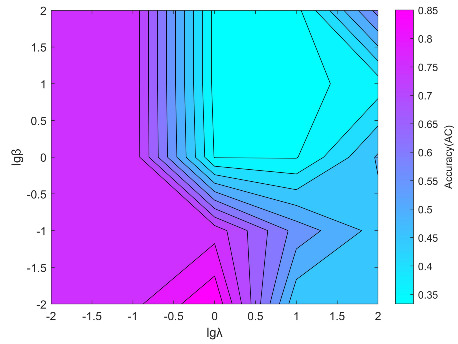
To show the effectiveness of our proposed method, we present the clustering performance in Table III, IV. In order to show the balanced performance of the clustering method, we take yale as an example. Fig. 3 shows the distribution of samples of each method in the cluster.

By analyzing the comparison results, we can draw the following conclusions.

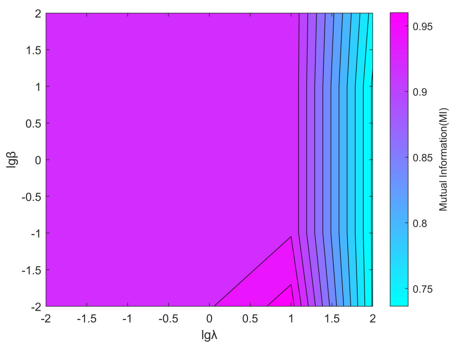
- Fig. 3 indicates that a soft-balanced clustering framework can well gain balanced clustering.
- It is obvious that the model using the ALM approach can converge on each dataset in Fig. 2. The oscillations in the figure are partly caused by the discrete label matrix B.



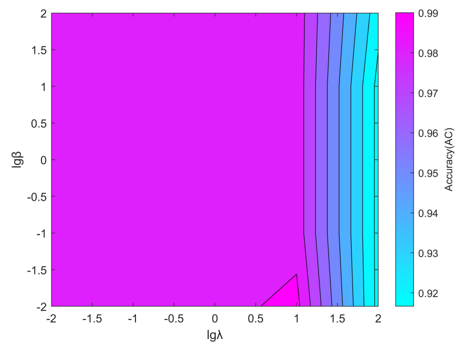
(a) Clustering MIs from Iris



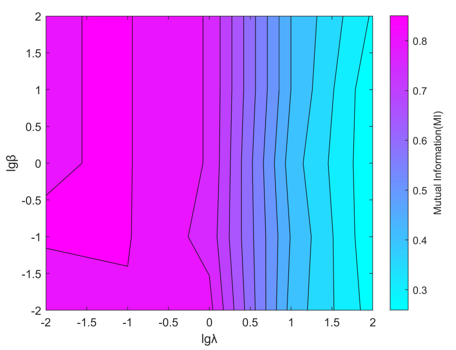
(b) Clustering ACs from Iris



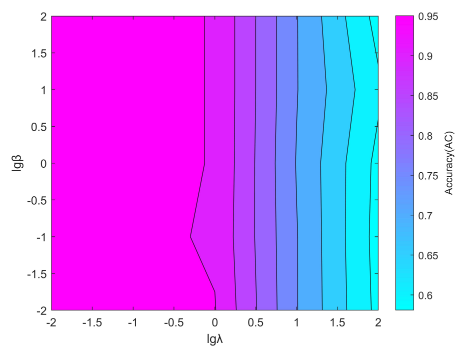
(c) Clustering MIs from Wine



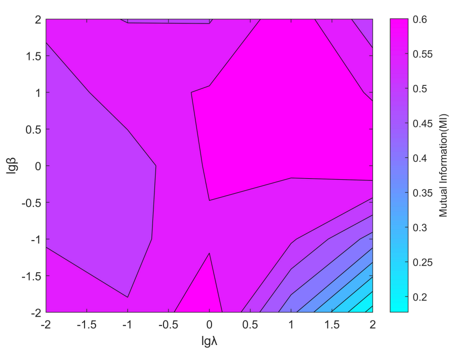
(d) Clustering ACs from Iris



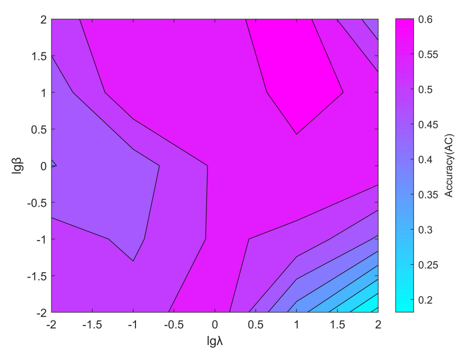
(e) Clustering MIs from Seeds



(f) Clustering ACs from Iris

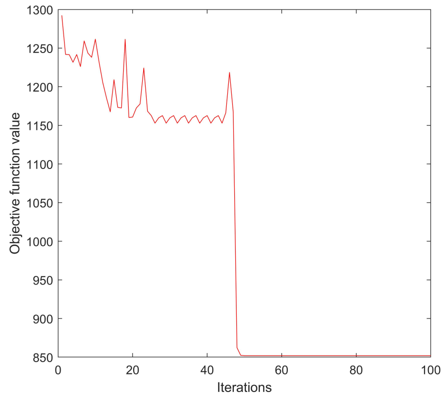


(g) Clustering MIs from Yale

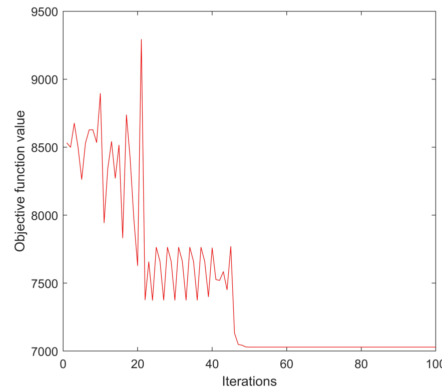


(h) Clustering ACs from Iris

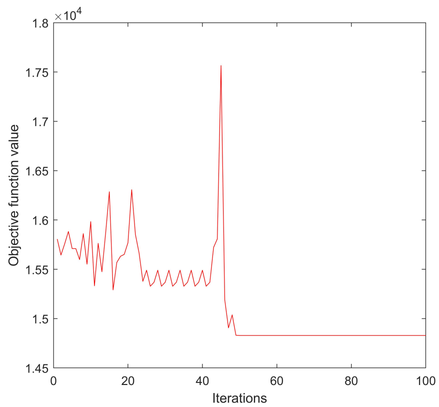
Fig. 1. Clustering NMIs and ACs from Iris, Wine, Seeds and Yale with different λ and β. The x-axis and y-axis are $lg\lambda$ and $lg\beta$.



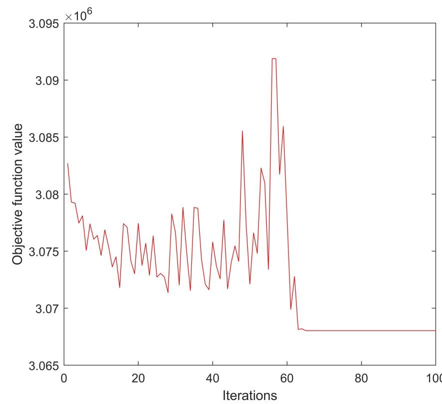
(a) The convergence curve of Iris



(b) The convergence curve of Wine



(c) The convergence curve of Seeds



(d) The convergence curve of Yale

Fig. 2. Iterations vs. objective values for FBC.

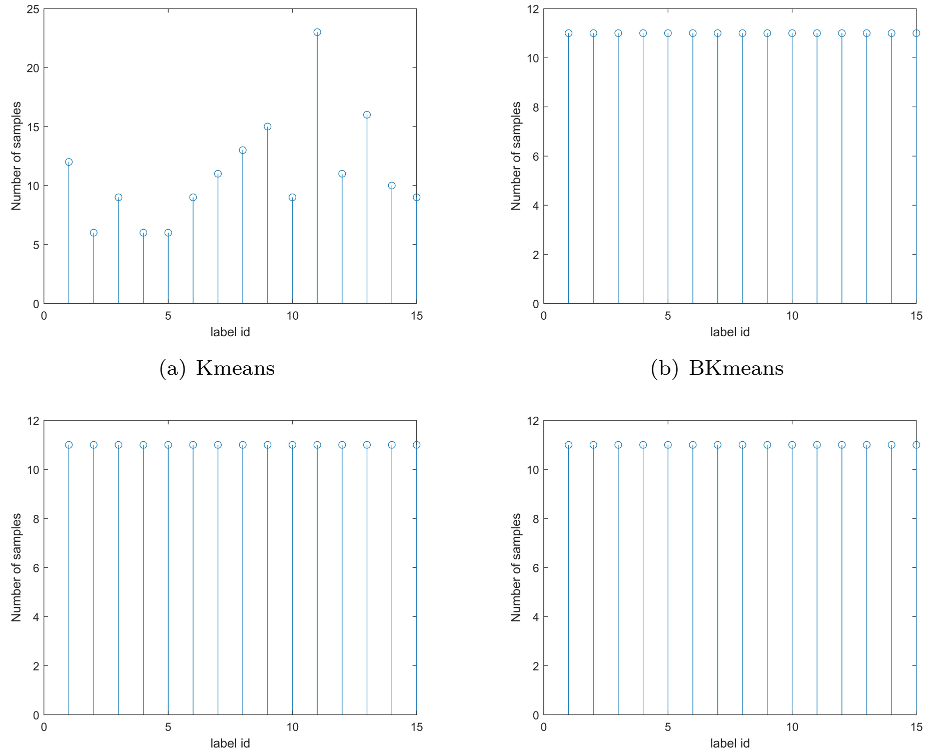
TABLE III
Clustering Performance (Evaluated by NMI) under The Optimal Parameter Settings for CKmeans, BKmeans, Kmeans and our methods on Yale, Wine, Iris, Seeds.

NMI

	Normalized Mutual Information (MIN/MAX/Avg.%)				
DATASET	FBC	FPSOBC	CKmeans	BKmeans	Kmeans
Yale	56.71/69.62/62.98	56.33/72.93/63.24	44.74/50.85/47.34	42.32/55.30/49.13	44.37/51.97/48.39
Wine	93.85/93.85/93.85	93.85/93.85/93.85	83.37/83.37/83.37	83.37/83.37/83.37	89.70/89.70/89.70
Iris	2.57/90.09/67.68	88.01/88.01/88.01	77.73/77.73/77.73	77.73/77.73/77.73	73.64/73.64/73.64
Seeds	84.51/88.40/87.76	88.40/88.40/88.40	64.90/64.90/64.90	64.90/64.90/64.90	69.35/69.35/69.35

TABLE IV
Clustering Performance (Evaluated by ACC) under The Optimal Parameter Settings for CKmeans, BKmeans, Kmeans and our methods on Yale, Wine, Iris, Seeds.

	Accuracy (MAX/MIN/Avg.%)				
DATASET	FBC	FPSOBC	CKmeans	BKmeans	Kmeans
Yale	50.91/66.67/59.21	51.52/70.91/59.64	38.18/44.85/41.15	35.76/52.73/43.94	38.18/46.06/42.42
Wine	98.61/98.61/98.61	98.61/98.61/98.61	98.83/95.83/95.83	98.83/95.83/95.83	97.22/97.22/97.22
Iris	36.00/97.33/82.20	96.67/96.67/96.67	92.00/92.00/92.00	92.00/92.00/92.00	88.67/88.67/88.67
Seeds	96.19/97.14/97.00	97.14/97.14/97.14	88.57/88.57/88.57	88.57/88.57/88.57	89.52/89.52/89.52



(a) Kmeans (b) BKmeans

(c) CKmeans (d) FBC

Fig. 3. Distribution of samples in clusters on Yale.

- Our soft-balanced clustering framework obtains better results than other algorithms.This phenomenon indicates that our method can use the sample data for better classification.
- Regardless of the dataset, FPSOBC always obtains a satisfactory and stable solution, which shows that it is significant to change the solution by PSO until a global solution is achieved.

V. Conclusion

In this paper, a fast balanced clustering framework is proposed, and PSO is effective to search for the optimal solution in a search space of feasible solution. The proposed method FPSOBC change the solution by PSO until a global solution is achieved. Experiments show that the soft-balanced clustering framework has better clustering performance than other balanced clustering algorithms. The satisfactory performance indicates that the proposed algorithm can search the global solution.

References

[1] Sarle and Warren S, “Algorithms for clustering data.”, 1990.
[2] J. Han, R. Quan, D. Zhang, and F. Nie, “Robust object co-segmentation using background prior,” IEEE Trans. Image Process., vol. 27, no. 4, pp. 1639-1651, 2018.
[3] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, “Advanced deeplearning techniques for salient and category-specific object detection: A survey,” IEEE Signal Process. Mag., vol. 35, no. 1, pp. 84-100, 2018.
[4] J. Han, G. Cheng, Z. Li, and D. Zhang, “A unified metric learning-based framework for co-saliency detection,” IEEE Trans. Circuits Syst. Video Technol., 2017.
[5] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, “When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 5, pp. 2811-2821, 2018.
[6] A. K. Jain, R. C. Dubes, “Algorithms for clustering data,” Prentice-Hall, Inc., 1988.
[7] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis. John Wiley & Sons, New York, 1973.
[8] E. S. Schaeffer, “Graph clustering,” Computer Science Review., vol. 1, no. 1, pp. 27-64,2007.
[9] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, “Parallel spectral clustering in distributed systems,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 3, pp. 568-586, Mar. 2011.
[10] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), vol. 2. Cambridge, MA, USA: MIT, pp. 849-856, 2002.
[11] Y. Chen, Y. Zhang, and X. Ji, “Size regularized cut for data clustering,” in Advances in Neural Information Processing Systems, Y. Weiss, P. B. Schölkopf, and J. C. Platt, Eds. Cambridge, MA, USA: MIT Press, pp. 211-218, 2006.
[12] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, “Constrained K-means clustering with background knowledge,” in Proc. ICML, vol. 1, pp. 577-584, 2001.
[13] J. Ye, Z. Zhao, and M. Wu, “Discriminative K-means for clustering,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), pp. 1649-1656, 2008.
[14] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang, “Parallel spectral clustering in distributed systems,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 3, pp. 568-586, Mar. 2011.

[15] M. I. Jain, P. Fränti, “Balanced k-means for clustering,” Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Springer, Berlin, Heidelberg., pp. 32-41, 2014.
[16] P. Bradley, K. Bennett, and A. Demiriz, “Constrained k-means clustering,” Microsoft Res., Redmond, WA, USA, May 2000, pp. 1¨C8.
[17] M. Levorato, R. Figueiredo, Y. Frota, and L. Drummond, “Evaluating balancing on social networks through the efficient solution of correlation clustering problems,” EURO J. Comput. Optim., vol. 5, no. 4, pp. 467-498, Dec. 2017.
[18] Y. Chen, Y. Zhang, and X. Ji, “Size regularized cut for data clustering,” in Advances in Neural Information Processing Systems, Y. Weiss, P. B. Schölkopf, and J. C. Platt, Eds. Cambridge, MA, USA: MIT Press, pp. 211-218, 2006.
[19] A. Banerjee and J. Ghosh, ¡°On scaling up balanced clustering algorithms,¡± in Proc. SIAM, vol. 2, pp. 333-349, 2002.
[20] J. Han, H. Liu, and F. Nie, “A local and global discriminative framework and optimization for balanced clustering,” IEEE transactions on neural networks and learning systems, vol. 30, no. 10, pp. 3059-3071, 2018.
[21] J. Kennedy, RC. Eberhart, “Particle swarm optimization,” In: Proc IEEE Int Conf on Neural Networks, WA Australia. pp. 1942¨C8, 1995.
[22] J. Kennedy, RC. Eberhart, Y. Shi, “Swarm intelligence,” San Francisco: Morgan Kaufmann Publishers, 2001.
[23] F. Rahimian, A. H. Payberah, S. Girdzijauskas, M. Jelasity, and S. Haridi, ”JA-BE-JA: A distributed algorithm for balanced graph partitioning,” in Proc. IEEE 7th Int. Conf. Self-Adapt. Self-Organizing Syst. (SASO), Sep. 2013, pp. 51-60, 2013.
[24] D.P.Bertsekas, ”Constrained optimization and Lagrange multiplier methods”, Academic Press, 1982.
[25] J. Eckstein, P. D. Bertsekas, “On the Douglas¡ªRachford splitting method and the proximal point algorithm for maximal monotone operators,” Mathematical Programming, vol. 55, no. 1, pp. 293-318, 1992.
[26] D. Cai, X. He, J. Han, and T. S. Huang, “Graph regularized nonnegative matrix factorization for data representation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 8, pp. 1548-1560, 2010.
[27] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, “Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering,” IEEE Trans. Neural Netw., vol. 22, no. 11, pp. 1796¨C1808, Nov. 2011.
[28] F. Nie, X. Wang, and H. Huang, “The constrained Laplacian rank algorithm for graph-based clustering,” in Proc. AAAI Conf. Artif. Intell., pp. 1969¨C1976, 2016.