A Novel Spectral Ensemble Clustering Algorithm Based on Social Group Migratory Behavior and Emotional Preference

Mingzhi Dai[1,2], Xiang Feng[1,2(✉)], Huiqun Yu[1,2], and Weibin Guo[1]

[1] Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
{Y30180707,xfeng,yhq,gweibin}@ecust.edu.cn
[2] Shanghai Engineering Research Center of Smart Energy, Shanghai 200237, China

Abstract. Clustering is an unsupervised machine learning technique for data mining to find objects with similar characteristics in a group. However, due to the lack of relevant prior information on the data, numerous single models or methods cannot identify the shape and size of the clusters. Therefore, an ensemble of multiple weak models is required to further mine the implicit information of the data and improve the clustering accuracy. LSMC-EPMC is an evolutionary clustering algorithm that consists of three parts, the emotional preference and migration behavior clustering (EPMC) model, the Laplacian spectral clustering model, and the Monte Carlo statistical data simulation model. This paper mainly integrates the spectral clustering model and the Monte Carlo statistical data simulation method into the EPMC algorithm by mapping the individual in EPMC and the optimized center point in the other two methods. Through numerous experiments, LSMC-EPMC shows a significantly increased performance to EPMC and is highly competitive with the other seven clustering algorithms on several standard datasets.

Keywords: Evolutionary optimization algorithm · Laplacian spectral-domain · Monte Carlo simulation · Ensemble learning · Data clustering

1 Introduction

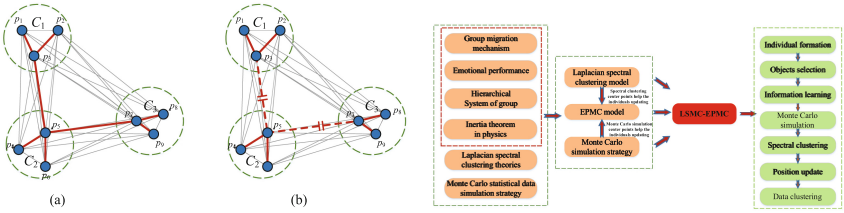
With the continuous growth and evolution of artificial intelligence, natural heuristic algorithms are becoming more and more popular among scholars. They can solve many complex optimization problems due to their intelligence, such as clustering tasks. These heuristic algorithms utilize principles similar to bionics to simulate the evolution, cooperation, or foraging process of animals or plants. For instance, as a well-known heuristic algorithm based on the foraging behavior of birds swarm system, particle swarm algorithm (PSO) [1] is widely used by scholars due to its fast convergence speed and easy implementation. The simulated

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13370, pp. 316–328, 2022.
https://doi.org/10.1007/978-3-031-10989-8_25

annealing (SA) algorithm [2] is a classic heuristic algorithm, which is a probability-based algorithm derived from the principle of solid annealing and has the characteristics of random iterative approximate asymptotic convergence. The ant colony optimization (ACO) algorithm [3] is inspired by the path planning behavior of ants in the process of foraging cooperation. Li et al. [4] propose a new heuristic optimization method which is an experience by the animal migration behavior (AMO), it mimics the process of animals transfer from one habitat to another. Inspired by natural biogeography and its mathematics, Simon et al. [5] proposed a biogeography-based optimization (BBO) algorithm to solve high-dimension optimization and real-world sensor selection problems.

Spectral clustering represents a bunch of clustering algorithms based on different graph cut theories. One of the classic graph cut algorithms is Cheeger cut. Its optimal segmentation is also an NP problem [6]. Szlam et al. [7] describe the process of Cheeger cuts balanced subgraphs. Therefore, the eigenvectors of the Laplacian matrix eigendecomposition can be used to settle the graph cut problem [8], so as to approximate the best partition mode of the clustering problem. As shown in Fig. 1(a), all data points can be considered fully connected. After the operation of graph cut, the optimal cutting mode can divide the whole graph into several partitions, which represent the most appropriate clustering mode.

In addition, as a classic numerical calculation method guided by probability and statistics theory [9] and the law of large numbers [10], the Monte Carlo method [11] utilizes an information-intensive and high-speed computing computer as a platform to transform complex research objects or calculation problems into simulations of random numbers and their digital characteristics. In machine learning [12,13], especially in reinforcement learning, a relatively fuzzy model is generally created for the obtained sample data set, and the parameters in the model are selected by the Monte Carlo method to make the residual error of the original data smaller.



(a) A graph with all data points are fully connected and the clustering mode C_1, C_2, C_3 by cutting two edges

(b) The sub-model structure of LSMC-EPMC

Fig. 1. Basic theory and sub-model structure of the LSMC-EPMC

In order to further efficient the clustering performance of emotion preference and migration behavior model, we integrated several learners and methods for learning, then proposed an ensemble algorithm called LSMC-EPMC in this

paper. The LSMC-EPMC contains three parts of the emotional preference and migration behavior model (EPMC) [14], the Laplacian spectral clustering model, and the Monte Carlo statistical data simulation model. The basic theory and sub-model structure of the LSMC-EPMC algorithm are shown in Fig. 1(b). The red frame represents the partition of the emotional preference and migration behavior model. The main contributions of this paper are as follows:

1. The manifold spectral clustering mode based on Laplacian eigenmaps is introduced to help update the position of the individual in the emotional preference and migration behavior model, and attempt to promote the performance when dealing with clustering tasks.
2. A Monte Carlo statistical data theory is designed to simulate the cluster center point to help the emotional preference and migration behavior model approaching the optimal individual, and endeavor to benefit the efficiency of clustering.
3. Numerous experiments were performed to compare the proposed ensemble model LSMC-EPMC with the other seven clustering algorithms on solving data clustering problems through testing on several standard datasets.

The rest content of this paper is organized as follows: The related models and theories of LSMC-EPMC are introduced in Sect. 2. Section 3 give the calculation steps and algorithm details of the proposed LSMC-EPMC. Section 4 shows the work of the experiment. Finally, Sect. 5 concludes the paper.

2 Mathematical and Physical Models

2.1 Emotional Preference and Migration Behavior Model

As a commonly used distance measurement criterion, Euclidean distance has been widely used in calculating the distance between data points. In this paper, we utilize the fitness function to evaluate the pros and cons of each individual.

Definition 1. *(Fitness function) As shown in Eq. 1, individuals in the population can be measured using fitness values, and the fitness function can be defined as:*

$$fit(p, M) = \frac{p \times \sum_{i=1}^{n} \left\| Ins_i - Ins_{label} \right\|^2}{\min \left\| x_a - x_b \right\|} \tag{1}$$

where p and M represent the number of the clusters and the individual matrix, respectively. And Ins_i and Ins_{label} represent an instance in the dataset and the label class it belongs to, x_a and x_b represent two clustering centers in individual matrix M. The smaller the numerator, the smaller value of the function. Therefore, the goal of the optimizing is to minimize the fitness function Eq. 1 and find the individual M_{op}.

This section briefly introduce the emotional preference and migration behavior (EPMC) model proposed by Feng et al. [14]. It consists of four sub-parts: the migration model, the emotional preference model, the social group model, and the inertial learning model. The first two models can help individuals find the global optimal learning object and the best learning object nearby. The third model divides the population into two groups to improve learning ability, and the last model can help individuals explore more solution space.

2.2 Manifold Laplacian Spectral Clustering Model

Let $G = (V, E)$ be an undirected graph with vertex set $V = v_1, ..., v_n$. Supposing that the graph G is weighted, w_{ij} represents the non-negative weight between the two vertices v_i and v_j. For any two vertices in V, there can be an edge connection or no edge connection. Since it is an undirected graph, $w_{ij} = w_{ji}$. The similarity matrix S of the sample point distance measurement is used to obtain the weighted adjacency matrix $W = (w_{ij})_{i,j=1,2,...,n}$ of the graph. For any point v_i in the graph, its degree d_i is defined as the sum of the weights of all edges connected to it.

According to the definition of the degree of each point, we can get a nn degree matrix D, which is a diagonal matrix, as defined in Eq. 2:

$$D = \begin{pmatrix} deg\,(v_1) & \cdots & \cdots \\ \vdots & \ddots & \vdots \\ \cdots & \cdots & deg\,(v_n) \end{pmatrix} \tag{2}$$

For the cut graph of the undirected graph G, our proposal is to cut the graph $G = (V, E)$ into k subgraphs that are not connected. A subgraph $G^S = \left(S, E^S\right)$ of $G = (V, E)$ is composed of a set of vertices $S \subseteq V$ and a set of edges $E^S \subseteq E$. The set of each subgraph point is: $G_1^S, G_2^S, \ldots G_k^S$, they satisfy $G_i^S \cap G_j^S = \emptyset$, and $G_1^S \cup G_2^S \cup \ldots \cup G_k^S = G$. For the set of any two subgraph points $A, B \subset V$, $A \cap B = \emptyset$, the weight of the cut between A and B is $W(A, B) = \sum_{i \in A, j \in B} w_{ij}$. Then for the set of k subgraph points $G_1^S, G_2^S, \ldots G_k^S$, the NCut is donated as Eq. 3,

$$NCut\left(G_1^S, G_2^S, \ldots G_k^S\right) = \frac{1}{2} \sum_{i=1}^{k} \frac{W\left(G_i^S, \bar{G}_i^S\right)}{vol\left(G_i^S\right)} \tag{3}$$

where $vol\left(G^S\right) = \sum_{i \in A} d_i$. The unnormalized graph Laplacian matrix is defined as:

$$L = D - W \tag{4}$$

In this way, we can continue to follow the idea of *Ratio*Cut to find the smallest first k eigenvalues of $L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$, then find the corresponding eigenvectors, and standardize them to get the final eigenmatrix F. Finally, it is sufficient to perform traditional clustering on the matrix F.

2.3 Determining Parameters for the Model Using the Monte Carlo Method

The *Naive* Monte Carlo method is the most popular and frequently used method to settle multi-dimensional MC problems. Supposing an approximate computation of the integral $Q[f] = \int_\Omega f(\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}$, and the expectation the random variable $\phi = f(\delta)$ is that $\mathbf{E}\phi = \int_\Omega f(\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}$. Among them, δ is a random point in the probability density function $p(x)$, and all variables $\delta_1, \delta_2, \ldots, \delta_N$ are independent with $\phi_1 = f(\delta_1), \ldots, \phi_N = f(\delta_N)$. Common Monte Carlo estimation is based on the N repeated simulations and simulate the dataset through the minimum cut set. Given the simulation set $\tau_i = f_s\left(x_1^{(i)}, \ldots, x_n^{(i)}\right)$, the common Monte Carlo estimate can be expressed as the following form: $\widehat{T}_s = \frac{1}{N}\sum_{i=1}^{N} f_s\left(x_1^{(i)}, \ldots, x_n^{(i)}\right)$. The variance estimated by the Monte Carlo algorithm is proportional to N^{-1}, and $Var\left(\hat{T}_s\right) = Var\left(\frac{1}{N}\sum_{i=1}^{N}\tau_i\right) = \frac{1}{N^2}Var\left(\sum_{i=1}^{N}\tau_i\right) = \frac{1}{N}Var\left(\tau_i\right)$. More generally, the central limit of Monte Carlo can be expressed as:

$$P\left(\left|T_s - E\left(T_s\right)\right|\right) > z\frac{\sqrt{\mathrm{Var}\left(\tau_i\right)}}{\sqrt{n}} \approx P(|Z| > z) \tag{5}$$

where $Z \sim N(0,1)$. Therefore, for the expected accuracy $\varepsilon > 0$ with a confidence level of $1-\alpha$, Monte Carlo simulation with $n = z_{\frac{\partial}{2}}^2 \mathrm{Var}\left(\tau_i\right)\varepsilon^{-2}$ is required, where the quantile $z\frac{\alpha}{2}$ is selected to ensure $P\left(Z > z\frac{\alpha}{2}\right) = \frac{\alpha}{2}$. Naturally, $z\frac{\alpha}{2}$ is a constant for any regular confidence level.

3 The Proposed LSMC-EPMC Optimization Algorithm

3.1 Main Framework of LSMC-EPMC Algorithm

The LSMC framework consists of 3 main components and 6 sub-parts. First, the emotional preference and migration behavior model initializes the population and other parameters, and evaluates and ranks the individuals in the population. Then individuals update their position based on the best individuals in the population and the excellent neighbor around them, and the model uses a certain strategy to eliminate inferior individuals in the population. Second, by matching individual cluster centers with a similar class of spectral cluster centers, Laplacian spectral cluster solutions can help individuals update their positions. Finally, the Monte Carlo method can simulate similar center points to assist individual's update according to the law of large numbers.

3.2 Calculation Steps of LSMC-EPMC Algorithm

The optimization steps of the proposed LSMC-EPMC algorithm can be listed as follows:

Step 1: Data preprocessing and initialization
During the preprocessing, the upper and lower bounds of the dataset are calculated. The parameter initialization contains the scale of the population (NP), the number the elite individuals($elite$), the initial number of clusters(k), the maximum number of iterations($iteration$), the maximum number of running rounds(run), and so on.

Step 2: Evaluation and grouping
All individuals evaluate their fitness values according to Eq. 1 and sort the NP individuals in the population from small to large. Among them, the top $elitenum$ individuals are considered as $elite$, and the remaining individuals are considered as ordinary individuals. Besides, all data points are assigned to the nearest centroid (a row of the individual matrix M_i). When a centroid has no data point or only one data point, the centroid will be ignored and the data point will be reassigned.

Step 3: Selection of learning objects
As depicted in Sect. 2.1 [14], the individual selects the best individual in the population.

Step 4: Laplacian spectral cluster centers learning

Hypothesis 1. Supposing that the emotional preference and migration behavior model contains p central points at the t-th iteration, the $elite$ are selected to be updated. When considering using the center point of spectral clustering to update the centroid in an individual, the mode combined with spectral clustering is recorded as SpectClus-p, that is, there are p spectral clustering center points that assist the base model to update the individual.

Hypothesis 2. Supposing that the classes with the largest number of temporary labels in the current iteration on the Laplacian learning are the actual classes of the dataset.

Hypothesis 3. In the current iteration, if the Euclidean distance between a certain spectral cluster center point and a certain centroid of the individual is the smallest, the spectral cluster center-point can be identified as the same class as the centroid.

After matching the actual centroid of the individual with the same class of the spectral clustering center point one by one, then the SpectClus-p method can utilize to update the individual. The SpectClus-p method acts on $elite$, and ordinary individuals follow the elite individuals in the population to learn and update to reach the optimal more properly.

Step 5: Individual learning and updating
After selecting the ideal learning object, the individual updates the position. If the individual's fitness becomes better, go to **Step 6**, otherwise, The model will determine whether to accept this update.

Step 6: Replace of emotional preference matrix

After the individual updates the position, the corresponding emotional preference matrix $feel_{i,j}$ is updated simultaneously. The elements in the emotional preference matrix are set to 1 initially.

Step 7: Elimination of inferior individuals
In each iteration, the LSMC-EPMC algorithm eliminates individuals with poor fitness values in the second half of the population with a certain probability P_e, and adds new individuals to the population.

Step 8: Monte Carlo simulate cluster centers learning

Hypothesis 1. Supposing that the classes with the largest number of temporary labels in the current iteration on the Monte Carlo simulate are the actual classes of the dataset.

Hypothesis 2. After using the Monte Carlo method to expand and filter the data features of several classes in the current iteration, the mean value of the expanded data is selected as the current better center point $Center_{mc}$.

Hypothesis 3. In the current iteration, if the Euclidean distance between a certain Monte Carlo simulation center and a certain centroid of the individual is the smallest, the Monte Carlo simulation center can be identified as the same class as the centroid.

After matching the actual centroid of the individual with the same class of the Monte Carlo simulation center one by one, then the Monte Carlo simulation center is utilized to update the individual. The Monte Carlo simulation method acts on *elite*, and ordinary individuals are updated synchronously in social group to reach the optimal.

Step 9: Termination
The algorithm repeats **Step 2** to **Step 8** until met the termination condition or reached the maximum number of iterations. Finally, the optimal solution is obtained, and the algorithm ends.

Suppose that the computational complexity of choosing a learning object is O_c in the LSMC-EPMC. Moreover, O_e and O_o represent the computational complexity of position updating rule for elite and ordinary, respectively. The computational complexity of spectral and Monte Carlo operator are represented by O_s and O_m, respectively. In addition, the size of elite and ordinary (N_e and N_o) of LSMC-EPMC. So the computation complexity of LSMC-EPMC is $N_e \cdot (O_c + O_e) + N_o \cdot (O_c + O_o) + O_s + O_m$.

4 Experiment and Results

This section discusses the computational experiments performed with the proposed LSMC-EPMC algorithm. And 11 UCI standard datasets were used in the experiment, including the Iris, Soybean, Glass, Seeds, Vowel, Car Evaluation (CE), User Knowledge (UK), Wine, BLOOD, Hagerman's Survival (HK) and Banknote. Authentication (BA).

The operating environment of all experiments is running on Lenovo Shenteng 1800 HPCC, which has 8 computing nodes and 1 console node. Each compute node is a high-performance server with two 2.4 GHz quad-core CPUs and 24 GB of memory. The operating system of all servers is Red Hat Enterprise Linux 7, and the experimental computing platform is MATLAB R2021b.

Table 1. Parameter settings details on eight algorithms

Algorithm	Parameter setting
EPMC	As depicted in [14]
K-Means	k is set to be equal to the real $ClusNum$ of the dataset
Cop-kmeans	k is set to be equal to the real $ClusNum$ of the dataset, the indices of the pairs that must be in the same cluster nML is set to 8, the indices of the pairs that cannot be in the same cluster nCL is set to 8
Graph-SSC	The certainty of each observed label is set to 10, the number of initial labels in each class is set to 10, the number of nearest neighbors is set to 29, and set the hyperparameter $\alpha = 100$, $\beta = e^{-3}$
PSO	The weight factor w reduces from 0.9 to 0.4 and c_1, c_2 are set to 2
AMO	The range of critical interval is set to 5
BBO	The number of $elite$ is set to $elitenum = 2$, and the mutation rate is set to $P_e = 0.05$
LSMC-EPMC	The number of $elitenum$ is set to 2, the number of iterations T is set to 100, the initial number of clusters k is set to 10 (Vowel is 15), the number of $runs$ is set to 25, set $\alpha = 0.8$ and $\beta = 0.2$, the scale of Monte Carlo simulation data is set to 5000, the hyperparameter of Gaussian similarity in spectral clustering is set to 0.9, and the max mutation rate $P_{e,max}$ is set to 0.05

The computational result is compared with the other seven algorithms, including a machine learning algorithm: K-means algorithm, two semi-supervised algorithms: Cop-kmeans algorithm, Graph-SSC algorithm, and four optimization algorithms: PSO algorithm, BBO algorithm, AMO algorithm and EPMC algorithm. In all experiments, the population size of the eight algorithms is set to 15, and the maximum number of iterations is set to 100. The initial number of clusters is set to 15 for Vowel and 10 for the rest of the datasets. All algorithms run 25 times independently, and we take the average value as the final result for comparison. The parameter setting details of the remaining algorithms are shown in Table 1.

The clustering performance of the eight algorithms is evaluated according to four criteria. The first one is the internal quality ($fitness$) measure, which is defined in Eq. 1. The smaller the $fitness$, the better the performance. The second one is the Jaccard coefficient ($JacIndex$). Kou et al. [15] use the Jaccard coefficient to test the quality of clustering results, which measure the similarity

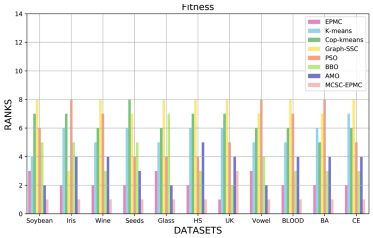
between instances properly. The larger the $JacIndex$, the better the algorithm. The third one is the Average number of clusters ($ClusNum$), which is calculated by Eq. 6. The closer the result is to the actual classes of the dataset, the better the algorithm. The last one is the $Time$, which records the speed of the eight algorithms on clustering. The smaller the $Time$, the better the algorithm.

$$ClusNum = \frac{\sum_{i=1}^{run} clusnum_i}{run} \tag{6}$$

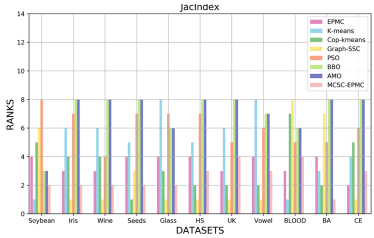
where $clusnum_i$ represents the final number of clusters in each round, and run represents the execute round of the algorithm.

The results of the $fitness$, $JacIndex$, $ClusNum$ and $Time$ executed by the eight optimization algorithms are listed in Tables 2 and 3, respectively. In Table 2, the data shows the Average Best-so-far (Avg), Best-so-far ($Best$), Worst-so-far ($Worst$), and the standard deviation of Best-so-far-solution (Std) of all execution results. The best Avg of the eight algorithms are shown in bold.

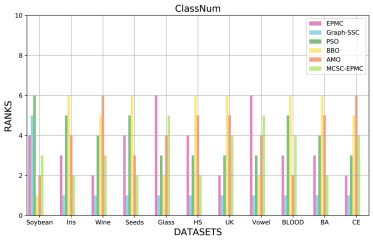
In order to analyze the performance of all algorithms more intuitively, a summary of Friedman test is used. The smaller the rank of the test result, the better the algorithm. Figure 2(a) describes the Friedman ranking results on the $fitness$ of eight algorithms. From Fig. 2(a) we can conclude that the LSMC-EPMC algorithm gets the ten best results among 11 datasets, and is only slightly



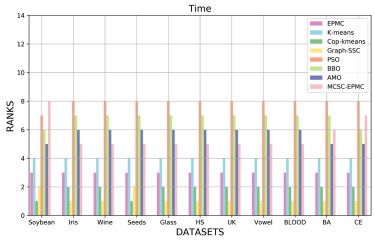
(a) Friedman ranking results on the $fitness$



(b) Friedman ranking results on the $JacIndex$



(c) Friedman ranking results on the $ClusNum$



(d) Friedman ranking results on the $Time$

Fig. 2. The Friedman ranking results on the $fitness$, $JacIndex$, $ClusNum$ and $Time$ of eight algorithms

Table 2. Fitness results on eight algorithms

Dataset	Indices	EPMC	K-means	Cop-kmeans	Graph-SSC	PSO	BBO	AMO	LSMC-EPMC
Iris	Avg	25.95	33.68	34.59	29.38	41.48	32.29	30.87	**20.46**
	Best	20.83	28.72	25.86	28.43	30.16	27.05	24.67	18.89
	Worst	34.60	42.28	46.39	31.26	49.38	35.78	38.18	23.18
	Std	2.85	5.83	5.69	0.96	5.71	2.45	3.46	0.998
Soybean	Avg	7.63	21.72	30.01	49.67	28.76	26.13	7.32	**6.11**
	Best	6.64	18.65	18.95	29.74	21.33	23.50	6.44	5.26
	Worst	9.54	31.09	40.47	54.00	34.86	28.72	7.97	7.38
	Std	0.75	5.04	4.27	7.29	3.11	1.24	0.40	0.543
Wine	Avg	16.21	33.86	57.78	109.74	61.57	17.90	18.15	**15.41**
	Best	17.69	36.52	51.52	68.63	49.01	18.74	18.77	16.69
	Worst	21.74	37.95	68.89	168.7	91.23	23.05	23.32	22.83
	Std	1.11	0.47	4.16	24.58	8.89	1.24	1.34	1.404
Seeds	Avg	20.48	60.16	67.60	62.66	35.46	49.7	23.94	**16.99**
	Best	17.20	58.59	63.66	59.65	30.52	43.35	18.02	16.24
	Worst	24.65	85.94	77.59	65.88	41.94	58.65	28.95	19.43
	Std	1.94	5.38	3.00	1.93	3.01	3.33	3.47	0.772
Glass	Avg	6.88	49.44	52.47	88.26	8.41	55.25	6.47	**5.69**
	Best	5.11	34.72	42.80	62.43	7.62	41.15	4.67	4.51
	Worst	9.42	84.34	64.71	104.1	9.21	72.76	8.12	6.85
	Std	1.16	11.01	6.11	9.25	0.37	7.54	0.89	0.774
HS	Avg	55.36	151.06	155.21	255.5	59.92	55.64	70.36	**54.36**
	Best	49.05	147.2	150.9	156.7	53.43	47.76	59.40	48.63
	Worst	63.35	187.8	159.7	722.6	63.02	60.90	80.26	58.98
	Std	3.97	8.04	2.38	115.7	2.71	2.98	5.85	3.44
UK	Avg	**163.73**	287.51	290.06	325.18	271.13	165.9	241.34	187.34
	Best	126.6	275.2	278.08	291.7	235.1	153.9	213.7	137.01
	Worst	217.8	293.7	308.85	372.6	307.1	169.8	253.1	212.41
	Std	30.80	7.26	9.49	25.37	17.19	4.69	10.98	20.57
Vowel	Avg	259.08	508.72	511.88	773.86	811.39	390.9	249.13	**221.0**
	Best	213.3	472.5	494.36	681.0	682.9	354.6	181.9	174.1
	Worst	364.9	528.3	540.54	875.5	907.9	420.9	347.6	282.8
	Std	47.00	13.28	10.19	46.12	55.09	17.76	48.03	31.80
BLOOD	Avg	27.47	154.33	172.48	5937	454.85	28.83	39.18	**26.09**
	Best	24.30	154.33	167.27	399.1	380.1	22.89	32.69	22.28
	Worst	33.66	154.33	177.06	48762	612.0	31.41	44.44	30.58
	Std	2.02	0	2.71	9915	62.39	2.20	3.35	1.81
BA	Avg	375.82	639.24	608.40	1281.85	1681.60	381.7	503.19	**367.04**
	Best	348.6	629.5	600.03	974.3	1123	289.6	409.2	342.31
	Worst	445.4	643.75	617.24	2399	2374	406.5	514.9	387.55
	Std	24.19	3.48	4.45	269.3	313.4	28.16	27.52	13.50
CE	Avg	1093.5	1622.9	1561.44	2002.45	1492.4	1371	1417.7	**1063.5**
	Best	730.2	1353	1506.6	1854	930	893.4	1173	1106.0
	Worst	1263	1587	1665.4	2304.6	1122	996.9	1268	1268.0
	Std	161.7	66.75	48.08	110.7	46.82	23.17	22.52	35.08

inferior to EPMC and BBO algorithm on the UK dataset during the $fitness$ criterion. Specifically, the proposed LSMC-EPMC algorithm and three classic algorithms (PSO, BBO, AMO) are superior to the two semi-supervised learning methods (Cop-kmeans, Graph-SSC) on the $fitness$ criterion. This may be due to the defective algorithm design, which leads to individuals escaping the optimal

Table 3. Results of *JacIndex*, *ClusNum* and *Time* on eight algorithms

Dataset	Measure	EPMC	K-means	Cop-kmeans	Graph-SSC	PSO	BBO	AMO	LSMC-EPMC
Iris	JacIndex	0.62	0.27	0.34	0.74	0.23	0.19	0.28	0.68
	ClusNum	2.56	–	–	3.00	3.88	6.32	3.84	3.08
	Time	9.74	10.88	0.72	0.05	45.20	37.20	25.80	13.23
Soybean	JacIndex	0.40	0.56	0.39	0.34	0.26	0.45	0.33	0.52
	ClusNum	2.16	–	–	2.08	6.36	4.36	3.00	2.8
	Time	4.84	5.29	0.04	0.06	15.40	13.00	9.99	53.66
Wine	JacIndex	0.30	0.20	0.26	0.34	0.21	0.12	0.12	0.31
	ClusNum	3.08	–	–	3.00	4.28	9.15	9.20	3.24
	Time	12.22	14.35	0.09	0.06	54.40	48.40	32.60	32.08
Seeds	JacIndex	0.51	0.27	0.60	0.51	0.23	0.16	0.26	0.57
	ClusNum	2.20	–	–	3.00	5.48	10.0	3.44	2.72
	Time	11.93	15.38	0.11	0.15	63.80	52.60	36.20	23.7
Glass	JacIndex	0.26	0.11	0.26	0.34	0.15	0.21	0.24	0.30
	ClusNum	2.00	–	–	6.00	4.60	5.72	3.00	2.56
	Time	11.84	19.88	0.12	0.09	65.10	56.90	37.20	25.14
HS	JacIndex	0.32	0.31	0.38	0.51	0.27	0.15	0.29	0.32
	ClusNum	4.76	–	–	2.00	4.76	8.40	6.00	4.76
	Time	16.45	18.45	0.07	0.06	91.60	53.80	51.30	22.05
UK	JacIndex	0.26	0.12	0.27	0.36	0.13	0.08	0.08	0.24
	ClusNum	3.32	–	–	4.00	5.76	10.0	9.20	6.6
	Time	20.75	27.97	0.36	0.08	121.0	99.40	68.20	38.14
Vowel	JacIndex	0.10	0.03	0.16	0.28	0.05	0.04	0.08	0.10
	ClusNum	2.20	–	–	11.0	6.32	8.04	3.32	2.52
	Time	73.4	90.13	0.84	0.25	393.0	391.0	272.0	147.8
BLOOD	JacIndex	0.40	0.55	0.20	0.20	0.39	0.22	0.46	0.39
	ClusNum	9.52	–	–	2.00	9.76	9.80	8.96	9.64
	Time	36.95	41.44	0.12	0.09	222.9	159.8	113.2	84.53
BA	JacIndex	0.36	0.36	0.38	0.30	0.33	0.25	0.32	0.38
	ClusNum	5.04	–	–	2.00	5.08	7.00	5.84	4.84
	Time	66.11	73.22	0.41	0.14	417.7	333.1	208.7	210.64
CE	JacIndex	0.32	0.25	0.24	0.42	0.19	0.11	0.12	0.30
	ClusNum	2.84	–	–	4.00	8.36	10.0	10.00	9.36
	Time	37.3	63.39	1.13	0.15	526.8	365.4	283.8	391.22

solution prematurely. In general, except the User Knowledge dataset, LSMC-EPMC can obtain superior results among eight algorithms include the EPMC optimization algorithm on the *fitness* criterion.

Then, Fig. 2(b), Fig. 2(c) and Fig. 2(d) indicate the Friedman ranking results on the *JacIndex*, *ClusNum* and *Time* of eight algorithms, respectively. As can be seen from Fig. 2(b), the semi-supervised learning method Graph-SSC is superior to other algorithms on most datasets during the *JacIndex* criterion, which possibly rely on the thorough consideration of the data by the graph structure. And the LSMC-EPMC can get the superior results among most datasets on the *JacIndex* criterion. Since K-means and Cop-kmeans have determined the initial number of clusters, Fig. 2(c) only shows the Friedman ranking results of the LSMC-EPMC algorithm and other six algorithms on the *ClusNum* criterion. As can be concluded from Fig. 2(c), the LSMC-EPMC algorithm is slightly inferior to the semi-supervised learning method Graph-SSC, but is higher than the

other five algorithms including EPMC on most datasets during the $ClusNum$ criterion. Besides, Fig. 2(d) describes the running time Friedman ranking of the eight algorithms. We can be seen from Fig. 2(d) that the LSMC-EPMC algorithm shows worse speed than two semi-supervised learning methods, but its Friedman ranking is higher than the other three classic algorithms and EPMC on most datasets during the $Time$ criterion. In addition, the Friedman ranking of the proposed LSMC-EPMC algorithm is superior to all other seven algorithms on the $fitness$ criterion.

5 Conclusion

Based on the emotional preference and transfer behavior model, an ensemble algorithm called LSMC-EPMC which merged several learners and methods was proposed in the paper. First, we incorporate the spectral clustering based on Laplacian eigenmaps to update the individual position in optimization. Second, a Monte Carlo statistical data theory is used to simulate the cluster center point and help to approach the optimal. Third, the proposed LSMC-EPMC is applied to settle the data clustering tasks.

Then, numerous experiments were performed to compare the proposed LSMC-EPMC with the other seven clustering algorithms on several standard datasets. The paper utilized four criteria to measure the clustering performance of the eight algorithms. In addition, the Friedman test was used to analyze the ranking of the eight algorithms. Through the Friedman ranking results, we can conclude that the clustering performance of the proposed LSMC-EPMC is better than the other seven algorithms including the EPMC.

However, there are still many flaws that need to be settled. For instance, on the high-dimensional dataset (Vowel dataset) or the large-scale dataset (BA dataset), the number of centers searched by the LSMC-EPMC is still far from the actual number of classes. Furthermore, time consumption also requires more attention. In the future, we will focus on the application of LSMC-EPMC to real-world unmanned system mixed precision problems, and further improve the biophysical and mathematical models of LSMC-EPMC to realize the parallelism.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grant NOs. 61772200, Shanghai Pujiang Talent Program (17PJ1401900), the Information Development Special Funds of Shanghai Economic and Information Commission under Grant NO. XX-XXFZ-02-20-2463, and the Key Program of National Natural Science Foundation of China (62136003).

References

1. Lv, J., Shi, X.: Particle swarm optimization algorithm based on factor selection strategy, pp. 1606–1611 (2019)
2. Xin, X., Li, K.-J., Sun, K., Liu, Z., Wang, Z.-D.: A simulated annealing genetic algorithm for urban power grid partitioning based on load characteristics, pp. 1–5 (2019)

3. Guan, B., Zhao, Y., Li, Y.: An improved ant colony optimization with an automatic updating mechanism for constraint satisfaction problems. Expert Syst. Appl. **164**, 114021 (2021)
4. Li, X., Zhang, J., Yin, M.: Animal migration optimization: an optimization algorithm inspired by animal migration behavior. Neural Comput. Appl. **24**, 1867–1877 (2014)
5. Simon, D.: Biogeography-based optimization. IEEE Trans. Evol. Comput. **12**, 702–713 (2008)
6. Qiu, H., Zheng, Q., Msahli, M., Memmi, G., Qiu, M., Jialiang, L.: Topological graph convolutional network-based urban traffic flow and density prediction. IEEE Trans. Intell. Transp. Syst. **22**(7), 4560–4569 (2021)
7. Szlam, A., Bresson, X.: Total variation and cheeger cuts, pp. 1039–1046 (2010)
8. Dai, M., Guo, W., Feng, X.: Over-smoothing algorithm and its application to GCN semi-supervised classification. In: Qin, P., Wang, H., Sun, G., Lu, Z. (eds.) ICPCSEE 2020. CCIS, vol. 1258, pp. 197–215. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-7984-4_16
9. Qin, W.: A study on real estate price by using probability statistics theory and grey theory, pp. 153–156 (2019)
10. Ma, H., Sun, Y., Miao, Yu.: Some extensions of the classical law of large numbers. Commun. Stat. Theory Methods **49**, 3228–3237 (2020)
11. Mikhov, R., et al.: A two-stage Monte Carlo approach for optimization of bimetallic nanostructures, pp. 285–288 (2020)
12. Mohamed, S., Rosca, M., Figurnov, M., Mnih, A.: Monte Carlo gradient estimation in machine learning. J. Mach. Learn. Res. **21**, 1–62 (2020)
13. Kimmel, R., Li, T., Winston, D.: An enhanced machine learning model for adaptive Monte Carlo yield analysis, pp. 89–94 (2020)
14. Feng, X., Zhong, D., Yu, H.: A clustering algorithm based on emotional preference and migratory behavior. Soft. Comput. **24**(10), 7163–7179 (2019). https://doi.org/10.1007/s00500-019-04333-4
15. Kou, G., Peng, Y., Wang, G.: Evaluation of clustering algorithms for financial risk analysis using MCDM methods. Inf. Sci. **275**, 1–12 (2014)