Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 8662238, 9 pages
https://doi.org/10.1155/2022/8662238

Research Article

Algorithm for Filling High Rank Matrix of Network Big Data Based on Density Peak Clustering

Deqiang Liu

Feixian Campus of Linyi University, Linyi 276000, China

Correspondence should be addressed to Deqiang Liu; ldq39@163.com

Received 17 February 2022; Revised 13 May 2022; Accepted 16 May 2022; Published 25 June 2022

Academic Editor: Wen-Tsao Pan

The traditional filling method for network big data matrix has poor filling effect and suffers from noise. Therefore, a filling algorithm for network big data high rank matrix based on density peak clustering is proposed. The missing data are replaced by small-interval data, the information entropy of the high rank matrix of network big data is calculated, the density peak clustering algorithm is optimized through the cluster center selection strategy, the block data set is obtained through the unknown block method, and the block filling is realized by the host filling algorithm. Experimental results show that the filling accuracy of the proposed algorithm is as high as 0.895, and the loss rate is between 2% and 12%.

1. Introduction

Matrix filling is one of research hotspots in the fields of matrix analysis, optimization, image processing, means filling the missing elements accurately through known elements in the case of missing elements in the sampling matrix, and finally completing the sampling matrix [1]. In practice, the sampling matrix sometimes has special structures, such as symmetric matrix and Toeplitz matrix, which play an important role in communication engineering and power system, especially in the field of signal and image processing [2–4].

There are two kinds of methods to deal with the problem of matrix rank minimization: one is to relax the rank function convex into the matrix kernel norm and establish the optimization model of the kernel norm; the second is to give the rank of the matrix in advance and establish a low-rank decomposition model [5]. Many domestic experts and scholars have also carried out a lot of researches on matrix filling and applied the research results to the fields of image processing, text analysis, and recommendation system [6].

In the aspect of image restoration, the large singular values in the data matrix retain more characteristics of the original data, while the small singular values contain more noise. Matrix filling technology has been widely used in data analysis, recommendation system, image filling, video denoising, and machine learning [7, 8].

The researches on big data require the involvement of a huge amount of information, which is usually collected and stored in daily life, but the process is carried out without supervision. Once external interference occurs, it will inevitably produce some missing data [9]. The collected data usually contain important information, and if it cannot be processed in time or improperly, there will be a serious impact on the real-time and effectiveness of the data, and even some wrong data information may appear, leading to users' wrong decisions. In view of missing data, we need to fill in the data in time [10–15].

Relevant scholars have studied this problem and achieved some results. Sun et al. [16] proposed the missing data filling algorithm based on the improved neural process [16], expressed the observed time series in a single way, obtained their respective characterization vectors by the neural network, obtained the distribution function of the data through the neural process model, introduced the correction coefficient in the training stage, determined the sampling rate of the training data more accurately according to the data missing rate, and estimated the missing value of the data through the trained model. The results show that this algorithm has good filling effect in the context of small data sets, but the filling accuracy of missing data is poor. Lin et al. [17] optimized K based on cuckoo algorithm—the missing data filling algorithm of means clustering [17]. By

taking the training error of neural network as the fitness function, the weight and threshold of neural network are optimized, the weight of neural network through cuckoo search algorithm is calculated, and *K*-means clustering algorithm is adopted to realize the optimization of missing data threshold and the filling of missing data. This method has high missing data filling efficiency, but the data loss rate is unacceptable.

This paper presents a high rank matrix filling algorithm for network big data based on density peak clustering. The specific steps are as follows.

In the first step, density peak clustering is introduced, the local density is calculated based on cutoff kernel function, and the clustering center is obtained by comprehensively considering the local density value and minimum distance value of data points [18–22]; the second step is to classify the nonclustering center points, identify the abnormal points, replace the lost data with small-interval data, and calculate the information entropy of the filling data; in the third step, the distance measurement is obtained by the density peak clustering algorithm to realize data segmentation, and the complete sub-data set is obtained by the host filling method to complete the high rank matrix filling algorithm of network big data [23–28].

The fourth step is to verify the effectiveness of the proposed method and draw a conclusion.

2. High Rank Matrix Filling Algorithm for Network Big Data

2.1. Big Data Information Entropy Calculation. Set the data with missing correlation to $W = (U, A)$, U represents the object set, and A represents the attribute set. If $u_i \in U$, $i = 1, 2, \ldots |U|$, attribute $\alpha \in A$, that is, the interval data will be missing data $f(u_i, \alpha)$, which can be represented by " $*$."

Compared with a missing data set, if the missing data are replaced by the inter-cell, the information entropy of the large data set will increase significantly [29–36].

On the big data set W, the attribute condition $B \in A$ is met, and $b_1, b_2, \ldots, b_N \in B$, $U_k \in U$. Assuming that the m data b_m is missing data, the data set U_k that has not been filled is called the original data, and the information entropy is calculated according to the formula as follows:

$$H_\lambda^N(A)|_0 = -\sum_{i=1}^{|U|} \frac{1}{|U|} \lg\left(\frac{1}{|U|}\sum_{j=1}^{|U|} P_{uiuj}^{bl}\right) \sum_{l=1}^{N} \lambda_l(k), \tag{1}$$

where P_{uiuj}^{bl} represents the minimum multi-interval similarity between u_i and u_j, and $\lambda_l(k)$ represents the correlation coefficient between the l index and the k index, both of which can be expressed as follows:

$$P_{uiuj}^{bl} = \frac{\left|f(u_i, b_l) \cap f\left(u_j, b_l\right)\right|}{\left|f(u_i, b_l) \cup f\left(u_j, b_l\right)\right|},$$
$$\lambda_l(k) = \frac{\lambda(b_k, b_l)}{\sum_{l-1}^{n-1} \lambda(b_k, b_l) + \lambda(b_k, b_k)}, \tag{2}$$

where $f(\cdot)$ represents the interval of missing data.

Let b_{km*} be a small-interval data, replace the lost data b_{km} in the big data set with b_{km*}, and at the same time, fill the U_k data set in the original big data.

$$f\left(u_{|U*|}, b_{km*}\right) = \left[f\left(u_{|U*|}, b_{km*}\right), f\left(u_{|U*|}, b_{km*}\right)\right],$$
$$\left|f\left(u_{|U*|}, b_{km*}\right)\right| = f^S\left(u_{|U*|}, b_{km*}\right)$$
$$-f^X\left(u_{|U*|}, b_{km*}\right) = \delta, \tag{3}$$

where S represents the lower limit of the interval and X represents the upper limit of the interval.

The newly obtained large data set can be expressed according to

$$W^* = (U^*, A^*). \tag{4}$$

The information entropy of the newly obtained data set can be calculated by

$$H_\lambda^N(A^*)|_U = \sum_{i=1}^{|U^*|} \frac{1}{|U^*|} \times \lg\left(\frac{1}{|U^*|}\sum_{j=1}^{|U^*|}\sum_{i=1}^{N} \lambda_l(k) P_{uiuj}^{bl}\right). \tag{5}$$

b_{km} indicates that the missing data in the large data set are small-interval data, so $\sum_{i=1}^{|U^*|} P_{uiuj}^{bl}$ is almost equal to zero, and $P_{uiuj}^{bl} = 1$. That is, $H_\lambda^N(A^*)$ is

$$H_\lambda^N(A^*) = \lg|U^*| - \frac{1}{|U^*|}\left[\sum_{l=1}^{N} \lambda_l(k)\right. \tag{6}$$

The m data d_{hm} in the original data U_k is missing data, and the length can be regarded as zero, while the original big data are $\lambda_m(k) P_{ujuk}^{b_{km}} = 0$. It can be confirmed that the big data information entropy and the original data information entropy after filling U_k meet the following relationship:

$$H_\lambda^N(A^*)|_0 > H_\lambda^N(A)|_0. \tag{7}$$

By using relatively small-interval data to replace missing data and continuously expanding the interval range, the information entropy in certain ranges will continue to decrease [37–40]. The similarity relationship of the above-mentioned use interval expands the range of the interval. The greater the position relative to the newly filled data interval, the greater the possibility that the object will be classified into other object data sets [41, 42]. When the filling range of the data interval is too large, the data classification is abnormal, resulting in data confusion. When the newly filled interval range increases from the minimum to the maximum, there will be at least one entropy representing the minimum [43–47].

2.2. Determination of Density Peak Clustering Center. According to the calculated information entropy of filling big data δ_i and ρ_i, the relationship between them is analyzed and the basis is put forward. The density peak clustering algorithm is optimized by using the clustering center selection strategy [48, 49]. According to the principle of cluster center selection, the difference degree of cluster points is

measured by using normalized product of adjacent distance δ_i and density ρ_i. According to the statistical characteristics and change trend of the difference degree, the largest group of points is selected as the cluster center. After obtaining the cluster center, the network big data are divided into different clusters according to the adjacent distance label so as to realize clustering [50–52].

In order to quantify the degree to which a data point of network big data is offset from the origin δ and ρ after normalization, the cluster center weight is introduced according to the positive proportional relationship:

$$\omega_i = \delta_i \rho_i. \tag{8}$$

In order to obtain the data point set with the largest deviation, the cluster center weights are arranged in the order from large to small. The first N points are taken, and N is usually set to 30. Take the point with the greatest deviation from the origin as the inflection point of the overall downward trend of the cluster center weight from acute to slow.

The downward trend of the weight of the survey center is described by the slope of the two-point line segment:

$$k_i^N = \frac{\delta_{i+N} - \delta_i}{N}. \tag{9}$$

In equation (9), k_i^N denotes the average change rate of cluster center weight within $[i, i+N]$ range, which reflects overall change trend of a certain range ψ. Then, the inflection point can be described as

$$v = \arg\left[\max\left(\frac{k_i^1}{k_1^{i-1}}\right)\right]. \tag{10}$$

In equation (10), k_1^{i-1} represents the slope from the first point to the i th point, which is the average change rate of point set $\{1, 2, \ldots, i\}$; k_i^l is used to describe the slope from the i th point to the $i+1$ st point.

Based on the above analysis, the cluster center selection process is given:

(1) Calculate the difference degree of each network data point ψ.

(2) The cluster center weights are arranged in order from large to small.

(3) Calculate k_i^l and k_1^{i-1} as well as the maximum value of k_i^1/k_1^{i-1}, and determine the inflection point $i = v$.

(4) Take the network data point $\{1, 2, \ldots, v\}$ before the inflection point as the cluster center point.

2.3. Unknown Block Calculation Method. Based on the understanding of the data set, the correlation missing data can be divided into two types: block known and block unknown. For known missing data, it can be directly divided into blocks by known information. In this case, there will be fewer variables, and the meaning of the variables is clear. In this paper, several data sets are used for experimental verification.

In actual operation, most of the missing data sets are unknown blocks, especially in cloud computing with missing big data, which makes it difficult to distinguish block information. In the case that the block is unknown, this paper adopts the density peak clustering algorithm and uses the method of improving the distance measurement to achieve the block of missing data.

The density peak clustering algorithm does not need to specify the number of categories in advance. The calculation method in the data set is as follows:

Input: the number of clusters is A, and the data set is K.

Output: $\alpha^{(i)}$ clusters of all objects.

Step 1: randomly select K cluster centers, denoted by $u_1, u_2, \ldots, u_k \in R^P$.

Step 2: iterate until convergence.

Calculate the cluster belonging to all objects $\alpha^{(i)}$ in A:

$$c^{(i)} = \arg\min_j \left\|\alpha^{(i)} - u_j\right\|^2. \tag{11}$$

Update the center of the class based on all the classes j:

$$u_j = \frac{\sum_{i=1}^n 1\left\{c^{(i)} = j\right\} a^{(i)}}{\sum_{i=1}^n 1\left\{c^{(i)} = j\right\}}. \tag{12}$$

Before describing the block calculation, the following problems should be solved:

(1) The clustering method can be divided into Q-type and R-type clustering. The K-means calculation method is actually the Q-type clustering method. R-type refers to relative variable clustering. To cluster the variables using the density peak clustering algorithm, the data set should be transferred to $A^T \in R^{p \times n}$ first and then to cluster A'.

(2) Since the block processing effect is poor, when the sample size of n is relatively small, A^T after transposition will be an uncertain data set and density peak clustering algorithm has no limitations in clustering. However, when n is relatively large, A^T is still a missing data after replacement, and the density peak clustering algorithm has certain limitations in clustering. In order to solve this problem, the sparse expression is used to select variables. The central idea is to constrain the weights of variables in the objective function, forcing variables with relatively small weights to not cluster, thereby retaining variables with large weights. In this way, the selection of variables is realized, and the objective function of the result is defined as follows:

$$\begin{aligned} c^{(i)} &= \arg\min_j \sum_{u=1}^p w_u \left\|\alpha_u^{(i)} - u_{ju}\right\|^2, \\ \text{s.t. } &\|w\|^2 \le 1, \quad \|w\|_1 \le s, w_u \ge 0 \forall_u. \end{aligned} \tag{13}$$

where w represents the variable weight vector, w_u represents the coefficient of the u variable weight, and s represents the adjusted parameter.

For the method mentioned above, selection of variables is adopted to solve the limitation of the density peak clustering algorithm when the number is large.

(3) Compared with the classical clustering algorithm, the Euclidean distance is usually used to calculate the distance. When a large amount of data is missing, it is difficult to calculate the Euclidean distance. Therefore, it is necessary to define the distance between the missing objects $\alpha^{(i)}$ and $\alpha^{(j)}$:

$$d\left(\alpha^{(i)}, \alpha^{(j)}\right) = \frac{n}{\sum_{u=1}^{n} I_u} \sqrt{\sum_{u=1}^{n} \left(\alpha_u^{(i)}, \alpha_u^{(j)}\right)^2 I_u}. \tag{14}$$

were $\alpha_u^{(j)}$ represents the value of the object $\alpha^{(i)}$ on the u variable.

(4) For classical clustering methods, arithmetic averaging is usually used to update the cluster centers, but this method is not applicable when the data are missing. Therefore, the paper proposes the clustering centers in the case of missing data. The s object $\{\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(s)}\}$ in the j cluster includes the missing part; that is, the cluster center u_j is taken in the i variable. The formula is

$$u_{ji} = \frac{s}{\sum_{u=1}^{s} I_{iu}} \sum_{u=1}^{s} \alpha_i^{(u)} I_{iu}, \tag{15}$$

where $I_{iu} = \begin{cases} 0, & \alpha_i^{(u)} = * \\ 1, & \alpha_i^{(u)} \neq * \end{cases}$, which is $u_j = [u_{j1}, u_{j2}, \ldots, u_{jp}]$.

After solving the above problems, the calculation method of KMB is given below:

Input: data set A, number of clusters K.

Output: block data set.

Step 1: transpose A to obtain A^T.

Step 2: arbitrarily select K cluster center points $u_1, u_2, u, \ldots, u_K \in R^n$.

Step 3: iterate until convergence.

Calculate the cluster belonging to all objects $\alpha^{(i)}$ in A^T:

$$c^{(i)} = \arg\min_j \frac{n^2 \sum_{u=1}^{n} \left(\alpha_u^{(i)} - u_{ju}\right)^2 I_u}{\left(\sum_{u=1}^{n} I_u^2\right)^2}. \tag{16}$$

Relative to all classes j, update the center of the cluster according to the definition in the text:

$$u_j = \left[u_{j1}, u_{j2}, \ldots, u_{jp}\right]. \tag{17}$$

Step 4: convert the cluster to complete the data set and obtain A_1.

Step 5: segment the data set A_1 according to the clustering result to obtain a block data set.

Block filling is a missing data filling method obtained by dividing the data set into blocks according to the characteristics of the data set. It is suitable for a wide range of data filling and relying on other variables, which is a big advantage. It is also called the host method that is widely used. Most of the traditional filling can be used in the host algorithm proposed in the article, except for mode filling and mean filling. However, some filling algorithms do not rely on variables, which meet the conditions as follows:

Input: Missing data set $S = (U, A, V, f)$.

Output: Complete data set $S' = (U', A, V, f')$.

Step 1: Determine the data block. If the block is known, then it can be directly divided; if it is unknown, it needs to be divided by the KMB algorithm.

Step 2: Through the block information. Segmentation of the missing data set S yields K sub-data sets $\mathrm{RS}_i = (U, A_i, V, f)$ and $i = 1, 2, \ldots, K$.

Step 3: Compared with all missing data sets, the complete sub-data set $S_i = (U', A_i, V, f')$, $i = 1, 2, \ldots, K$ can be obtained by using host filling.

Step 4: Combine all the complete sub-data sets S_i to obtain a complete data set $S' = (U', A, V, f')$.

To block the initial data set and fill all blocks in parallel, the calculation time for filling should be reduced to $\max(t_1, t_2, \ldots, t_k)$, where K represents the number of blocks and t_k represents the filling time of K blocks. When the data set dimensions and data volume are large, the block filling effect is obvious.

3. Experimental Study

This experiment uses the movie rating data information obtained from the Movie Lens data set, randomly selects the rating data of 1500 users for 3000 movies, and converts it into a 3000×1500 partial observable user-movie rating data matrix X. Given three cases of rank $r = 10, 500, 1000$, the proportion of observable items is set to rate = 0.1, 0.3, 0.5, 0.8 in the three cases. Comparative analysis is conducted in terms of the iteration convergence time time, the number of iterations n, and the relative error $\varepsilon = \|X - M\|_F^2 / \|M\|_F^2$ between the repair matrix and the original matrix. The results are shown in Figure 1.

It can be seen from Figure 1 that when the rank of the matrix to be processed is 10, with the increase of observable items, the relative convergence errors of the four filling algorithms all have a downward trend in a certain period of time. That is to say, when there are more observable items, the accuracy of the algorithm convergence is higher, and the singular value threshold truncation algorithm and the algorithm proposed in this paper have higher accuracy of convergence than the other two algorithms. It can be seen from Figure 1(b) that when the rank is 500, with the increase of observable items, the relative error accuracy of the proposed matrix filling algorithm is better than other algorithms in a certain period of time. As an improved method of accelerating the nearest neighbor gradient algorithm, the augmented Lagrangian algorithm has better convergence rate and accuracy. It can be seen from Figure 1(c) that when the rank is 1000, under the condition of less observable items, the accuracy of augmented Lagrangian algorithm is better than other algorithms. In the case of many observable items, the error convergence accuracy of the proposed

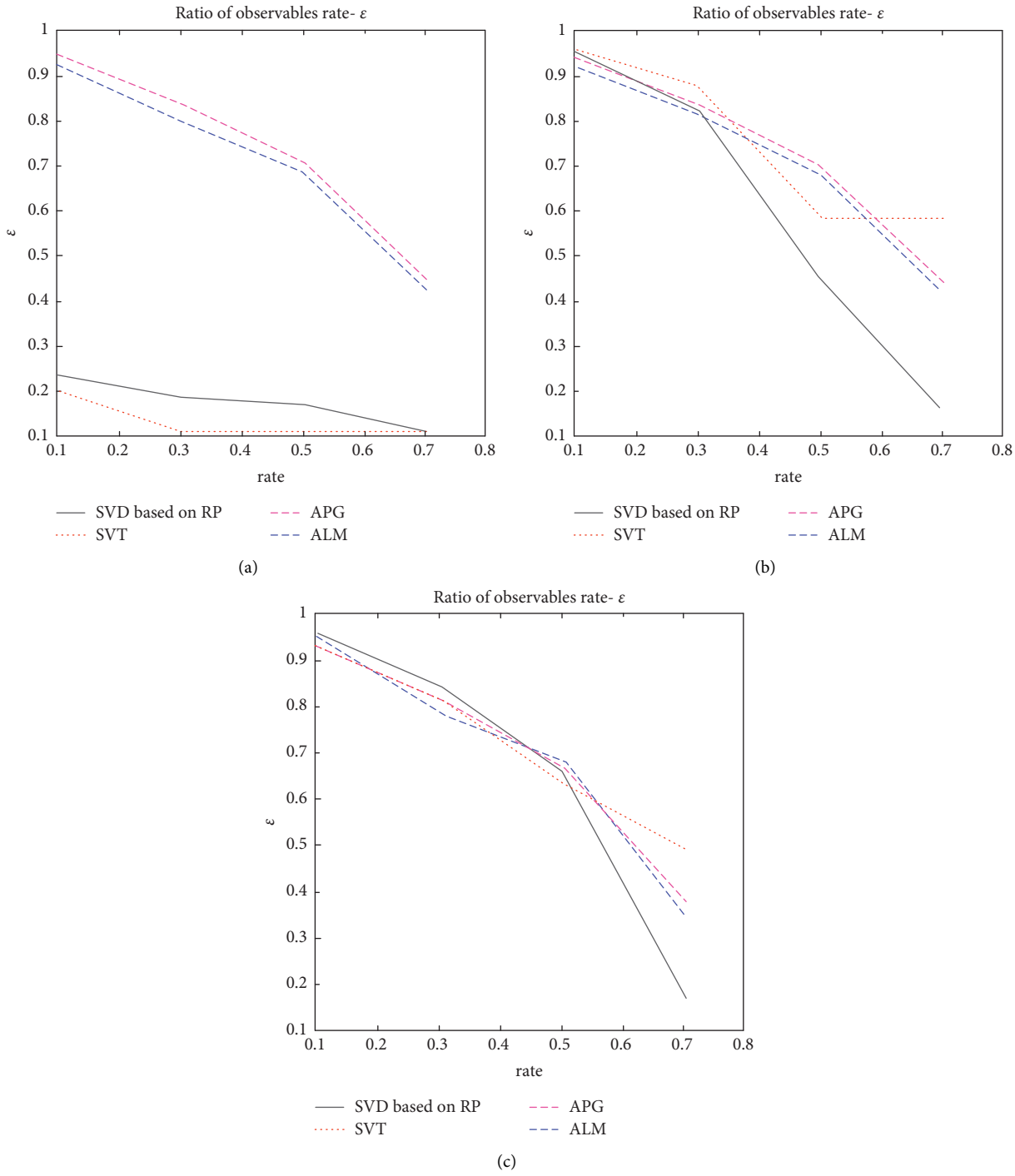

Figure 1: Line graph of the relationship between the ratio of observable items and the convergence error. (a) $r = 10$. (b) $r = 500$. (c) $r = 1000$.

matrix filling algorithm is significantly better than other algorithms.

3.1. d_2 Comparison of Filling Accuracy. The filling accuracies of different methods are measured by taking two standards as indicators. One standard is d_2, which is used to measure the degree of matching between the real value and the filling value.

$$d_2 = 1 - \left[\frac{\sum_{i=1}^{n} (e_i - r_i)^2}{\sum_{i=1}^{n} \left(|e_i - E| + |r_i - R|\right)^2}\right]. \tag{18}$$

According to Table 1, it can be seen that for any combination of missing, the proposed algorithm is obviously higher than the other two algorithms. In addition, the more the missing data of correlation, the lower the d_2 obtained by the other two methods. The filling accuracy will decrease

Table 1: d_2 filling accuracy index.

Combination		Algorithm		
Missing rate/%	Missing pattern	Proposed algorithm	FIMUS	DMI
1	Single	0.837	0.748	0.734
	Multiple	0.819	0.729	0.723
3	Single	0.895	0.729	0.714
	Multiple	0.846	0.707	0.698
5	Single	0.853	0.694	0.683
	Multiple	0.843	0.684	0.674
10	Single	0.867	0.659	0.646
	Multiple	0.846	0.638	0.618

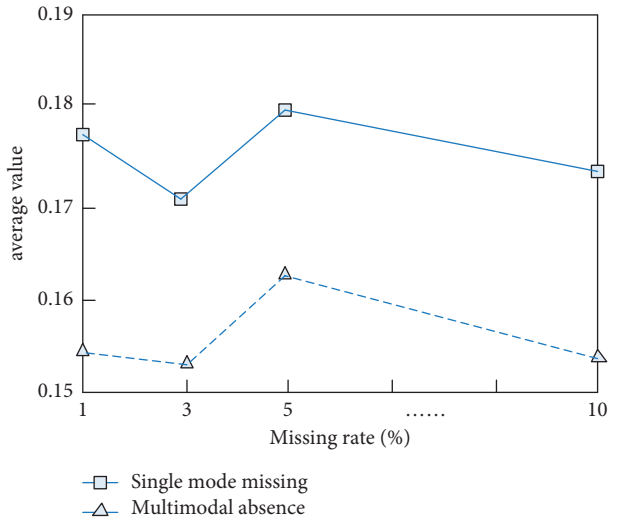

Figure 2: RMSE average.

with the missing rate of the data. However, the filling accuracy of the proposed algorithm has always been maintained at a very high level. In terms of d_2, the proposed filling method is obviously higher than the other two algorithms as well.

3.2. RMSE Mean Analysis. The average error between the filled value and the true value is measured according to

$$\mathrm{RMSE} = \left(\frac{1}{n} \sum_{i=1}^{n} \left| r_i - e_i \right|^2 \right)^{1/2}. \tag{19}$$

where n is the number of missing values, r_i is the true value of the i missing value, e_i is the filled value of the i missing value, R is the average value of r_i, and E is the average value of e_i, $i = 1, 2, \ldots, n$, the meaning of the two standards. The larger the value of d_2, the higher the filling accuracy. Conversely, the smaller the RMSE value, the higher the filling accuracy.

As can be seen from Figure 2, the filling accuracy of the proposed method is relatively stable, the data missing will be between 2% and 12%, the value will be above 0.8, and the RMSE value will be between 0.15 and 0.2. Compared with the single correlation missing, the filling accuracy of single missing mode will be significantly higher than that of multi-filling mode. Since the missing data of multi-fill pattern are relatively large, the interference of feature extraction and restoration is higher than that of single fill pattern. This proves that the proposed method has stronger stability and higher filling accuracy than the other two methods.

Table 2: High rank matrix filling accuracy of density peak clustering method.

Data volume (GB)	High rank matrix filling accuracy (%)		
	Google Dataset Search	Google Trends	EU Open Data Portal
50	92.65	96.28	98.32
100	97.26	99.32	99.01
150	98.93	98.71	99.85
200	97.61	99.63	96.16

Table 3: Cuckoo algorithm optimization K high rank matrix filling accuracy of means clustering method.

Data volume (GB)	High rank matrix filling accuracy (%)		
	Google Dataset Search	Google Trends	EU Open Data Portal
50	53.15	58.88	53.63
100	57.23	60.12	62.10
150	60.02	56.26	56.09
200	55.96	53.17	52.61

3.3. Analysis of Filling Effect of High Rank Matrix under Multiple Data Sets. In order to further verify the filling effect of high rank matrix of network big data, filling accuracy of density peak clustering method, means clustering method and improved neural process method are comparatively analyzed based on Google dataset search dataset (https://toolbox.google.com/datasetsearch), Google Trends dataset (https://trends.google.com/trends/explore), and EU open data portal dataset (https://data.europa.eu/euodp/en/data/), as shown in Tables 2–4.

According to Tables 2–4, the highest filling accuracy of means clustering method under Google dataset search, Google Trends, and EU open data portal datasets is 60.02%, 60.12%, and 62.10%, respectively. In contrast, the highest filling accuracy of the improved neural process method under Google dataset search, Google Trends, and EU open

Table 4: High rank matrix filling accuracy of improved neural process method.

Data volume (GB)	High rank matrix filling accuracy (%)		
	Google Dataset Search	Google Trends	EU Open Data Portal
50	53.26	49.82	49.32
100	55.12	50.33	59.16
150	60.98	56.29	48.22
200	59.90	58.36	66.10

data portal datasets is 60.98%, 56.29%, and 66.10%, respectively. The highest filling accuracy of the improved neural process method under Google Dataset Search, Google Trends, and EU Open Data Portal data sets is 60.98%, 56.29%, and 66.10%, respectively. The highest filling accuracy of density peak clustering method under Google Dataset Search, Google Trends, and EU Open Data Portal data sets is 98.93%, 99.63, and 99.85%, respectively. The above data show that the density peak clustering method has higher filling accuracy as it uses small-interval data to replace the lost data. By optimizing the density peak clustering algorithm through the cluster center selection strategy, obtaining the block data set through the unknown block method, and realizing the block filling by using the host filling algorithm, the filling noise of the network big data matrix can be effectively avoided and the filling effect can be improved.

4. Conclusion

This paper presents a high rank matrix filling algorithm for network big data based on density peak clustering. The density peak clustering is introduced, the missing data are replaced by small-interval data, and the information entropy of filled data is calculated. Combined with the density peak clustering algorithm and the improved distance measurement in the case of missing data, the missing data are partitioned, and the host filling can be used to obtain a complete sub-data set. The following conclusions are drawn through experiments:

(1) When the rank is 500, with the increase of observable terms, the relative error accuracy of the matrix filling algorithm proposed in this paper is better than that of other algorithms. When there are many observable items, the error convergence accuracy of the proposed matrix filling algorithm is obviously better than that of other algorithms as well.

(2) For any missing combination, the proposed algorithm is obviously higher than the other two algorithms. In addition, in the case of more correlation missing data, the filling accuracy of the proposed method is always stable.

(3) The filling accuracy of the proposed method is relatively stable, with missing data ranging between 2% and 12%, $D2$ value more than 0.8, and the RMSE value between 0.15 and 0.2. The filling accuracy of single missing mode will be significantly higher than that of multi-filling mode, because the missing data of multi-filling mode are relatively large, and the interference to feature extraction and restoration is higher than that of single missing mode.

Data Availability

The data sets used and/or analyzed during the current study are available from the author on reasonable request.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] D. Belomestny and M. Trabs, "Low-rank diffusion matrix estimation for high-dimensional time-changed Lévy processes," *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, vol. 54, no. 3, pp. 1583–1621, 2018.

[2] I. Tsamardinos, G. Borboudakis, P. Katsogridakis, P. Pratikakis, and V. Christophides, "A greedy feature selection algorithm for big data of high dimensionality," *Machine Learning*, vol. 108, no. 2, pp. 149–202, 2018.

[3] Z. Sabir, C. M. Khalique, M. A. Z. Raja, and D. Baleanu, "Evolutionary computing for nonlinear singular boundary value problems using neural network, genetic algorithm and active-set algorithm," *The European Physical Journal Plus*, vol. 136, no. 2, p. 195, 2021.

[4] K. Nisar, Z. Sabir, M. A. Zahoor Raja et al., "Evolutionary integrated heuristic with gudermannian neural networks for second kind of lane–Emden nonlinear singular models," *Applied Sciences*, vol. 11, no. 11, p. 4725, 2021.

[5] X. Zhou, D. Yao, M. Zhu et al., "Vigilance detection method for high-speed rail using wireless wearable EEG collection technology based on low-rank matrix decomposition," *IET Intelligent Transport Systems*, vol. 12, no. 8, pp. 819–825, 2018.

[6] J. Yu, G. Zhou, and C. Li, "Low tensor-ring rank completion by parallel matrix factorization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 21, no. 4, pp. 1–14, 2020.

[7] B. Chen, Z. Yang, and Z. Yang, "An algorithm for low-rank matrix factorization and its applications," *Neurocomputing*, vol. 275, no. 31, pp. 1012–1020, 2018.

[8] Q. Jiang, F. Shao, W. Lin, K. Gu, G. Jiang, and H. Sun, "Optimizing multistage discriminative dictionaries for blind image quality assessment," *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 2035–2048, 2018.

[9] J. Zhang and G. Qu, "Physical unclonable function-based key sharing via machine learning for IoT security," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 7025–7033, 2020.

[10] K. H. Thung, P. T. Yap, E. Adeli, S. W. Lee, and D. Shen, "Conversion and time-to-conversion predictions of mild cognitive impairment using low-rank affinity pursuit denoising and matrix completion," *Medical Image Analysis*, vol. 45, no. 12, pp. 68–82, 2018.

[11] S. Yang, J. Wang, B. Deng, M. R. Azghadi, and B. Linares-Barranco, "Neuromorphic context-dependent learning framework with fault-tolerant spike routing," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[12] Q. Sun, K. Lin, C. Si, Y. Xu, S. Li, and P. Gope, "A secure and anonymous communicate scheme over the internet of things," *ACM Transactions on Sensor Networks*, vol. 18, no. 3, pp. 1–21, 2022.

[13] B. Cao, J. Zhao, X. Liu et al., "Multiobjective evolution of the explainable fuzzy rough neural network with gene expression programming," *Ieee Transactions on Fuzzy Systems*, p. 1, 2022.

[14] X. Liu, J. Zhao, J. Li, B. Cao, and Z. Lv, "Federated neural architecture search for medical data security," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5628–5636, 2022.

[15] B. Cao, M. Li, X. Liu, J. Zhao, W. Cao, and Z. Lv, "Many-Objective deployment optimization for a drone-assisted camera network," *IEEE transactions on network science and engineering*, vol. 8, no. 4, pp. 2756–2764, 2021.

[16] X. L. Sun, Y. Guo, N. Li, and X. X. Song, "Missing data filling algorithm based on improved neural process," *Journal of University of Chinese Academy of Sciences*, vol. 38, no. 02, pp. 280–287, 2021.

[17] F. Lin, Y. Zheng, L. Pan, and Z. Zuo, "Attenuation of noisy environment-induced neuroinflammation and dysfunction of learning and memory by minocycline during perioperative period in mice," *Brain Research Bulletin*, vol. 159, no. 06, pp. 16–24 + 30, 2020.

[18] B. Cao, Y. Zhang, J. Zhao, X. Liu, L. Skonieczny, and Z. Lv, "Recommendation based on large-scale many-objective optimization for the intelligent internet of things system," *IEEE Internet of Things Journal*, p. 1, 2021.

[19] B. Cao, J. Zhang, X. Liu et al., "Edge-Cloud resource scheduling in space-air-ground integrated networks for internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5765–5772, 2022.

[20] B. Cao, J. Zhao, Z. Lv, and P. Yang, "Diversified personalized recommendation optimization based on mobile data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2133–2139, 2021.

[21] B. Cao, W. Zhang, X. Wang, J. Zhao, Y. Gu, and Y. Zhang, "A memetic algorithm based on two_Arch2 for multi-depot heterogeneous-vehicle capacitated arc routing problem," *Swarm and Evolutionary Computation*, vol. 63, Article ID 100864, 2021.

[22] B. Cao, Z. Sun, J. Zhang, and Y. Gu, "Resource allocation in 5G IoV architecture based on SDN and fog-cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3832–3840, 2021.

[23] Y. Hu, X. Liu, and M. Jacob, "A generalized structured low-rank matrix completion algorithm for MR image recovery," *IEEE Transactions on Medical Imaging*, vol. 38, no. 8, pp. 1841–1851, 2019.

[24] I. Masliah, A. Abdelfattah, A. Haidar et al., "Algorithms and optimization techniques for high-performance matrix-matrix multiplications of very small matrices," *Parallel Computing*, vol. 81, no. 7, pp. 1–21, 2019.

[25] B. Cao, Y. Gu, Z. Lv, S. Yang, J. Zhao, and Y. Li, "RFID reader anticollision based on distributed parallel particle swarm optimization," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3099–3107, 2021.

[26] B. Cao, S. Fan, J. Zhao et al., "Large-Scale many-objective deployment optimization of edge servers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3841–3849, 2021.

[27] Z. Lv, S. Lv, H. Feng, and H. Lv, "Clinical characteristics and analysis of risk factors for disease progression of COVID-19: a retrospective Cohort Study," *International Journal of Biological Sciences*, vol. 17, pp. 1–7, 2021.

[28] B. Li, J. Yang, Y. Yang, C. Li, and Y. Zhang, "Sign language/gesture recognition based on cumulative distribution density features using UWB radar," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

[29] W. Deng, Y. Guo, J. Liu, Y. Li, D. Liu, and L. Zhu, "A missing power data filling method based on improved random forest algorithm," *Chinese Journal of Electrical Engineering*, vol. 5, no. 4, pp. 33–39, 2019.

[30] S. C. Chu, Y. Chen, F. Meng, C. Yang, J. S. Pan, and Z. Meng, "Internal search of the evolution matrix in QUasi-affine transformation evolution (QUATRE) algorithm," *Journal of Intelligent and Fuzzy Systems*, vol. 38, no. 5, pp. 5673–5684, 2020.

[31] R. Y. Zhao and W. J. Li, "Simulation of the influence of mobile communication delay on information parallel transmission efficiency," *Computer Simulation*, vol. 37, no. 4, pp. 192–195, 2020.

[32] M. Gao, L. Chen, B. Li, and W. Liu, "A link prediction algorithm based on low-rank matrix completion," *Applied Intelligence*, vol. 48, no. 12, pp. 4531–4550, 2018.

[33] G. Sun, Y. Cong, J. Dong, Y. Liu, Z. Ding, and H. Yu, "What and How: generalized lifelong spectral clustering via dual memory," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3895–3908, 2021.

[34] Y. He, L. Dai, and H. Zhang, "Multi-Branch deep residual learning for clustering and beamforming in user-centric network," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2221–2225, 2020.

[35] F. Liu, G. Zhang, and J. Lu, "Heterogeneous domain adaptation: an unsupervised approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5588–5602, 2020.

[36] W. Yang, X. Chen, Z. Xiong, Z. Xu, G. Liu, and X. Zhang, "A privacy-preserving aggregation scheme based on negative survey for vehicle fuel consumption data," *Information Sciences*, vol. 570, pp. 526–544, 2021.

[37] F. Bu, "An efficient fuzzy c-means approach based on canonical polyadic decomposition for clustering big data in IoT," *Future Generation Computer Systems*, vol. 88, no. 11, pp. 675–682, 2018.

[38] M. d'Errico, E. Facco, A. Laio, and A. Rodriguez, "Automatic topography of high-dimensional data sets by non-parametric density peak clustering," *Information Sciences*, vol. 560, pp. 476–492, 2021.

[39] B. Li, Y. Feng, Z. Xiong, W. Yang, and G. Liu, "Research on AI security enhanced encryption algorithm of autonomous IoT systems," *Information Sciences*, vol. 575, pp. 379–398, 2021.

[40] J. Mou, P. Duan, L. Gao, X. Liu, and J. Li, "An effective hybrid collaborative algorithm for energy-efficient distributed permutation flow-shop inverse scheduling," *Future Generation Computer Systems*, vol. 128, pp. 521–537, 2022.

[41] F. Gao, D. Yu, and Q. Sheng, "Analytical treatment of unsteady fluid flow of nonhomogeneous nanofluids among two infinite parallel surfaces: collocation method-based study," *Mathematics*, vol. 10, no. 9, p. 1556, 2022.

[42] D. Yu and R. Wang, "An optimal investigation of convective fluid flow suspended by carbon nanotubes and thermal radiation impact," *Mathematics*, vol. 10, no. 9, p. 1542, 2022.

[43] J. C. Fan and T. W. S. Chow, "Sparse subspace clustering for data with missing entries and high-rank matrix completion," *Neural Networks*, vol. 93, pp. 36–44, 2017.

[44] H. Yang, C. Zhang, G. Chen, and B. Zhao, "Big data-oriented intelligent control algorithm for marine communication transmission channel," *Journal of Coastal Research*, vol. 93, no. sp1, p. 741, 2019.

[45] Y. Y. Fu, "Face recognition using scalable constraints data fusion," *Computer Informatization and Mechanical System*, vol. 3, no. 1, pp. 120–122, 2020.

[46] K. Nisar, Z. Sabir, M. A. Zahoor Raja et al., "Design of morlet wavelet neural network for solving a class of singular pantograph nonlinear differential models," *IEEE Access*, vol. 9, pp. 77845–77862, 2021.

[47] M. Ali and J. Gao, "Classification of matrix-variate Fisher-Bingham distribution via maximum likelihood estimation using manifold valued data," *Neurocomputing*, vol. 295, no. 21, pp. 72–85, 2018.

[48] B. Eriksson, L. Balzano, and R. Nowak, "High-rank matrix completion and subspace clustering with missing data," 2011, https://arxiv.org/abs/1112.5629.

[49] C. Lane, R. Boger, C. You, M. Tsakiris, and R. Vidal, "Classifying and comparing approaches to subspace clustering with missing data," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 669–677, IEEE, Seoul, Korea (South), October 2019.

[50] M. Li, S. Chen, Y. Shen, G. Liu, I. W. Tsang, and Y. Zhang, "Online multi-agent forecasting with interpretable collaborative graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[51] Z. Chen, J. Tang, X. Y. Zhang, D. K. C. So, S. Jin, and K. K. Wong, "Hybrid evolutionary-based sparse channel estimation for IRS-Assisted mmWave MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1586–1601, 2022.

[52] Q. Meng, Q. Ma, and G. Zhou, "Adaptive output feedback control for stochastic uncertain nonlinear time-delay systems," *IEEE Transactions on Circuits and Systems. II, Express Briefs*, p. 1, 2022.