IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 7, 2020, accepted July 21, 2020, date of publication July 29, 2020, date of current version August 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012740

Fast Clustering by Affinity Propagation Based on Density Peaks

YANG LI[1], CHONGHUI GUO[1], AND LEILEI SUN[2], (Member, IEEE)
[1]Institute of Systems Engineering, Dalian University of Technology, Dalian 116024, China
[2]School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Corresponding author: Chonghui Guo (dlutguo@dlut.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 71771034 and Grant 71421001, in part by the Science and Technology Program of Jieyang under Grant 2017xm041, and in part by the Scientific and Technological Innovation Foundation of Dalian under Grant 2018J11CY009.

ABSTRACT Clustering is an important technique in data mining and knowledge discovery. Affinity propagation clustering (AP) and density peaks and distance-based clustering (DDC) are two significant clustering algorithms proposed in 2007 and 2014 respectively. The two clustering algorithms have simple and clear design ideas, and are effective in finding meaningful clustering solutions. They have been widely used in various applications successfully. However, a key disadvantage of AP is its high time complexity, which has become a bottleneck when applying AP for large-scale problems. The core idea of DDC is to construct the decision graph based on the local density and the distance of each data point, and then select the cluster centers, but the selection of the cluster centers is relatively subjective, and sometimes it is difficult to determine a suitable number of cluster centers. Here, we propose a two-stage clustering algorithm, called DDAP, to overcome these shortcomings. First, we select a small number of potential exemplars based on the two quantities of each data point in DDC to greatly compress the scale of the similarity matrix. Then we implement message-passing on the incomplete similarity matrix. In experiments, two synthetic datasets, nine publicly available datasets, and a real-world electronic medical records (EMRs) dataset are used to evaluate the proposed method. The results demonstrate that DDAP can achieve comparable clustering performance with the original AP algorithm, while the computational efficiency improves observably.

INDEX TERMS Exemplar-based clustering, affinity propagation, density peaks.

I. INTRODUCTION

Clustering is a discovery process that groups a set of data such that the intracluster similarity is maximized and the intercluster similarity is minimized [1]. Generally clustering is used for two aims: (a) receiving a primary understanding of raw data and (b) reducing the size of a huge amount of raw data [2]. Because of the importance of clustering, a large number of clustering algorithms have been proposed and applied widely in many domains [3], [4]. Affinity propagation clustering (AP) [5] and density peaks and distance-based clustering (DDC) [6] are two significant clustering algorithms proposed in 2007 and 2014 respectively. The implementation of an exemplar-based clustering is to find some representative data points called exemplars as centers and assign the remaining data points to their nearest centers [7]. How to determine the K appropriate exemplars from N data points is the key to the exemplar-based clustering method, and it is an NP-hard combinatorial optimization problem. However, AP and DDC provide new methods for solving the problem. They have been successfully used in many practical problems.

Different from traditional iterative algorithms, such as k-means and k-centers, AP does not need to choose K initial exemplars. The core idea of AP is to regard all the data points as potential exemplars. Only the pairwise similarities of data points are needed. Messages are exchanged between data points until a high-quality set of exemplars emerges. The number of exemplars is automatically generated. Due to its effectiveness and simplicity, AP has been widely used [8]–[10]. Meanwhile, a lot of variants of the AP algorithm have been proposed (e.g., Semi-supervised AP [11]; K-AP [12]; Hierarchical AP [13]; Incremental AP [14]). The AP algorithm uses the greedy strategy, which maximizes the value of the global function of the clustering network during

The associate editor coordinating the review of this manuscript and approving it for publication was Mamoun Alazab.

every iteration. In each iteration, some $N \times N$ matrices are used, so the time complexity of AP is $O(N^2T)$, where N is the number of data points and T is the number of iterations. Because of the high computational complexity, AP clustering has rarely been used in large-scale clustering problems.

DDC selects exemplars as centers at one time based on the assumptions that cluster centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any points with a higher local density. The framework of DDC is to determine, based on the distance function, the local density of each data point and the shortest distance among each data point and other data points with higher local density are computed to construct the decision graph first, then the cluster centers based on the decision graph are selected, and finally, the remaining data points are placed into the nearest cluster with higher local density. As the two indicators are easy to compute and effective, DDC has been widely used [15]–[17]. However, in most cases, the centers of the data set are not very obvious. DDC has relatively strong subjectivity for the selection of the cluster centers based on the decision graph and the clustering result sensitive to the parameters involved in the DDC algorithm. The assignment strategy for the remaining points can create a domino effect, that is, once one data point is assigned erroneously, more data points may subsequently be assigned erroneously [18].

As the time complexity of AP is high and the selection of the cluster centers in DDC is relatively subjective, the aim of this paper is to develop a clustering algorithm to overcome these shortcomings. We propose a two-stage fast AP clustering algorithm DDAP, which can largely improve the efficiency of the AP algorithm while achieving comparable clustering performance with the original AP. First, we select a small number of potential exemplars based on the two quantities of each data point in DDC to greatly compress the scale of the similarity matrix. Then, we implement message-passing on the incomplete similarity matrix. Fig.1 shows the flow diagram of the proposed algorithm. In experiments, two synthetic datasets, nine publicly available datasets, and a real-world electronic medical records (EMRs) dataset are used to evaluate the proposed method. The results demonstrate that DDAP can achieve comparable clustering performances with the original AP algorithm, while the computational efficiency improves observably.

The rest of this paper is organized as follows. We briefly review the related work in Section 2. In Section 3, we detail the design ideas and propose the DDAP clustering algorithm. Experimental results on different kinds of data sets are presented in Section 4 to demonstrate the effectiveness and efficiency of the proposed algorithm. Conclusions are provided in Section 5.

II. RELATED WORK

In this section, we briefly introduce the related work about exemplar-based clustering. Then we introduce the basic ideas and formulas of AP and DDC, and some works about improving the efficiency of AP.

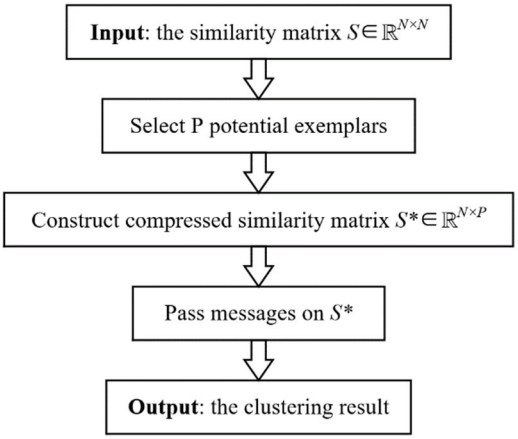

FIGURE 1. Flow diagram of the proposed algorithm.

The goal of exemplar-based clustering is to find an exemplar set that the sum of similarities between each data point and its exemplar is maximized. How to determine the K appropriate exemplars from N data points is the key to the exemplar-based clustering method, and it is an NP-hard problem [19]. Widely used algorithms for solving this problem are k-means [20], k-centers [7], and some variants, which are implemented in an iterative relocation manner. K-centers uses ''medoids'' instead of ''centroids'', which makes it more robust to noise and outliers. K-centers starts with an initial exemplar set and then refines the exemplar set along the gradient descent direction. The exemplar set found by k-centers is usually a local optimal solution. AP clustering considers all the data points as potential exemplars, and it does not need to choose K initial exemplars in advance. More importantly, a large number of experiments validate that the exemplar set found by AP is usually superior to k-centers.

A. CLUSTERING BY AFFINITY PROPAGATION

Frey and Dueck [5] proposed the standard AP algorithm as a clustering algorithm by propagating messages between a pair of data points. There are two kinds of message exchanges between data points, and each considers a different kind of competition. The responsibility $r(i, k)$ which expresses the support of data point i to candidate exemplar point k, and the availability $a(i, k)$ which expresses the appropriateness of data point k as the exemplar of data point i . The main procedures of the standard AP algorithm include updating $r(i, k)$ and $a(i, k)$ iteratively until convergence, i.e.,

$$r(i,k) \longleftarrow s(i,k) - \max_{k's.t.k'\neq k}\{a(i,k') + s(i,k')\} \tag{1}$$

$$a(i,k) \longleftarrow \begin{cases} \min\{0, r(k,k) \\ \quad + \sum\limits_{i's.t.i'\neq\{i,k\}} \max\{0, r(i',k)\}\}, & i \neq k \\ \sum\limits_{i's.t.i'\neq\{i,k\}} \max\{0, r(r',k)\}, & i = l \end{cases} \tag{2}$$

where $s(i, k)$ is the similarity between data point i and data point k. The exemplar of data point i can be obtained by

$$e(i) = \arg\max_k \{r(i, k) + a(i, k)\} \tag{3}$$

In the standard AP algorithm, the number of clusters is not a prerequisite. It takes the pairwise similarities between data points as input, so that it can deal with unstructured data. Thus, the algorithm is suitable for a wide variety of applications. However, the time computational complexity of AP is high, and it is not suitable for large-scale clustering problems.

According to the AP design principle, eliminating unnecessary message exchanges to make the algorithm converge quickly is a reasonable method for improving the efficiency of AP. Based on this idea, many fast sparse AP clustering algorithms have been proposed [21]–[24]. Fujiwara *et al.* [21] estimated the upper and lower bounds of exchanged messages between each object pair and then constructed a sparse factor graph according to these bounds. Jia *et al.* [22] and Xiao *et al.* [23] constructed a sparse factor graph using the k-nearest neighbors (k-NN) technique. To solve the problem that the application of k-NN unexpectedly results in too many fragments, which leads to too many exemplars in a further step, Jia *et al.* [22] added some edges to the original sparse factor graph and then implemented message propagation again on the new sparse graph, Xiao *et al.* [23] only implemented message propagation again between exemplars obtained by the k-NN sparse graph to obtain the final clustering result. Sun *et al.* [24] used a center-based agglomerative clustering method to obtain a set of potential exemplars to compress the similarity matrix first and then further reduced the new similarity matrix by sparseness according to k-nearest neighbors.

B. CLUSTERING BY DENSITY PEAKS AND DISTANCE

Rodriguez and Laio [6] proposed a clustering method based on density peaks and distance. This method is based on the assumption that cluster centers have relatively high local density and that they are at a relatively large distance from any point with a higher local density. For each data point i two indicators are computed: its local density ρ_i and its distance δ_i from points of higher density. Both these quantities depend only on the distances d_{ik} between data points. The local density ρ_i of data point i is defined as

$$\rho_i = \sum_k \chi(d_{ik} - d_c) \tag{4}$$

where $\chi(x) = 1$ if $x < 0$, $\chi(x) = 0$ and otherwise, and d_c is a cutoff distance. Then, the minimum distance between data point i and any other data point with higher density can be found by

$$\delta_i = \begin{cases} \max\limits_k(d_{ik}) & if\ \rho_i = \max\limits_j(\rho_i) \\ \max\limits_{k:\rho_k > \rho_i}(d_{ik}) & otherwise \end{cases} \tag{5}$$

Those data points with both relatively large ρ_i and δ_i are chosen as centers of clusters. The other data points should be assigned to the same cluster as its nearest neighbor of higher density.

Due to the good property of density peaks, some researchers have used it to improve the AP clustering algorithm. In view of the unsatisfactory clustering effect of the AP clustering algorithm when dealing with datasets of complex structures, Wang *et al.* [25] used an algorithm of density peaks which have an advantage in the manifold clustering with the idea of semi-supervised, built pairwise constraints to adjust the similarity matrix, and then executed the AP clustering on the new similarity matrix. The original AP used the Euclidean distance of the data sample as the only standard for similarity calculation. This method of calculation has considerable limitations for data with high dimensionality and sparsity, and the convergence and clustering accuracy of the algorithm are greatly affected. Wang *et al.* [26] constructed the density property to calculate the similarity and proposed AP clustering based on gravity (GAP). The proposed algorithm is more accurate in calculating the similarity of simple points through the local density of corresponding points and then the gravity formula is used to update the similarity matrix. In this paper, we focus on how to improve the computational efficiency of the AP clustering algorithm with the two indicators in DDC.

III. THE DDAP CLUSTERING ALGORITHM

A. DESIGN IDEAS

Both AP and DDC have been successfully used many practical problems. In AP, the number of exemplars is automatically generated, but the time complexity is high. DDC can easily select exemplars at one time, but the number of selected exemplars is determined to be relatively subjective. Therefore, we propose an algorithm that combines their strengths while overcoming their shortcomings.

We know that reducing unnecessary message propagation can improve AP algorithm efficiency. The standard AP algorithm simultaneously considers all data points as potential exemplars and propagates messages between each pair of data points. When the number of data points is large, the computation of the message propagating process is very huge. An intuitive idea is to find a small representative potential exemplar set in advance, in which each of these data points is likely to be a final exemplar, then implement message propagation based on these potential exemplars. The motivation of the previous studies is inspired by the fact that a data point plays an important role in the selection of exemplars near to it, but has nothing to do with exemplars far away from it. Therefore, message exchanges between distant objects can be omitted. However, this idea only considers local similarity information rather than both local and global similarity information, which results in a fact that the distribution of potential exemplars cannot well represent the distribution of the whole data set, then a significant decline in clustering performance appears [24]. To overcome this disadvantage,

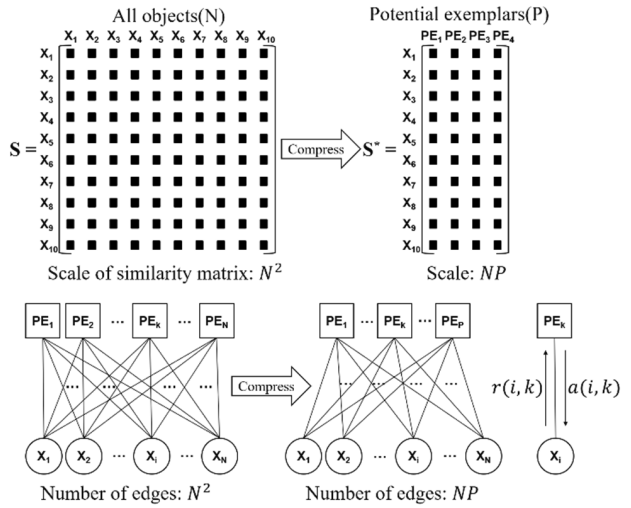

FIGURE 2. Schematic of DDAP. The compression of the similarity matrix and the corresponding change in the message propagation. $\mathbf{X}=\{\mathbf{X_1},\mathbf{X_2},\ldots,\mathbf{X_N}\}$ is the set of *N* data points and $\mathbf{PE}=\{\mathbf{PE_1},\mathbf{PE_2},\ldots,\mathbf{PE_P}\}$ is the set of *P* potential exemplars.

we need to design a new method to find a representative potential exemplar set to ensure that it contains most possible exemplars and can reflect the distribution of the whole data set. By doing this, the final exemplar set is guaranteed to be not too far away from the real optimal exemplar set.

Cluster centers have relatively high local density and are at a relatively large distance from any point with a higher local density. In DDC, those data points with both relatively large ρ_i and δ_i are chosen as cluster cores. The two indicators of data points are easy to calculate and perform well.

Inspired by these, we can select a small representative potential exemplar set based on the two indicators to compress the similarity matrix. We preserve the similarities of each data point with the selected potential exemplars. Therefore, the scale of the similarity matrix is reduced from N^2 to NP, where N is the number of data points and P is the number of selected potential exemplars. The messages are only exchanged between data points and the selected potential exemplars. A large number of unnecessary message exchanges are omitted. The schematic of the compression phase in the DDAP algorithm is shown in Fig.2.

B. THE DDAP ALGORITHM

For each data point i, we calculate its local density ρ_i and its distance δ_i from points of higher density. We combine the two indicators to measure the representativeness of the data point to become a potential exemplar. The product γ_i is

$$\gamma_i = \rho_i * \delta_i \tag{6}$$

According to DDC, the data points with both relatively large ρ_i and δ_i are chosen as cluster centers, so the γ_i of the center is relatively large. We choose the top P data points of the product γ_i as the representative potential exemplars, and preserve the similarities of each data point with the selected P potential exemplars.

Then, we implement message propagation based on these potential exemplars. When the design strategy is applied, the equations for responsibility and availability calculations are transformed as follows:

$$r(i,k) \leftarrow s(i,k) - \max_{k's.t.k'\neq k\cap k'\in PE}\{a(i,k')+s(i,k')\} \tag{7}$$

$$a(i,k) \leftarrow \begin{cases} \min\{0, r(k,k) \\ \quad + \sum\limits_{i's.t.i'\notin\{i,k\}\cap i'\in PE} \max\{0, r(i',k)\}\}, & i\neq k \\ \sum\limits_{i's.t.i'\neq k\cap i'\in PE} \max\{0, r(r',k)\}, & i=k \end{cases} \tag{8}$$

where PE is the representative potential exemplar set. The equation for selecting the exemplar of data point i is now given as

$$e(i) = \arg\max_{k\in PE}\{r(i,k)+a(i,k)\} \tag{9}$$

Algorithm DDAP presents a full description of the proposed algorithm. Fig.3 is a toy example to illustrate the differences between the standard AP and the proposed DDAP. The 25 two-dimensional data points were used by Frey and Dueck in [5], using Euclidean distance as the similarity.

The standard AP clustering algorithm considers all data points as potential exemplars and propagates messages between each pair of data points. In each iteration, $N \times N$ responsibilities and availabilities need to be computed, so the time complexity of AP is $O(N^2T)$, where T is the number of iterations. In DDAP, the similarity matrix is dramatically compressed and messages propagate between data points and the P potential exemplars. Only $N \times P$ responsibilities and availabilities need to be computed in each iteration. Therefore, the time complexity of DDAP in Step 4 and Step 5 is $O(NPT)$, where $P \ll N$, and T is the number of iterations. Computing the two indicators in Step 1 increases certain time cost, which requires comparing two real values N^2 times. However, this time cost is much less than the time cost of computing responsibilities and availabilities (for example, N sum operations and $N-1$ comparison operations are required for computing one responsibility value). Therefore, the overall time complexity of DDAP is nearly $O(NPT)$ as the time cost in Step 1 can be ignored compared with Steps 4 and 5. By compressing the similarity matrix and propagating messages based on a high-quality potential exemplar set, the complexity of AP clustering is reduced from quadratic complexity to linear complexity.

IV. EXPERIMENTS

In this section, we evaluate our method on both synthetic data and real-world data. The real-world data contain 9 datasets from the UCI machine learning repository and an electronic medical records (EMRs) dataset. The experiments are conducted on a personal computer with an Intel i7-6700 CPU, 8G RAM, Windows 10 64bit OS, and MATLAB 2016 programming environment. We compare the proposed DDAP with k-centers clustering, DDC, the standard AP clustering and

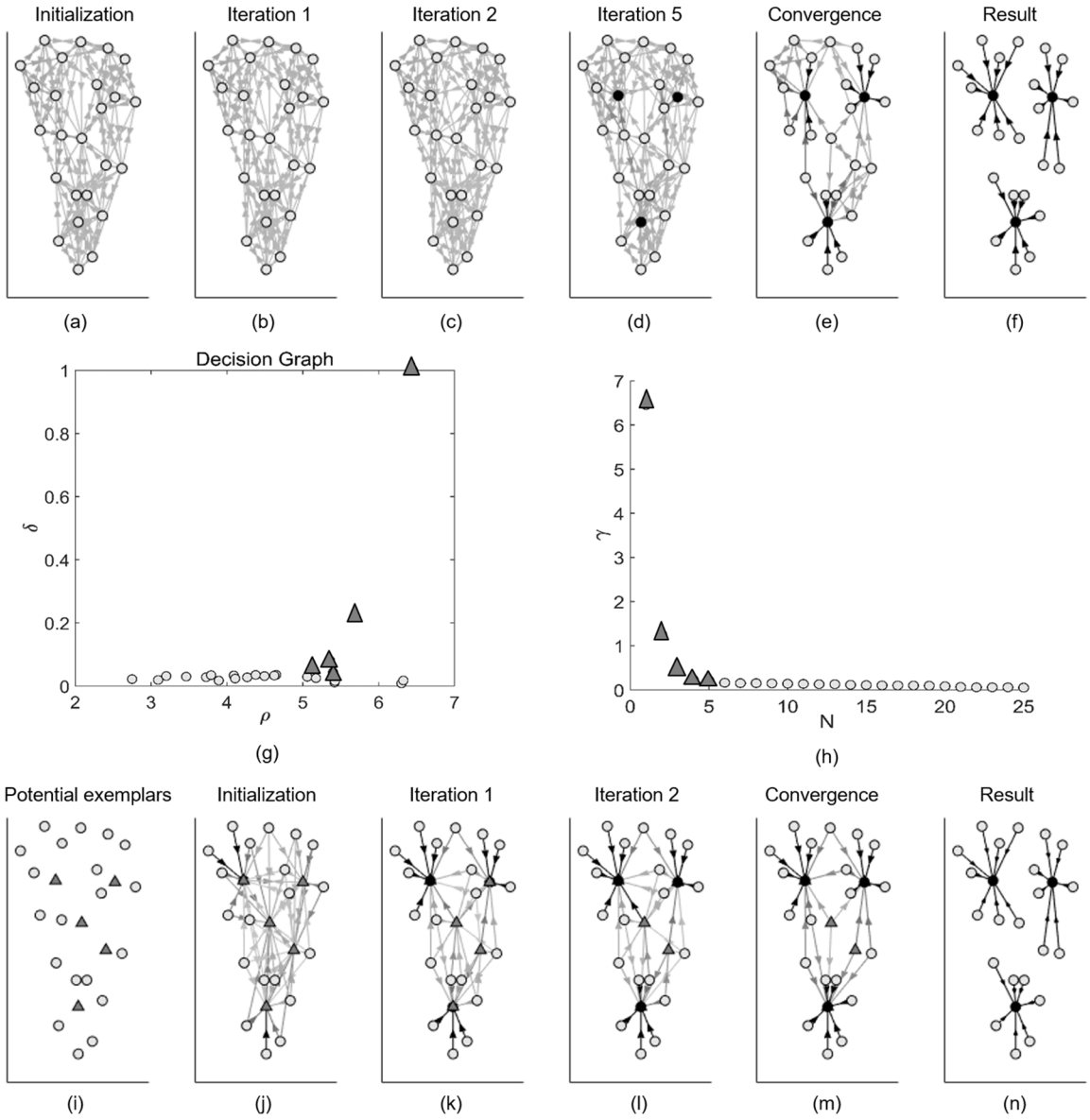

FIGURE 3. A toy example to illustrate the differences between AP and DDAP. (a) ∼ (f) demonstrate how the standard AP clustering works. The darkness of the arrow directed from data point i to data point j corresponds to the strength of the transmitted message that point i belongs to exemplar point j. Exemplars are marked as solid points. Message propagation on the initially completed similarity matrix in (a). Responsibilities and availabilities converge in (e) on the 5th iteration, and the clustering result is shown in (f). (g) ∼ (n) show how the DDAP algorithm works. (g) is the decision graph constructed based on ρ and δ. There are only two prominent points in the upper right corner. (h) shows the value of γ in decreasing order. We choose the top 5 data points as the potential exemplars, which are marked as triangle points in (g)∼(i). Then, messages propagate on the compressed similarity matrix in (j). The process converges in (m) on the second iteration. The final clustering result is shown in (n).

FastAP in [24]. The algorithms proposed in [21]–[23] are not compared in this paper. On the one hand, it has been reported that the clustering performance of these algorithms is inferior to that of the original AP algorithm, although the clustering efficiency has been improved, while the goal of this paper is to propose a clustering algorithm in which its clustering quality is comparable with that of the original AP. On the other hand, the number of clusters is difficult to control in their work, and the number of clusters will greatly affect the evaluation of the clustering performance.

Clustering validation plays an important role in cluster analysis, and numerous measures and methods have been proposed [27], [28]. Clustering quality and running time are the most widely used criteria to evaluate a clustering algorithm. In this paper, we use three quality evaluation criteria (sum of similarities, clustering accuracy, normalized mutual information) and an efficiency criterion (CPU time).

The goal of exemplar-based clustering is to find an exemplar set in which the sum of similarities between each data point and its exemplar is maximized. Therefore, the sum

Algorithm 1 DDAP

Input: the similarity matrix S of N data points, the number of potential exemplars P;

Output: the clustering result $\{e(i)\}$, where $e(i)$ denotes the exemplar of data point i;

Steps:

1. For each data point i, calculate ρ_i and δ_i according to (4) and (5), then calculate γ_i according to (6);
2. Sort N data points by $\{\gamma_i\}$ in descending order, select the first P data points as the potential exemplar set PE;
3. Construct compressed similarity matrix S^*;
4. Implement message propagation on S^* according to (7) and (8);
5. Repeat Step 4 until convergence, and output the clustering result $\{e(i)\}$ according to (9);

of similarities (SS) is the most important criterion of the exemplar-based clustering algorithm. It is defined as

$$SS = \sum_{i=1}^{N} s(i, e(i)) \tag{10}$$

where $s(i, e(i))$ represents the similarity between data point i and its exemplar e(i), and N is the number of data points. A larger SS indicates a better clustering performance.

Normalized mutual information (NMI) evaluates the effectiveness of clustering algorithms by computing the mutual information between real cluster labels and the clustering results. In the exemplar-based clustering, the result label of each data point is equal to its exemplar's label. According to [29]

$$NMI = \frac{I(\boldsymbol{e}, \hat{\boldsymbol{e}})}{\sqrt{H(\boldsymbol{e})H(\hat{\boldsymbol{e}})}} \tag{11}$$

where $I(\boldsymbol{e}, \hat{\boldsymbol{e}})$ represents the mutual information between clustering results $\boldsymbol{e}$ and real cluster labels $\hat{\boldsymbol{e}}$. $H(\cdot)$ represents the information entropy of the variable.

Accuracy is a more intuitive criterion for reflecting the clustering quality, which is defined as

$$ACC = \frac{\sum_{i=1}^{N} \phi(e\,(i)\,, \hat{e}(i))}{N} \tag{12}$$

where e(i) represents the clustering result label of data point i, which is equal to its exemplar's label, $\hat{e}(i)$ is the real cluster label of data point i. $\phi(i, j) = 1$ if $i = j$, and $\phi(i, j) = 0$ otherwise.

In summary, four criteria are used to evaluate the performance of the proposed algorithm. SS is a similarity-based measure, NMI and ACC (accuracy) evaluate the consistency of clustering results with real category labels, and computational time reflects the efficiency of the algorithms.

A. RESULTS ON SYNTHETIC DATASETS

The synthetic data contain uniformly distributed datasets and Gaussian distributed datasets. The first experiment is performed on datasets randomly generated according to a 2D uniform distribution. The number of data points in these datasets is set from 200 to 2,000, and the number of clusters varies from 2 to 10. The cutoff distance in DDC is set to 0.15. The compression rate (the ratio of the number of potential exemplars to the total number of data points) is set to 0.1.

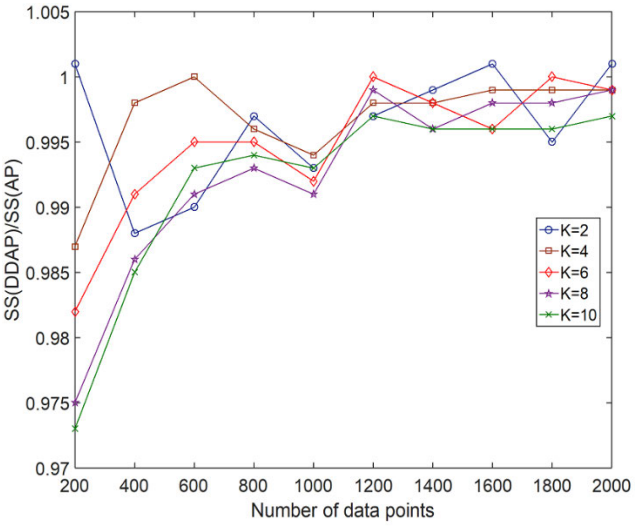

FIGURE 4. Comparison of DDAP and AP by SS on uniformly distributed datasets.

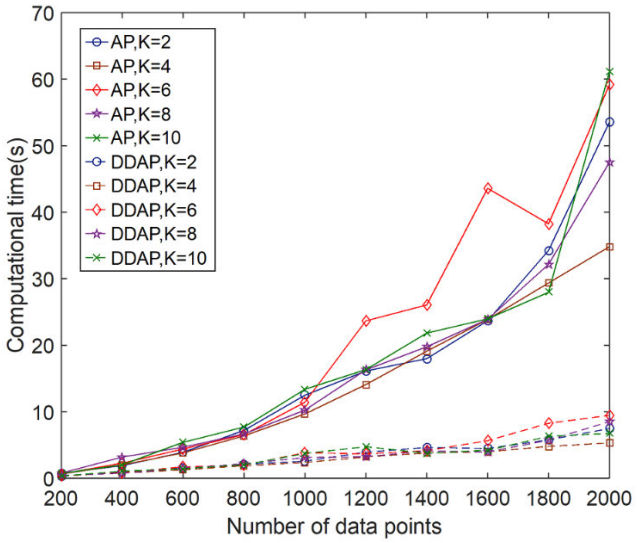

FIGURE 5. Comparison of DDAP and AP by computational time on uniformly distributed datasets.

The complete experimental results are shown in Table 1. The low time cost of DDC tells us that the computational time of the two indicators in DDC is very short. In particular, Fig.4 compares DDAP and the standard AP by SS, and it can be concluded that AP achieves better clustering performance in most cases (the ratio is lower than 1). DDAP can achieve comparable clustering performance with AP (most ratios are between 0.99 and 1), and the clustering performance of DDAP and AP becomes close with the increase of data points. Fig.5 shows the computational time of DDAP and AP. The

TABLE 1. Computational experiment on uniformly distributed datasets.

Datasets	Methods	K=2		K=4		K=6		K=8		K=10	
		SS	Time(s)	SS	Time(s)	SS	Time(s)	SS	Time(s)	SS	Time(s)
N=200	AP	155.73	0.746	171.76	0.667	177.21	0.663	181.09	0.787	183.56	0.723
	DDAP	155.92	0.310	169.56	0.371	173.94	0.366	176.63	0.424	178.64	0.350
	DDC	147.25	0.006	151.51	0.007	152.40	0.003	159.83	0.005	161.78	0.004
N=400	AP	312.84	1.918	341.35	2.016	352.39	2.280	360.73	3.199	365.66	2.037
	DDAP	309.16	1.010	340.66	0.804	349.39	0.860	355.60	0.776	360.08	1.072
	DDC	284.94	0.015	292.96	0.013	296.13	0.011	307.09	0.011	309.25	0.012
N=600	AP	469.59	3.910	515.76	3.827	530.95	4.407	541.18	4.676	547.43	5.421
	DDAP	464.82	1.515	515.57	1.276	528.09	1.729	536.48	1.494	543.52	1.446
	DDC	437.21	0.026	447.28	0.024	450.15	0.024	457.74	0.025	465.92	0.026
N=800	AP	626.60	7.209	690.04	6.380	707.14	6.573	720.59	6.676	729.48	7.704
	DDAP	625.02	2.107	687.15	1.894	703.92	1.972	715.73	2.187	724.93	2.034
	DDC	579.41	0.055	599.47	0.040	612.44	0.046	615.55	0.042	620.59	0.047
N=1000	AP	784.77	12.494	863.50	9.679	885.67	11.441	900.75	10.260	911.65	13.386
	DDAP	779.16	2.580	858.67	2.384	878.71	3.848	892.47	3.124	905.09	3.733
	DDC	725.27	0.067	736.02	0.079	745.06	0.063	72.62	0.077	776.64	0.075
N=1200	AP	940.57	16.135	1034.59	14.091	1062.27	23.672	1083.30	16.291	1096.05	16.345
	DDAP	938.06	3.928	1032.62	3.216	1062.11	3.673	1081.73	3.243	1092.51	4.757
	DDC	870.09	0.104	882.29	0.120	888.17	0.096	906.24	0.116	926.65	0.115
N=1400	AP	1103.07	17.987	1206.82	19.110	1237.98	26.052	1263.13	19.775	1277.72	21.827
	DDAP	1102.16	4.643	1204.12	3.819	1235.64	4.153	1258.37	4.138	1272.95	3.773
	DDC	1028.31	0.116	1040.60	0.147	1051.42	0.151	1070.07	0.154	1080.84	0.190
N=1600	AP	1249.17	23.684	1374.76	23.873	1413.41	43.586	1442.05	23.901	1459.14	23.973
	DDAP	1249.84	4.488	1372.77	3.986	1407.42	5.662	1439.79	3.954	1452.84	4.222
	DDC	1165.07	0.190	1191.76	0.191	1202.72	0.213	1229.39	0.193	1233.35	0.189
N=1800	AP	1423.01	34.245	1554.21	29.363	1590.28	38.254	1622.23	32.166	1644.69	27.955
	DDAP	1415.43	5.784	1552.94	4.806	1589.61	8.273	1618.99	5.736	1637.78	6.308
	DDC	1314.99	0.214	1324.39	0.235	1333.56	0.214	1345.51	0.228	1348.36	0.219
N=2000	AP	1563.49	53.537	1722.23	34.819	1769.04	59.149	1803.74	47.485	1825.51	61.082
	DDAP	1564.89	7.483	1719.80	5.326	1766.94	9.465	1801.18	8.527	1820.25	6.761
	DDC	1458.50	0.280	1494.72	0.332	1529.30	0.277	1542.88	0.279	1543.31	0.267

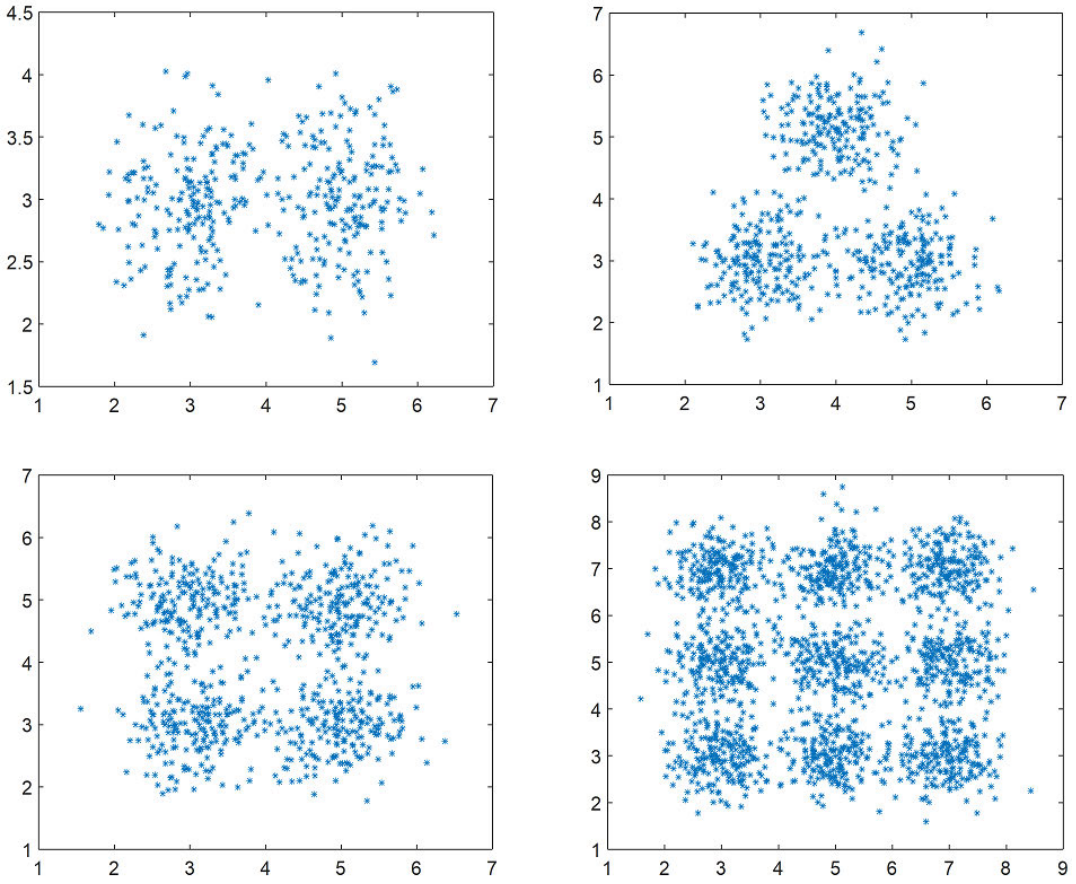

FIGURE 6. Four Gaussian distributed datasets.

time cost of AP is quadric to the number of data points, while the time cost of DDAP is linear to the number of data points. The experimental result is consistent with the theoretical analysis of the time complexity of AP and DDAP.

The second computational experiment is performed on Gaussian distributed datasets. The algorithms are tested by labeled data sets. We can not only compare AP and DDAP by SS and Time, but also compare them with NMI and ACC. Fig.6 shows the four synthetic datasets. The variance of each Gaussian distribution is 0.2. The experimental results are presented in Table 2. DDAP achieves comparable clustering performance with AP while the computational efficiency has been improved observably.

TABLE 2. Computational experiments on Gaussian distributed datasets.

Datasets	Methods	SS	NMI	ACC	Time(s)
N=400	AP	340.93	0.888	0.985	1.704
	DDAP	340.93	0.888	0.985	0.674
	DDC	340.78	0.888	0.985	0.010
N=600	AP	537.93	0.953	0.990	3.471
	DDAP	537.88	0.953	0.990	1.119
	DDC	537.55	0.950	0.990	0.024
N=800	AP	717.04	0.910	0.976	5.591
	DDAP	716.60	0.906	0.975	1.560
	DDC	715.90	0.902	0.974	0.046
N=1800	AP	1681.02	0.926	0.968	29.586
	DDAP	1680.61	0.925	0.968	4.735
	DDC	1679.45	0.922	0.965	0.228

B. RESULTS ON UCI DATASETS

Nine real-world datasets in the UCI machine learning repository are used to evaluate the clustering algorithms, while user knowledge modeling is abbreviated UKM, WDBC is Wisconsin diagnostic breast cancer, MF is multiple features and EGSSD is electrical grid stability simulated data. The class label of each object in the nine datasets is known, which means that the clustering algorithms can be compared with four criteria in this experiment. We compare the proposed DDAP with k-centers, DDC, the original AP algorithm and FastAP in [24]. The k-NN sparsification rate in FastAP is 0.5. The compression rate of the first eight datasets is 0.1, and the last one is 0.01.

Table 3 presents a brief description of the nine datasets and their experimental results. It can be observed that *SS* achieved by k-centers, AP, FastAP and DDAP is comparable while DDC is slightly smaller. In terms of *NMI* and *ACC*, the five methods are comparable in most cases. DDAP obtains the highest *NMI* on UKM, WDBC, Yeast and EGSSD. From the perspective of accuracy, DDAP performs the same as AP on three datasets, and performs best on the other six datasets.

TABLE 3. A brief description of the nine UCI datasets and their experimental results.

Datasets	N	Attributes	Classes	Methods	SS	NMI	ACC	Time(s)
Iris	150	4	3	K-centers	131.1	0.7264	0.8411	0.0002
				DDC	129.0	0.7744	**0.9067**	0.0020
				AP	**132.0**	**0.7777**	0.9000	0.3698
				FastAP	131.4	0.7661	0.9000	0.2681
				DDAP	131.6	0.7578	0.9000	0.1812
Car	260	6	4	K-centers	183.3	0.5218	0.6433	0.0004
				DDC	177.3	0.5398	0.5346	0.0044
				AP	**187.1**	**0.7350**	0.7885	0.8675
				FastAP	182.5	0.6177	0.7154	0.5991
				DDAP	186.0	0.7218	**0.8038**	0.4217
UKM	403	5	4	K-centers	307.7	0.2225	0.5055	0.0006
				DDC	291.3	0.2156	0.5434	0.0106
				AP	**310.6**	0.1073	0.4342	1.9713
				FastAP	309.9	0.1743	0.4640	0.8343
				DDAP	309.0	**0.4537**	**0.6774**	0.8567
WDBC	569	30	2	K-centers	472.4	0.6163	0.9248	0.0028
				DDC	448.1	0.0299	0.6274	0.0198
				AP	**472.6**	0.6202	0.9227	2.8155
				FastAP	465.6	0.4989	0.8910	1.8699
				DDAP	468.6	**0.6827**	**0.9455**	0.9898
DrivFace	606	6400	3	K-centers	395.7	0.0449	**0.9012**	0.0024
				DDC	394.6	**0.0466**	0.9010	0.0229
				AP	407.2	0.0253	0.9010	4.8332
				FastAP	402.0	0.0341	0.9010	1.3271
				DDAP	**407.3**	0.0248	0.9010	1.1238
Yeast	652	6	4	K-centers	516.4	0.2478	0.5341	0.0036
				DDC	499.3	0.1777	0.4540	0.0262
				AP	**522.4**	0.2781	0.5506	5.6949
				FastAP	521.5	0.2686	0.5383	1.3736
				DDAP	518.5	**0.2858**	**0.5537**	2.0575
Semeion	1593	256	10	K-centers	606.9544	0.3756	0.4808	0.0053
				DDC	526.7890	**0.4682**	0.4306	0.1671
				AP	628.2391	0.3821	0.4997	22.2098
				FastAP	**629.0412**	0.3823	0.5204	7.2860
				DDAP	628.2560	0.4247	**0.5461**	4.7633
MF	2000	649	10	K-centers	1099.6	0.6354	0.6806	0.0115
				DDC	948.0	0.7170	0.6340	0.2638
				AP	1127.8	0.6825	0.7575	37.1058
				FastAP	1121.1	**0.7489**	0.8360	11.3633
				DDAP	**1129.4**	0.7403	**0.8390**	5.4949
EGSSD	10000	14	2	K-centers	6266.7	0.0529	0.6585	0.5655
				DDC	6059.2	0.0068	0.6380	7.9533
				AP	**6303.4**	**0.1329**	**0.7120**	3499.0
				FastAP	6202.2	0.1326	**0.7120**	116.77
				DDAP	**6303.4**	**0.1329**	**0.7120**	41.347

TABLE 4. Computational experiments on UCI datasets. Comparison of FastAP and DDAP in terms of computational time.

Datasets	Methods	T-PE	T-KNN	T-CAP	Time(s)
Iris	FastAP	0.0270	0.0370	0.2041	0.2681
	DDAP	0.0020	—	0.1792	0.1812
Car	FastAP	0.0307	0.0516	0.5168	0.5991
	DDAP	0.0044	—	0.4173	0.4217
UKM	FastAP	0.0471	0.0802	0.7070	0.8343
	DDAP	0.0106	—	0.8461	0.8567
WDBC	FastAP	0.1481	0.1149	1.6069	1.8699
	DDAP	0.0150	—	0.9748	0.9898
DrivFace	FastAP	0.0620	0.1254	1.1397	1.3271
	DDAP	0.0150	—	1.1088	1.1238
Yeast	FastAP	0.0820	0.1619	1.1297	1.3736
	DDAP	0.0160	—	2.0415	2.0575
Semeion	FastAP	0.4352	3.2021	3.6487	7.2860
	DDAP	0.1671	—	4.5962	4.7633
MF	FastAP	0.6927	4.8422	5.8284	11.3633
	DDAP	0.2638	—	5.2311	5.4949
EGSSD	FastAP	25.2838	66.0300	25.4546	116.77
	DDAP	7.9500	—	33.3969	41.347

Regarding the computational efficiency, DDAP speeds up 2–8 times than AP. The larger the dataset, the more significantly DDAP speeds up. In particular, the dataset EGSSD contains 10,000 nodes, and the compression rate is 0.01, that is, 100 potential exemplars are selected for message exchanges. The clustering results of DDAP and AP are the same, but the DDAP computational time is less than one minute while the AP computational time is nearly one hour.

Table 4 presents the specific computational time of DDAP and FastAP, where T-PE represents the computational time of selecting potential exemplars, T-KNN represents the time of sparsing the similarity matrix by k-NN, T-CAP represents the time of the AP clustering based on the compressed similarity matrix, and Time is the overall time. When the dataset is small, the overall times of DDAP and FastAP are comparable, when the dataset becomes larger, DDAP is significantly faster than FastAP. The main reason is that the method of selecting potential exemplars in FastAP is more time-consuming than the method in DDAP, and the process of sparsing the similarity matrix by k-NN also increases the computational time of FastAP. Fig.7 shows the specific computational time of DDAP and FastAP on UCI datasets. It demonstrates that the time cost of selecting potential exemplars can be ignored compared with the time cost of message propagation.

All the experiments demonstrate that DDAP can achieve comparable clustering performance with the standard AP with a significantly improved computational efficiency.

C. RESULTS ON THE EMRS DATASET

In the past decade, EMRs data mining has advanced rapidly [30]. The availability of massive EMRs has enabled a new paradigm for optimizing healthcare practices [31]. The EMRs

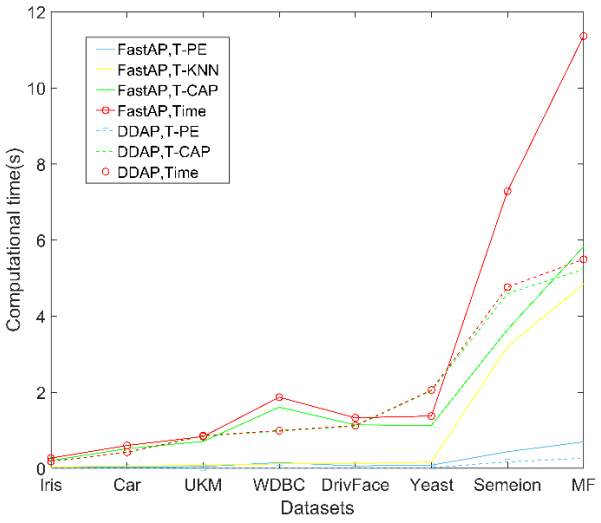

FIGURE 7. Comparison of DDAP and FastAP in terms of computational time on UCI datasets.

dataset used in this paper was collected from the information systems of 9 hospitals. All of the EMRs are about patients who have used Xiyanping injection, which is a traditional Chinese medicine injection with significant anti-inflammatory and antipyretic effects and favorable safety. This EMRs dataset was collected to evaluate the usage of Xiyanping injection in actual clinical applications. For example, safe and effective dosage, therapeutic effects for different diseases, safe and effective combinations with other medicines, and some other research. In this paper, we focus on discovering typical combinations of Xiyanping injection and other medicines in patients with respiratory disease, which is one of the most common diseases. In this dataset, respiratory disease mainly includes three types: lung infections, bronchitis, and upper respiratory tract infections.

For a patient, a doctor's order consists of medicine name, dosage, delivery route, frequency, starting time, and ending time. After preprocessing the raw data, including removing some incorrect records and unifying the medicine names, we obtained 800 respiratory disease patients with 10,400 doctor's orders, nearly 13 orders for each patient. The number of all medicines used was 421.

Fig.8 is the medicine combination network we constructed, where each node represents a medicine, the edge between two nodes represents the combined use of two medicines by a patient, and the weight of the edge represents the number of times the combination is used. The node degree in this network is consistent with the power-law distribution. We remove edges with low weights and obtain the core of the network shown in Fig.9 Xiyanping injection, 5% dextrose injection and 0.9% sodium chloride injection are at the center of the subnetwork. This subnetwork is a very densely connected group and it is difficult to find high-quality typical combinations of Xiyanping injection and other medicines by a community partitioning algorithm.

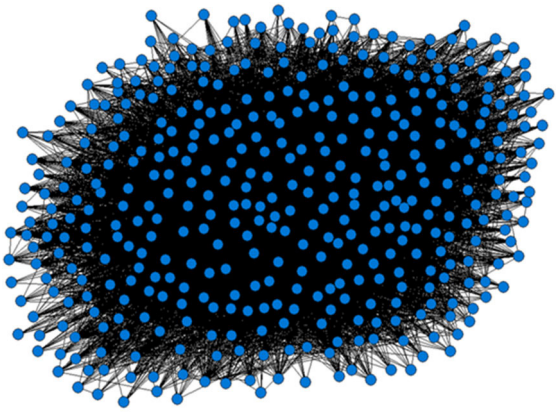

FIGURE 8. The medicine combination network.

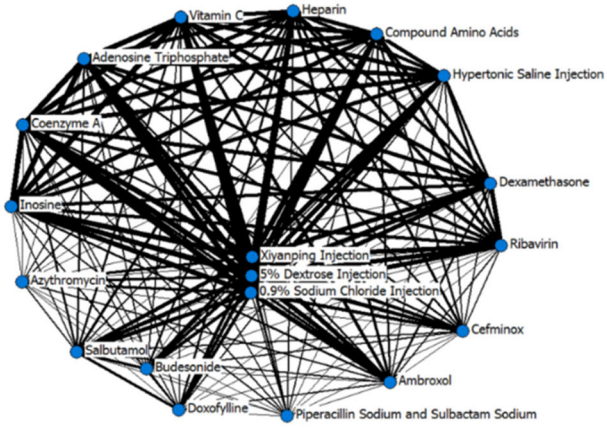

FIGURE 9. The core of the medicine combination network.

We perform a cluster analysis of treatments to discover the typical combined use of Xiyanping injection and other medicines in actual clinical practice. The experiment consists of three steps: 1) computing similarities between treatments, 2) clustering treatments, and 3) extracting typical medicine combinations from treatment clusters.

For the sake of simplicity, the treatment of the patient i is defined as

$$T_i = \{M_{i1}, \ldots, M_{ik}, \ldots, M_{is}\} \tag{13}$$

where M_{ik} represents the kth medicine used by patient i, and patient i used a total of s medicines. The similarity between two treatments is measured by the Jaccard coefficient

$$\mathrm{s}\left(T_i, T_j\right) = \frac{\left|T_i \cap T_j\right|}{\left|T_i \cup T_j\right|} \tag{14}$$

We calculate ρ_i and δ_i for each treatment. Fig.10 is the decision graph based on the two indicators. We can see that there is only one obvious exemplar at the top right corner. Fig.11 is the value of $\gamma_i = \rho_i\delta_i$ in decreasing order. It is difficult to determine the number of exemplars and the decision is subjective.

We implement our method with a compression rate of 0.1, which means that the number of potential exemplars $P = 80$.

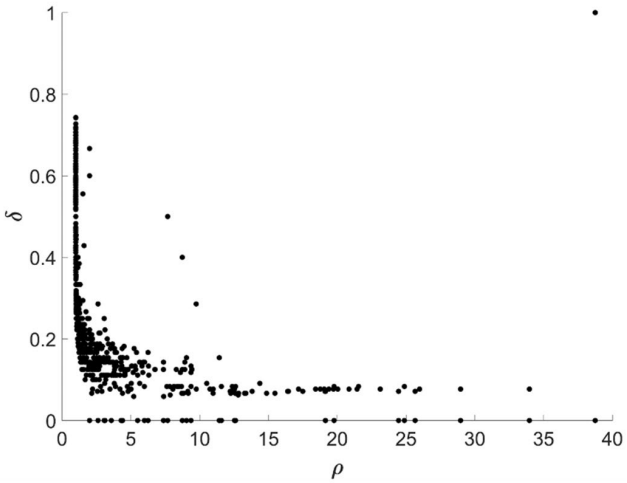

FIGURE 10. The decision graph of the EMRs dataset.

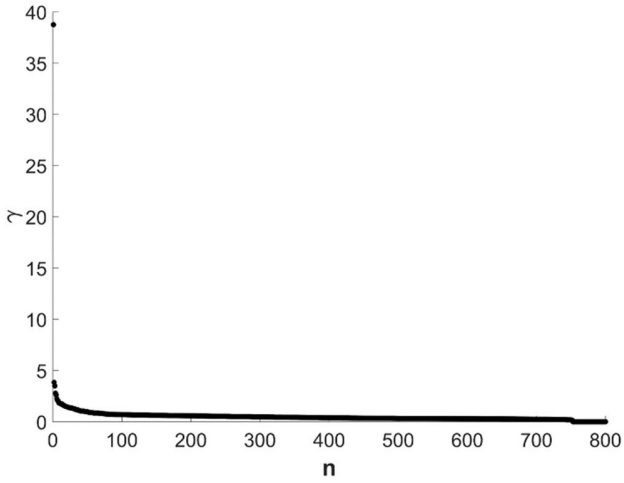

FIGURE 11. The value of $\gamma_i = \rho_i\delta_i$ in decreasing order for the EMR dataset.

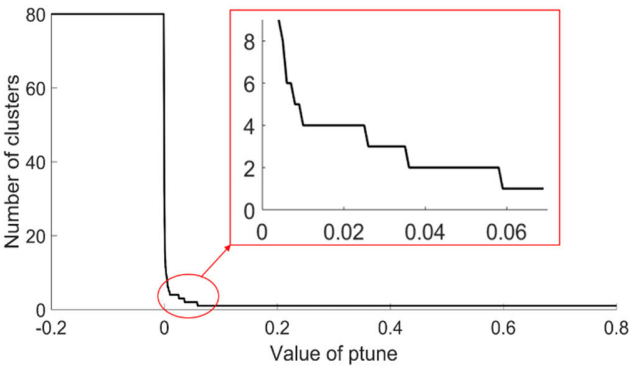

FIGURE 12. The number of clusters changes with the value of *ptune*, which is a variant of the preference p. $p = median\,(S) - ptune * N$, where N is the number of data points, S is the similarity matrix, and *ptune* is more stable than p.

One advantage of AP is that the number of clusters does not need to be specified in advance, and the appropriate number of clusters emerges from the message propagation method and depends on the preference p. Fig.12 shows the effect of

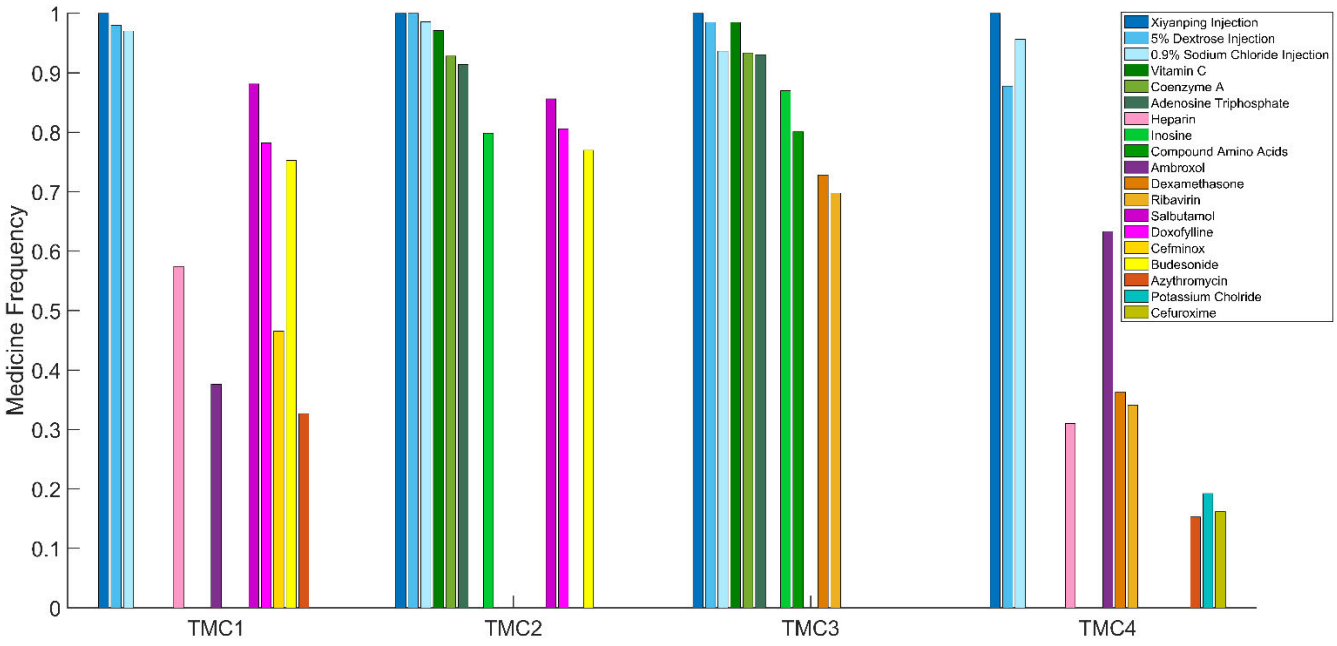

FIGURE 13. Four typical treatment medicine combinations extracted from the EMRs dataset.

TABLE 5. Computational experiments on the EMRs dataset.

Methods	SS	Time(s)
AP	459.6	29.65
DDAP	460.5	4.493
DDC	444.6	0.145

TABLE 6. Clusters of the EMRs dataset by DDAP.

Cluster	Number of Patients	Percent
1	101	12.625
2	139	17.375
3	331	41.375
4	229	28.625

the value of *ptune* on the number of clusters. We can see that the appropriate number of clusters can be 2 or 4. When the number of clusters is 2, one cluster is too small and the other one is too large, and most treatments are clustered in one cluster. To obtain a meaningful result, we choose the number of clusters to be 4. Table 5 shows the experimental results for SS and Time.

We divide the 800 treatments into four clusters. Each cluster corresponds to a typical combination of Xiyanping injection and other medicines. Table 6 shows the size of the four clusters found by DDAP. For each cluster, we choose the top 10 popularly used medicines. A total of 19 medicines are obtained for four clusters. The functions of these medicines are divided into seven categories, as shown in Table 7. Fig.13 illustrates the four extracted typical treatment medicine combinations for respiratory disease.

TABLE 7. The function of the main medicines.

	Function	Medicine
A	Anti-inflammatory, antifebrile antibiotic, antiviral, antibacterial	Xiyanping Injection, Dexamethasone, Ribavirin, Cefminox, Budesonide, Azythromycin, Cefuroxime
B	diluent	5% Dextrose Injection, 0.9% Sodium Chloride Injection
C	participate in human metabolism to provide energy and improve nutrition	Vitamin C, Coenzyme A, Adenosine Triphosphate, Compound Amino Acids, Inosine
D	expectorant	Ambroxol
E	antiasthmatic	Salbutamol, Doxofylline
F	anticoagulant	Heparin
G	regulate the balance of water and electrolytes in the body	Potassium Chloride

Because Xiyanping injection needs to be diluted with 5% dextrose injection or 0.9% sodium chloride injection, these three medicines are the most frequently used in each cluster. In typical medicine combination 1 (TMC1), salbutamol and doxofylline are frequently used, which are antiasthmatic.

Therefore, TMC1 is a combination of Xiyanping injection and antiasthmatic medicines for patients with antiasthmatic needs. In TMC2, in addition to salbutamol and doxofylline, there is a set of medicines affecting human metabolism to provide energy and improve nutrition. This shows that patients in this cluster are weak and need nutritional support. The patient situations in this cluster are complicated. In TMC3, dexamethasone and ribavirin are used frequently. For different types of infections, Xiyanping injection can be combined with other types of antibiotics to reduce the time of fever and shorten the course of treatment. In TMC4, Xiyanping injection is combined with expectorant ambroxol.

Our method can automatically extract typical treatments from a large scale EMR dataset. These four typical treatment medicine combinations are different from each other because of the different usages of the 19 most popular medicines for respiratory disease. By this experiment, we can conclude that our DDAP method can quickly find representative exemplars and obtain excellent clustering results while significantly improving computational efficiency.

V. CONCLUSION

AP and DDC have been widely used in various applications successfully. However, the time complexity of AP is high and the selection of the cluster centers in DDC is relatively subjective. In this paper, we propose a two-stage clustering algorithm DDAP to overcome these shortcomings. We compress the similarity matrix greatly by selecting a small number of potential exemplars based on two quantities, then implement message-passing on the incomplete similarity matrix to obtain the final clustering result. Experimental results on two synthetic datasets, nine publicly available datasets, and a real-world EMRs dataset demonstrate that DDAP can achieve comparable clustering performance with the original AP algorithm, while the computational efficiency improves observably.

The main contribution of the paper is to greatly improve the computing efficiency of the AP algorithm by selecting representative potential exemplars. Since AP is an exemplar-based clustering algorithm, it is only suitable for clustering data with spherical clusters. In practical applications, the data is more complicated. For the future work, we can extend the fast AP algorithm to cluster data with complex shaped clusters.

REFERENCES

[1] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Waltham, MA, USA: Morgan & Claypool, 2011.
[2] A. Nazari, A. Dehghan, S. Nejatian, V. Rezaie, and H. Parvin, "A comprehensive study of clustering ensemble weighting based on cluster quality and diversity," *Pattern Anal. Appl.*, vol. 22, no. 1, pp. 133–145, Feb. 2019.
[3] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.
[4] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015.
[5] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.
[6] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[7] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ, USA: Wiley, 2009.
[8] R. Guan, X. Shi, M. Marchese, C. Yang, and Y. Liang, "Text clustering with seeds affinity propagation," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 4, pp. 627–637, Apr. 2011.
[9] P. A. Karegar, "Wireless fingerprinting indoor positioning using affinity propagation clustering methods," *Wireless Netw.*, vol. 24, no. 8, pp. 2825–2833, Nov. 2018.
[10] W.-F. Guo and S.-W. Zhang, "A general method of community detection by identifying community centers with affinity propagation," *Phys. A, Stat. Mech. Appl.*, vol. 447, pp. 508–519, Apr. 2016.
[11] I. E. Givoni and B. J. Frey, "Semi-supervised affinity propagation with instance-level constraints," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, Clearwater, FL, USA, 2009, pp. 161–168.
[12] X. Zhang, W. Wang, K. Nørvåg, and M. Sebag, "K-AP: Generating specified K clusters by efficient affinity propagation," in *Proc. IEEE Int. Conf. Data Mining*, Sydney, NSW, Australia, Dec. 2010, pp. 1187–1192.
[13] I. E. Givoni, C. Chung, and B. J. Frey, "Hierarchical affinity propagation," in *Proc. 27th Conf. Uncertainty Artif. Intell.*, Arlington, VA, USA: AUAI Press, 2011, pp. 238–246.
[14] L. Sun and C. Guo, "Incremental affinity propagation clustering based on message passing," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 11, pp. 2731–2744, Nov. 2014.
[15] X. Bai, P. Yang, and X. Shi, "An overlapping community detection algorithm based on density peaks," *Neurocomputing*, vol. 226, pp. 7–15, Feb. 2017.
[16] J. Xu, G. Wang, and W. Deng, "DenPEHC: Density peak based efficient hierarchical clustering," *Inf. Sci.*, vol. 373, pp. 200–218, Dec. 2016.
[17] D. Cheng, Q. Zhu, J. Huang, Q. Wu, and Y. Lijun, "Clustering with local density peaks-based minimum spanning tree," *IEEE Trans. Knowl. Data Eng.*, early access, Jul. 29, 2019, doi: 10.1109/TKDE.2019.2930056.
[18] D. Cheng, Q. Zhu, and Q. Wu, "A local cores-based hierarchical clustering algorithm for data sets with complex structures," in *Proc. IEEE 42nd Annu. Comput. Softw. Appl. Conf.*, Jul. 2018, pp. 410–419.
[19] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. I: The *p*-centers," *SIAM J. Appl. Math.*, vol. 37, no. 3, pp. 513–538, Dec. 1979.
[20] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.* Oakland, CA, USA: Univ. California Press, 1967, pp. 281–297.
[21] Y. Fujiwara, G. Irie, and T. Kitahara, "Fast algorithm for affinity propagation," in *Proc. 22nd Int. Conf. Artif. Intell.*, Barcelona, Spain, 2011, pp. 2238–2243.
[22] Y. Jia, J. Wang, C. Zhang, and X. Hua, "Finding image exemplars using fast sparse affinity propagation," in *Proc. 16th ACM Int. Conf. Multimedia*, New York, NY, USA, 2008, pp. 639–642.
[23] J. Xiao, J. Wang, P. Tan, and L. Quan, "Joint affinity propagation for multiple view segmentation," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, 2007, pp. 1–7.
[24] L. Sun, C. Guo, C. Liu, and H. Xiong, "Fast affinity propagation clustering based on incomplete similarity matrix," *Knowl. Inf. Syst.*, vol. 51, no. 3, pp. 941–963, Jun. 2017.
[25] L. Wang, X. Tao, X. Han, J. Han, Y. Liu, and G. Mu, "Semi-supervised affinity propagation based on density peaks," *Teh. Vjesn*, vol. 23, no. 6, pp. 1787–1794, 2016.
[26] L. Wang and Z. Hao, "Gravity theory-based affinity propagation clustering algorithm and its applications," *Tech. Gazette*, vol. 25, no. 4, pp. 1125–1135, Apr. 2018.
[27] D. Cheng, Q. Zhu, J. Huang, Q. Wu, and L. Yang, "A novel cluster validity index based on local cores," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 985–999, Apr. 2019.
[28] S.-O. Abbasi, S. Nejatian, H. Parvin, V. Rezaie, and K. Bagherifard, "Clustering ensemble selection considering quality and diversity," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1311–1340, Aug. 2019.
[29] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Is a correction for chance necessary?" in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, New York, NY, USA, 2009, pp. 1073–1080.
[30] J. Chen, W. Wei, C. Guo, L. Tang, and L. Sun, "Textual analysis and visualization of research trends in data mining for electronic health records," *Health Policy Technol.*, vol. 6, no. 4, pp. 389–400, Dec. 2017.
[31] L. Sun, C. Liu, C. Guo, H. Xiong, and Y. Xie, "Data-driven automatic treatment regimen development and recommendation," in *Proc. 22th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1865–1874.

YANG LI received the B.S. degree in mathematics from the Dalian University of Technology, in 2011, where she is currently pursuing the Ph.D. degree with the Institute of Systems Engineering. Her research interests include complex networks, super networks, cluster analysis, and healthcare data mining.

CHONGHUI GUO received the B.S. degree in mathematics from Liaoning University, in 1995, the M.S. degree in operational research and control theory, and the Ph.D. degree in management science and engineering from the Dalian University of Technology, in 2002. He was a Postdoctoral Research Fellow with the Department of Computer Science, Tsinghua University. He is currently a Professor with the Institute of Systems Engineering, Dalian University of Technology. He has published over 100 peer-reviewed papers in academic journals and conferences, besides five text-books and two monographs. He has been the Principal Investigator of over ten research projects from the Government and the Industry. His research interests include data mining and knowledge discovery.

LEILEI SUN (Member, IEEE) received the B.S. and M.S. degrees from the School of Control Theory and Control Engineering, in 2009 and 2012, respectively, and the Ph.D. degree in management science and engineering from the Dalian University of Technology, in 2017. From September 2015 to August 2016, he was a Visiting Ph.D. Student with Rutgers, The State University of New Jersey. He was a Postdoctoral Research Fellow with the Department of Management Science and Engineering, Tsinghua University. He is currently an Assistant Professor with the School of Computer Science and Engineering, Beihang University. His main research interests include machine learning and data mining, including deep spatio-temporal networks, deep multi-task learning, graph neural networks, and time series data analysis and mining.