IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

SPECIAL SECTION ON DATA-DRIVEN MONITORING, FAULT DIAGNOSIS AND CONTROL OF CYBER-PHYSICAL SYSTEMS

Received January 6, 2018, accepted February 16, 2018, date of publication February 22, 2018, date of current version March 13, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2808485

Enhanced Dominant Sets Clustering by Cluster Expansion

JIAN HOU, (Member, IEEE), AND AIHUA ZHANG
[1]College of Engineering, Institute of Automation, Bohai University, Jinzhou 121013, China

Corresponding author: Jian Hou (dr.houjian@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61473045, and in part by the Natural Science Foundation of Liaoning Province under Grant 20170540013.

ABSTRACT While a vast amount of clustering algorithms of different types are available in the literature, the majority of existing algorithms depend on carefully tuned parameters to obtain satisfactory results. In this paper, we reduce the dependence on parameters on the basis of the dominant sets algorithm. The dominant sets algorithm is a parameter-independent clustering approach, which uses the pairwise data similarity matrix as input. If the data for clustering are in the form of feature vectors, it is necessary to measure the data similarity and build the similarity matrix. With the commonly used Gaussian kernel, the involved parameter is found to exert a significant influence on the clustering results. We study in depth why and how the dominant sets clustering results are influenced by the parameter and attribute the influence to the dominant set definition, which imposes a somewhat too strict constraint on internal similarity. A two-step clustering algorithm is then proposed to solve this problem. First, we transform the similarity matrix by histogram equalization before clustering, and this is shown to eliminate the influence of similarity parameter effectively. In the second step, we expand the clusters to maximize the ratio of internal similarity with respect to external similarity. Our algorithm is designed to achieve the balance between high internal similarity and low external similarity, thereby relieving the dependence on the similarity parameter. In experiments on ten publicly available data sets, our algorithm is shown to perform well in comparison with several other algorithms which benefit from carefully tuned parameters.

INDEX TERMS Clustering, fault diagnosis, pattern classification, dominant set.

I. INTRODUCTION

Data clustering is an important machine learning technique and has received extensive attention for decades and numerous clustering algorithms have been proposed [1]–[3]. In centroid-based clustering algorithms, k-means and its variants are commonly used due to their simpleness and effectiveness. Density-based algorithms, e.g., DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [4] and OPTICS (Ordering Points To Identify the Clustering Structure) [5], detect clusters based on the density difference across cluster borders, and can usually be used to detect non-spherical clusters. As a prominent distribution-based algorithm, EM (Expectation Maximization) is based on the assumption that the data distribution can be approximated by the mix of a set of predefined distribution models. Popular clustering algorithms also include mean-shift and NCuts (Normalize Cuts) [6], which is a typical example of spectral clustering [7]. In recent developments, AP (Affinity Propagation) [8] is proposed to determine the centers and members of clusters as the outcome of passing among data the affinity messages. Rodriguez and Laio [9] proposed to select cluster centers by making use of the local density of data and the distance to the nearest neighbors with larger density. On condition that the cluster centers are identified correctly, this algorithm generates excellent clustering results on some datasets. Data clustering has been shown to be quite useful and has potential to be applied in various fields, including pattern recognition, data mining, image analysis and fault diagnosis, etc [10]–[12].

While a large amount of clustering approaches have been proposed from different perspectives, and some of them have achieved impressive success in real applications, we still need to deal with some issues in order to apply these algorithms to practical clustering tasks. Firstly, the majority of clustering algorithms require one or more user-specified parameters as the input, and their results rely heavily on the input parameters. One of the most commonly required parameters is the number of clusters, which has a direct influence on the clustering results of k-means-like algorithms, NCuts and the general spectral clustering algorithms. Some algorithms are able to determine the number of clusters automatically, but their results depend on other parameters. For example, the parameter input of DBSCAN includes *Eps* denoting a neighborhood radius and *MinPts* denoting the minimum cluster size, and AP requires as input the preference values of all the data for clustering. The density peak based algorithm in [9] is found to be sensitive to the cutoff distance, and the cluster centers may need to be selected manually. Secondly, many algorithms, e.g., k-means-like algorithms, can only generate clusters of spherical shapes. This means that these algorithms are not able to generate satisfactory results in many cases, even if they are assigned carefully tuned parameters. In addition, outlier detection [4] and overlapping clustering [13] may be quite useful in some cases, although they are out of the ability of many existing algorithms.

While some solutions have been found for each of the aforementioned problems, few of existing algorithms have shown the potential to solve all these problems. In this aspect, the dominant sets (DSets) algorithm [14] seems to provide a promising approach. The DSets algorithm defines a dominant set as a subset of data with high internal similarity and low external similarity, enabling a dominant set to be regarded as a cluster. Given the pairwise data similarity matrix as input, the DSets algorithm extracts the clusters one by one and the number of clusters is obtained in the clustering process automatically. Since the data in a dominant set are highly similar to each other, the outliers are left unclustered. Furthermore, the dominant set concept can be applied in overlapping clustering by casting the problem in a game-theoretic framework [15], [16]. Finally, in DSets clustering the ordering of clusters reflects the density difference among these clusters, and each data in a cluster is assigned a weight representing the relationship with other data. These properties may also be quite useful and are not shared by any other existing algorithms, to the best of our knowledge. Based on these nice properties, the DSets algorithm has been successfully applied to various tasks [17]–[20]. Some closely related works also include [21]–[25].

The DSets algorithm itself requires the pairwise data similarity matrix as the single input and no parameters are involved. However, in many cases the data for clustering are in the form of feature vectors, and it is necessary to measure the data similarity and build the pairwise similarity matrix. Although it is possible to use non-parametric similarity measures, e.g., cosine similarity, the experimental study in [26] shows that this is not a good option. With the Gaussian kernel $s(x, y) = exp(-d(x, y)/\sigma)$, the similarity parameter σ is introduced. Given a dataset, σ impacts on the similarity matrix, and then influences the DSets clustering results. We investigate how σ influences the clustering results and attribute the influence to the dominant set definition, which requires each pair of data in a cluster are similar to each other. We then propose to use a two-step algorithm to solve the problem. The similarity matrix is transformed by histogram equalization [27] before clustering. This transformation is shown to eliminate the dependence on σ's effectively and generate small clusters. Then in the second step we expand the small clusters to improve the clustering results [28]. The effectiveness of our algorithm is validated in extensive experiments on several datasets. Since the parameter σ is introduced in applying the DSets algorithm to clustering data in vector form, in this paper our work is limited to the special case that data are represented as vectors and the pairwise similarity is measured by $s(x, y) = exp(-d(x, y)/\sigma)$. In other words, the parameter dependence problem is with this special case, but not the DSets algorithm itself. For each of expression, we use *target case* to denote the above-mentioned special case in this paper. This paper has the following contributions. First, we make an in-depth study on why and how the parameter σ impacts on the DSets clustering results in the target case, based on which we explain the experimental results in details. Second, we show in theory that the influence of σ can be eliminated completely by histogram equalization of similarity matrices, and discuss practical issues in implementation. Third, we use both Normalized Mutual Information (NMI) and Rand index to evaluate the clustering results and make the obtained conclusions more convincing.

We organize the rest of this paper as follows. The brief introduction of the DSets algorithm and its problems are presented in Section II. Then in Section III we show how these problems can be solved and provide the details of our algorithm. We use extensive experiments to validate the major steps and the whole procedures of our algorithm in Section IV and discuss some related issues in Section V. Finally, the concluding remarks are given Section VI.

II. DOMINANT SETS CLUSTERING AND PROBLEMS

In this section we firstly introduce the definition of dominant set and the DSets algorithm. Then we discuss the problem of the DSets algorithm and analyze the reason.

A. DOMINANT SET

Dominant set is defined as a graph-based cluster concept and the n data for clustering are represented with a graph. As customary, we use an edge-weighted graph $G = (V, E, w)$ to represent the pairwise relationship among the data. Here V denotes the set of data to be clustered, E conveys the edge relationship among the data, and w represents the edge weight. With the pairwise similarity matrix $A = (a_{ij})$, we have $w_{ij} = a_{ij}$ if $(i, j) \in E$ and $w_{i,j} = 0$ otherwise. Since we use the graph in clustering and one data should not be similar to itself, the graph G has no self-loops and $a_{ii} = 0$ for $i = 1, \cdots, n$.

The basic requirement of a cluster is the high internal similarity and low external similarity. In [14] a weight criterion $w_S(i)$ is defined to differentiate between the *high* and *low* similarities. Specifically, a positive $w_S(i)$ means that i has high similarity with the data in a subset $S \subseteq V$ and a negative $w_S(i)$ indicates the reverse. Based on this criterion, it is natural to obtain a dominant set S by including all the data i with positive $w_S(i)$ and excluding all those with negative weights. The dominant set obtained this way satisfies the condition of high internal similarity and low external similarity, and therefore is qualified to be regarded as a cluster. We briefly introduce the definition of dominant set in the following, and the details can be found in [14].

The criterion $w_S(i)$ is defined as

$$w_S(i) = \begin{cases} 1, & \text{if} \quad |S| = 1, \\ \sum_{j \in S \setminus \{i\}} \phi_{S \setminus \{i\}}(j, i) w_{S \setminus \{i\}}(j), & \text{otherwise.} \end{cases} \tag{1}$$

where

$$\phi_S(i, j) = a_{ij} - aw_S(i). \tag{2}$$

and

$$aw_S(i) = \frac{1}{|S|} \sum_{k \in S} a_{ik}. \tag{3}$$

with $i \in S, j \notin S$ and $S \subseteq V$.

We now present the formal definition of dominant set as follows. With $W(S) = \sum_{i \in S} w_S(i)$, we call the subset S such that $W(T) > 0$ for all non-empty $T \subseteq S$ as a dominant set if

1) $w_S(i) > 0$, for all $i \in S$.
2) $w_{S \bigcup \{i\}}(i) < 0$, for all $i \notin S$.

It is shown in [14] that a dominant set can be extracted with the replicator dynamics developed in evolutionary game theory. Specifically, we use a vector $x \in R^n$ to denote the weights of all the n data, where the data with positive weights belong to a dominant set. The weight vector can be obtained with the replicator dynamics as

$$x_k^{(t+1)} = x_k^{(t)} \frac{(Ax^{(t)})_k}{x^{(t)T} Ax^{(t)}} \tag{4}$$

where $k = 1, \ldots, n$. In this paper, however, we use the more efficient infection and immunization dynamics proposed in [29].

By treating a dominant set as a cluster, the DSets algorithm generates clusters sequentially. Specifically, we obtain a cluster, and continue to extract the next one in the remaining unclustered data. In this way we accomplish the clustering process and the number of clusters is determined automatically.

B. PROBLEMS

The DSets algorithm requires only the pairwise data similarity matrix as input. However, in applying the DSets algorithm to the target case where data are in the form of feature vectors, the similarity parameter σ is introduced. Given the data for clustering, the variance of σ result in the change of similarity matrices, which is then found to yield different clustering results. For illustration, we apply the DSets algorithm to ten datasets, including Aggregation [30], Pathbased [31], D31 [32], R15 [32], Flame [33], Jain [34] and four UCI datasets Wine, Iris, Glass and Yeast. The properties of these datasets are shown in Table 1. We evaluate the clustering results with NMI (Normalized Mutual Information) and Rand index. Both criterions compare the clustering results with the ground truth and use high scores to denote accurate clustering results. The DSets clustering results on the ten datasets with different σ's are reported in Figure 1. Note that for ease of expression, in Figure 1 the horizontal axes show only the coefficients of σ's, and the real values of σ's are the products of $\overline{d}$ and the horizontal axes.

TABLE 1. The properties of the datasets used in experiments.

	# of points	Data dimension	# of clusters
Aggregation	788	2	7
Pathbased	300	2	3
D31	3100	2	31
R15	600	2	15
Jain	373	2	2
Flame	240	2	2
Wine	178	13	3
Iris	150	4	3
Glass	214	9	7
Yeast	1484	8	10

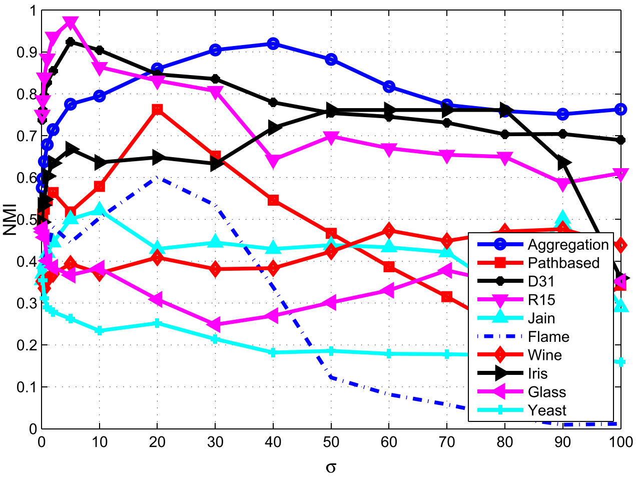


FIGURE 1. Influence of σ on the DSets clustering results.

In Figure 1 we observe that for all the datasets, σ has a significant influence on the clustering results. This implies that a careful parameter tuning process is necessary for satisfactory clustering results. However, Figure 1 shows that the best-performing σ's vary widely with different datasets, and there doesn't exist a fixed σ which is appropriate for different datasets.

C. THE REASON

In order to relieve the parameter dependence problem shown in the above subsection, we firstly study why σ influences the clustering results of the DSets algorithm. Since in the DSets algorithm the dominant sets are treated as clusters, we start our investigation from the dominant set definition.

In dominant set definition each data i in a dominant set S has a positive $w_S(i)$. Since $w_S(i)$ is defined in Eq. (1) in a recursive form, its meaning is not straightforward.

However, noticing that one major component in Eq. (1) is $\phi_{S\setminus\{i\}}(j,i)$, the value of $w_S(i)$ can be regarded as a weighted sum of $\phi_{S\setminus\{i\}}(j,i)$, where $j \in S \setminus \{i\}$. By ignoring the weight items $w_{S\setminus\{i\}}(j)$ in Eq. (1), we can approximate Eq. (1) to be

$$w'_S(i) = \begin{cases} 1, & \text{if} \quad |S| = 1, \\ \sum_{j\in S\setminus\{i\}} \phi_{S\setminus\{i\}}(j,i), & \text{otherwise.} \end{cases} \tag{5}$$

From Eq. (2) and Eq. (3), and by defining

$$\delta(i,S) = \frac{1}{|S|}\sum_{k\in S} a_{ik}, \tag{6}$$

$$\delta(S) = \frac{1}{|S|(|S|-1)}\sum_{j\in S,k\in S} a_{jk}, \tag{7}$$

we further obtain

$$w'_S(i) = \begin{cases} 1, & \text{if} \quad |S| = 1, \\ \delta(i,S\backslash\{i\}) - \delta(S\backslash\{i\}), & \text{otherwise.} \end{cases} \tag{8}$$

Here we see that $w_S(i)$ provides a measure of the comparison between two similarity values, i.e., the average similarity between i and the data in $S \setminus \{i\}$, and the average overall similarity in $S \setminus \{i\}$. This is consistent with the statement in [14] that $w_S(i)$ *gives us a measure of the overall (relative) similarity between vertex i and the vertices of* $S \setminus \{i\}$ *with respect to the overall similarity among the vertices in* $S \setminus \{i\}$. Therefore a positive $w_S(i)$ means approximately that $\delta(i,S\backslash\{i\})$ is greater than $\delta(S\backslash\{i\})$, which further indicates that i is similar to all the other data in S. Since each data i in a dominant set has a positive $w_S(i)$, we know that in a dominant set each pair of data are similar to each other. In fact, we can also understand this argument based on another statement in [14] that dominant sets are the extension of maximal cliques in unweighted graphs to edge-weighted graphs. Since in a clique each pair of vertices are connected, we know that in a dominant set each pair of data are similar to each other.

Based on the above conclusion, we are ready to explain how σ impacts on the clustering results. It is evident from $s(x,y) = exp(-d(x,y)/\sigma)$ that a large σ results in large similarity values. In this case, it is easy to find large subsets of data with high pairwise similarity and obtain large clusters. On the contrary, a small σ results in small similarity values and then small clusters. The influence of σ on cluster sizes is illustrated in Figure 2. Evidently both too small and too large clusters degrade the clustering quality, as illustrated in Figure 1.

III. OUR ALGORITHM

In the target case the DSets clustering results are found to be influenced by the parameter σ, and it is difficult to find out an appropriate σ applicable to different datasets. In addition, on some datasets even the best-performing σ's perform unsatisfactorily. This means that it may not be a good option to attempt to find out the appropriate σ. Therefore we resort to a different solution. We firstly use histogram equalization transformation of similarity matrices to remove the influence of σ on the clustering results, and then try to solve the problem caused in the first step by means of cluster expansion. The details of these two steps are presented in the following subsections respectively.

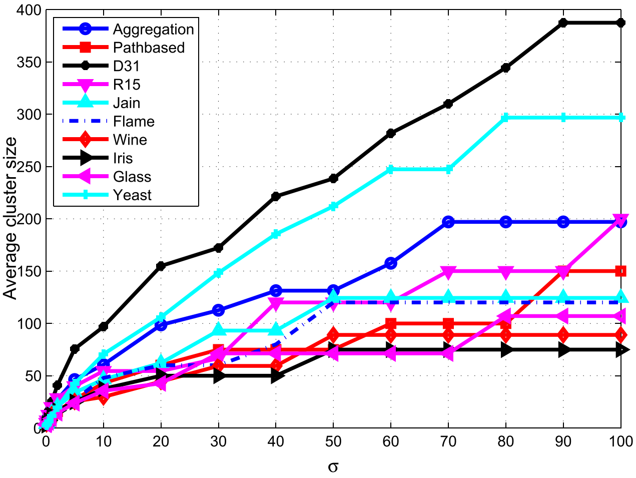


FIGURE 2. The average cluster sizes from DSets clustering with different σ's.

A. HISTOGRAM EQUALIZATION TRANSFORMATION

In the last section we see that σ influences the similarity values and then the clustering results. However, the similarity measure $s(x,y) = exp(-d(x,y)/\sigma)$ shows that σ influences only the absolute magnitude of similarity values, and it cannot change the relationship among similarity values. In other words, if $d(x_1,y_1) > d(x_2,y_2)$, then $s(x_1,y_1) < s(x_2,y_2)$ holds for any positive σ. Therefore if we sort the pairwise similarity values in the increasing order, the ordering of these similarity values keeps the same for arbitrary positive σ. This observation motivates us to transform the similarity matrix by histogram equalization, which generates the new similarity values based on the ordering of original similarity values.

Histogram equalization, as a widely used image enhancement technique, is proposed to increase the overall intensity contrast in an image based on the intensity histogram [27]. It adjusts the intensity levels so that they are distributed in the intensity range more evenly. By extending the intensity level to general scalar data, we can transform a set of data by histogram equalization in the following way. Let's say that N data to be transformed are denoted by $\gamma_p, p = 1,\cdots,N$. In the first step we quantize the data into M bins and obtain a M-bin histogram as $H = \{h_q\}, q = 1,\cdots,M$, where h_q is the number of data falling in the q-th bin. Then the data in the q-th bin are assigned the new value as

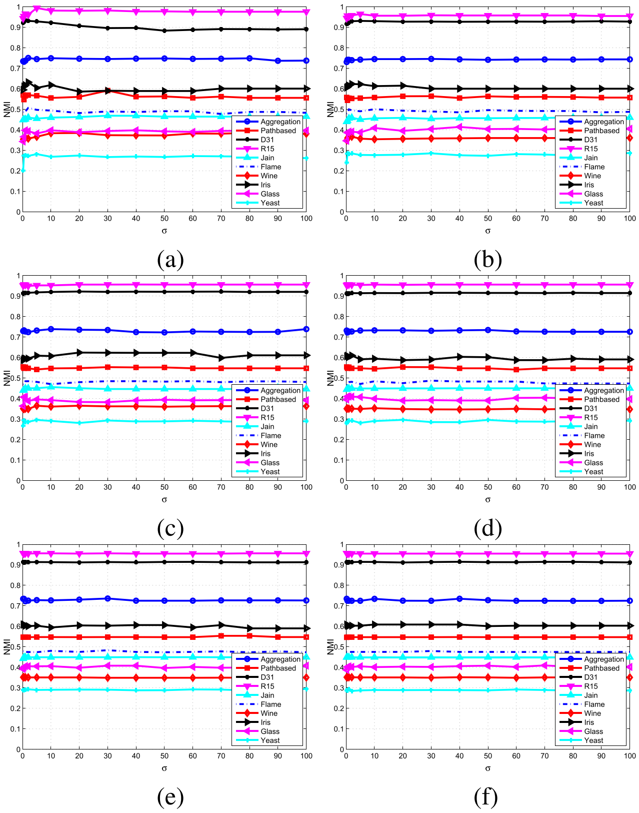
$$s'_q = \frac{1}{N}\sum_{k=1}^{q} h_k. \tag{9}$$

In Eq. (9) N is a constant, and the new value s'_q is influenced only by $\sum_{k=1}^{q} h_k$, which is the total number of data in the q-th bin and in the bins with smaller values. If M is sufficiently large that each bin contains only data of identical value, then the new value of one data is determined only by the percentage of data with equal or smaller values. In other words, the new values are influenced only by the magnitude ordering of the original values. Since $s(x, y) = exp(-d(x, y)/\sigma)$ shows that σ has no influence on the magnitude ordering, it is evident that σ does not influence the new values after histogram equalization transformation. This means that if we use histogram equalization to transform the pairwise similarity values in the similarity matrix, we are able to remove the influence of σ on the similarity matrix and then on the clustering result completely, on condition that each histogram bin contains only identical similarity values.

As the original similarity values are continuous and few of them are identical, the number of bins M usually needs to be very large in order that each bin contains only identical similarity values. This means a large computation burden in histogram equalization. With a relatively small M, it is likely that one bin contains non-identical similarity values. In this case, the different similarity values in one bin will be assigned the same new value after histogram equalization. The similarity value range of each bin is fixed, but the original similarity values change with the variance of σ. As a result, the membership of each bin also varies with σ. For example, with σ_1 the data a_{ij} and a_{mn} are in the same bin and assigned the same new value by histogram equalization, and with σ_2 their values change and they may be in different bins and are assigned different new values. This means that σ still has an influence on the new similarity values and new similarity matrix. Consequently, the clustering results are still influenced by σ. In order to use a relatively small M in histogram equalization, we need to study the influence of σ in this case and limit the influence to an acceptable level. Intuitively, with the increase of M, the number of data in a bin decreases and the influence of σ will decrease correspondingly. This argument is illustrated in Figure 3, where the influence of σ's on clustering results with different histogram bins are reported. For space reason, we only use NMI to evaluate the clustering results here.

From Figure 3 we observe that with $M \geq 100$ the influence of σ has become negligible. Based on the above observation and for efficiency reason, in this paper we adopt $M = 100$. While σ can be selected arbitrarily in this case, we use $\sigma = \overline{d}$ which generates medium similarity values. In the remaining of this paper, we use DSets-histeq to denote the DSets algorithm with the similarity matrix transformed by histogram equalization.

Although Figure 3 shows that the influence of σ can be removed effectively, it also indicates that the clustering results are usually not very good. The reason is that the clusters generated by DSets-histeq are usually smaller than the ground truth (GT), as illustrated in Figure 4. This problem should be solved in order to improve the clustering results. For space reason, in this table and following, we use D1, D2, $\cdots$, D10 to denote the ten datasets in the order of Aggregation, Pathbased, D31, R15, Jain, Flame, Wine, Iris, Glass and Yeast.



(a) (b)

(c) (d)

(e) (f)

FIGURE 3. The influence of σ's on clustering results, with different histogram bins in histogram equalization. (a) M = 10. (b) M = 20. (c) M = 50. (d) M = 100. (e) M = 200. (f) M = 500.

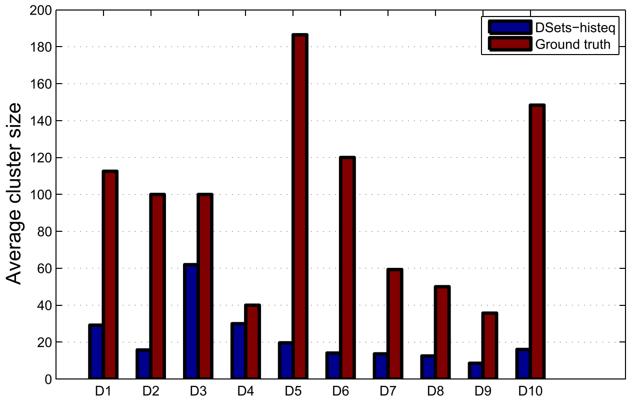
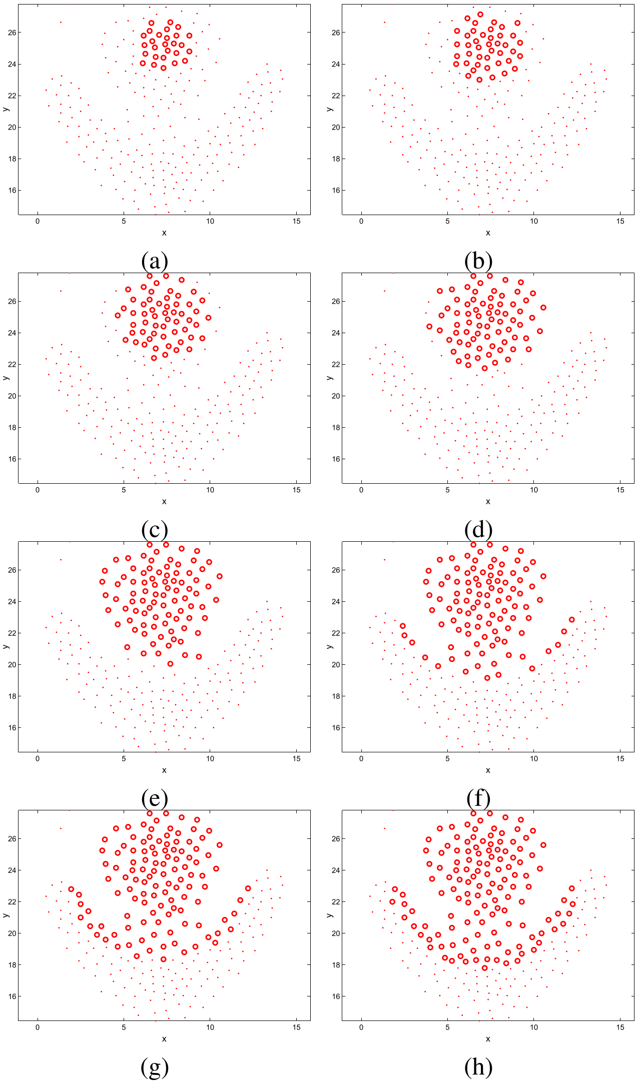


FIGURE 4. The average cluster sizes from DSets-histeq and comparison with ground truth.

B. CLUSTER EXPANSION

The property of good clustering results can be stated as high internal similarity s_{inter} and low external similarity s_{exter}, or high ratio of s_{inter} with respect to s_{exter}. As mentioned in Section II.C, the dominant set definition requires one data to be similar to all the others in the same cluster. Compared with DBSCAN where one data only needs to be



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

FIGURE 5. **The original cluster and clusters obtained after expansions. (a) Original cluster. (b) to (h) denote the clusters after one to seven expansions.**

similar to the nearest neighbors in the same cluster, this requirement is a little too strict. In addition, the histogram equalization transformation increases the contrast of similarity values in the similarity matrix. These two factors lead DSets-histeq to generate small clusters, which are usually subsets of the real clusters. In this case, while the internal similarity is high, the external similarity is also quite high. Consequently, the ratio of s_{inter} with respect to s_{exter} may not be large. In order to solve this problem and improve clustering results, we propose to expand clusters to maximize the ratio of s_{inter} with respect to s_{exter}. In the following an example is presented to illustrate the cluster expansion process.

We firstly use DSets-histeq to extract the first cluster S on the Flame dataset, as illustrated in Figure 5(a). In the next step, we repeatedly add the nearest neighbors into the cluster and calculate the ratio of s_{inter} with respect to s_{exter}. The obtained clusters and corresponding ratios are shown in Figure 5 and Figure 6, respectively. The nearest neighbors are selected based on their average similarity with the data in the cluster. The internal similarity is calculated as

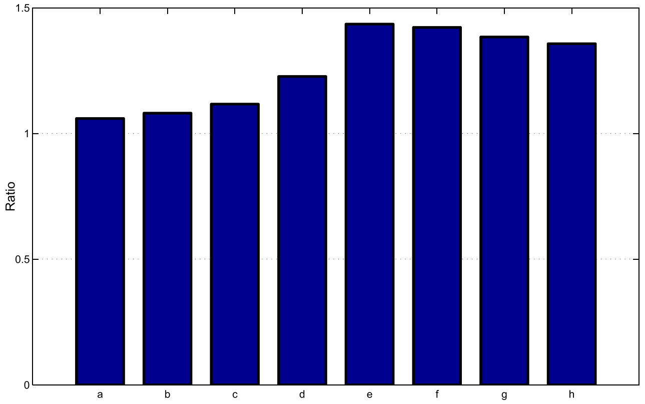


FIGURE 6. **The inter-exter similarity ratios in the cluster expansion process.**

$$s_{inter} = \frac{1}{\|S\|(\|S\|-1)} \sum_{i\in S}\sum_{j\in S} s(i,j). \tag{10}$$

In order to calculate the external similarity, we find the set S_e of nearest neighbors as another cluster. The external similarity is then defined as

$$s_{exter} = \frac{1}{\|S\|\|S_e\|} \sum_{i\in S}\sum_{j\in S_e} s(i,j) \tag{11}$$

The ratio of internal and external similarity is then represented as

$$Ratio = \frac{s_{inter}}{s_{exter}} \tag{12}$$

From the correspondence between Figure 5 and Figure 6 we see that when S is still a subset of the real cluster (Figure 5(a) to Figure 5(e)), *Ratio* keeps increasing. In contrast, Figure 5(f) to Figure 5(h) show that if S is expanded outside the real cluster, *Ratio* starts to decrease. This observation indicates that we can use the switch between the increase and decrease of *Ratio* to terminate the cluster expansion process. In practical application the differences between adjacent *Ratio*'s are usually quite small, and it may not be reliable to use one decrease to judge the switch. Therefore we use two consecutive decreases of *Ratio* as the cluster expansion termination criterion.

In summary, the whole clustering process is accomplished in the following steps.

1) Calculate the pairwise similarity matrix with $s(x, y) = exp(-d(x, y)/\sigma)$ where $\sigma = \overline{d}$.
2) Transform the pairwise similarity matrix by histogram equalization.
3) Apply the DSets algorithm to generate one cluster.
4) Add the nearest neighbors to the cluster.
5) Calculate the ratio with Eq. (12).
6) Repeat Step 4 to Step 5, until the ratio decreases twice consecutively.

7) Remove the cluster, and continue to Step 3, until all data are grouped into clusters.

IV. EXPERIMENTAL VALIDATION

Our clustering algorithm is composed of two major steps, i.e., transforming similarity matrices by histogram equalization and expanding clusters based on the ratio of internal and external similarity. We have shown the effect of the first step in previous sections. In the following we firstly evaluate the cluster expansion step, and then the whole algorithm is tested on several datasets and compared with other algorithms.

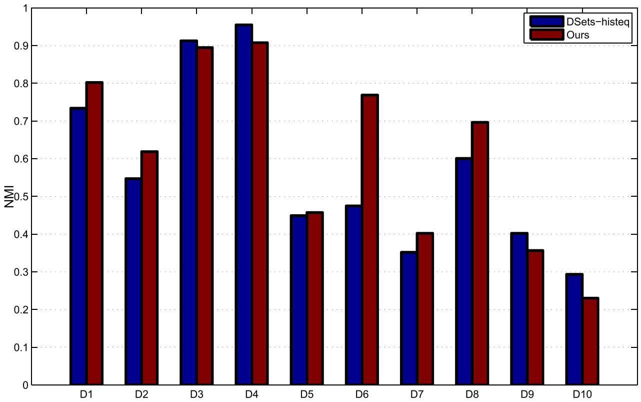


FIGURE 7. The clustering results of our algorithm and of DSets-histeq.

A. CLUSTER EXPANSION

As our algorithm includes DSets-histeq and the cluster expansion step, we firstly compare the results of our algorithm with those of DSets-histeq in Figure 7. From the comparison we see that on six out of the ten datasets, our algorithm outperforms DSets-histeq with a significant advantage. On the other four datasets, our algorithm is outperformed by DSets-histeq only slightly. This comparison confirms the effectiveness of the cluster expansion method in improving the clustering results.

B. COMPARISON

Finally, a comparison is made between our algorithm and some others, including the original DSets algorithm, NCuts, k-means, DBSCAN and AP. With the original DSets algorithm σ is selected to be $20\overline{d}$ which generates the best average accuracy. Since NCuts and k-means require the number of clusters to be specified, we feed the ground truth numbers of clusters to these two algorithms, and report the average results of five tests. For DBSCAN, we manually select $MinPts = 3$ and determine Eps with the method proposed in [35]. As the AP algorithm requires as input the preference value of all data, we use the method provided by Brendan and Delbert [8] to calculate the range $[p_{min}, p_{max}]$ of this parameter, and manually select $p_{min} + 9.3step$ as the preference value, where $step = (p_{max} - p_{min}/10)$. The comparison of the clustering results is shown in Table 2.

From Table 2 we observe that on half or more of the ten datasets, our algorithm generates the best or near-best results.

TABLE 2. Comparison of cluster results (NMI) among different algorithms on ten datasets.

	DSets	k-means	NCuts	DBSCAN	AP	Ours
Aggregation	0.86	0.83	0.77	0.92	0.80	0.80
Pathbased	0.76	0.55	0.53	0.64	0.40	0.62
D31	0.85	0.93	0.96	0.84	0.63	0.89
R15	0.83	0.96	0.99	0.87	0.81	0.91
Jain	0.43	0.36	0.33	0.73	0.46	0.46
Flame	0.60	0.40	0.44	0.83	0.57	0.77
Wine	0.41	0.43	0.36	0.05	0.38	0.40
Iris	0.65	0.76	0.74	0.75	0.79	0.70
Glass	0.31	0.43	0.39	0.40	0.33	0.36
Yeast	0.25	0.27	0.27	0.11	0.19	0.23
Average	0.59	0.59	0.58	0.61	0.54	0.61

The average result of our algorithm on the ten datasets is the best (with Rand index) or the second best (with NMI). Considering that the algorithms for comparison benefit from carefully selected parameters, we believe the effectiveness of our algorithm is validated. In addition, in the ten clustering algorithms tested in the experiments, none performs better than the others consistently.

V. DISCUSSION

Compared with DSets-histeq, our algorithm uses cluster expansion to increase cluster sizes and improve clustering results. In this sense, it is easy to understand that our algorithm performs better than DSets-histeq on six datasets, as shown in Figure 7. However, we also observe that our algorithm is outperformed by DSets-histeq on the remaining four datasets, namely D3 (D31), D4 (R15), D9 (Glass) and D10 (Yeast), indicating that cluster expansion degrades the clustering results. We discuss the possible reasons of this observation as follows. Our approach to improve clustering results by cluster expansion is based on the assumption that the clusters from DSets-histeq are subsets of the real ones. Although we have shown that DSets-histeq tends to generate small clusters which are usually subsets of real clusters, we have no guarantee that this assumption holds for all datasets. In the case that the clusters from DSets-histeq are already larger than the real ones, or one such cluster contains data from multiple real clusters, cluster expansion only degrades the clustering results further. Second, in our approach the cluster expansion is accomplished by maximizing the ratio of internal and external similarity. This criterion is suitable for spherical clusters, but may not be effective for non-spherical ones. In fact, the common internal evaluation criteria, e.g., Dunn index, Silhouette coefficient and Davies-Bouldin index, are also based on an implicit assumption that clusters are spherical. These observations show that in order to improve the clustering results further, it is necessary to design some internal evaluation criteria suitable for clusters of non-spherical shapes. Third, it is easy to understand that our imperfect cluster expansion method may not be able to find out the cluster border accurately on some datasets.

With the above discussion, we are ready to explain the observations in Figure 7. Considering that the clustering

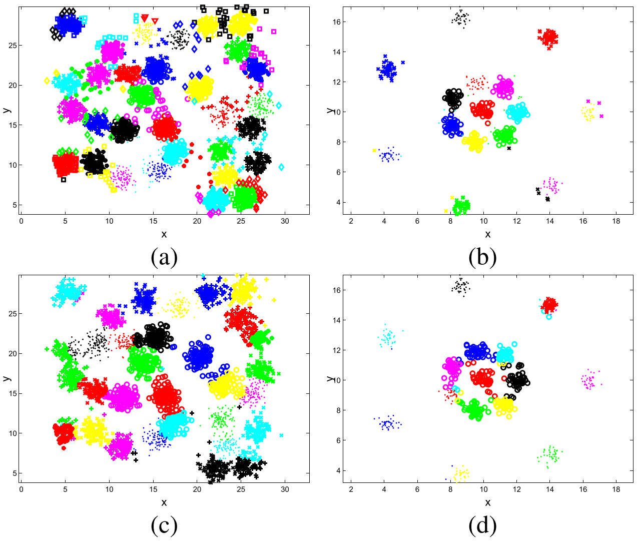


FIGURE 8. Clustering results on D31 and R15 datasets with DSets-histeq and our algorithm. The results in the top row are from DSets-histeq, and those in the bottom row are from our algorithm.

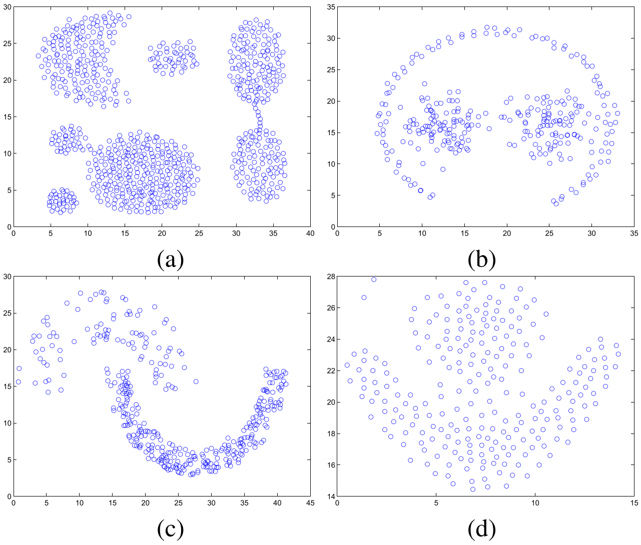


FIGURE 9. The 2D datasets. (a) Aggregation. (b) Pathbased. (c) Jain. (d) Flame.

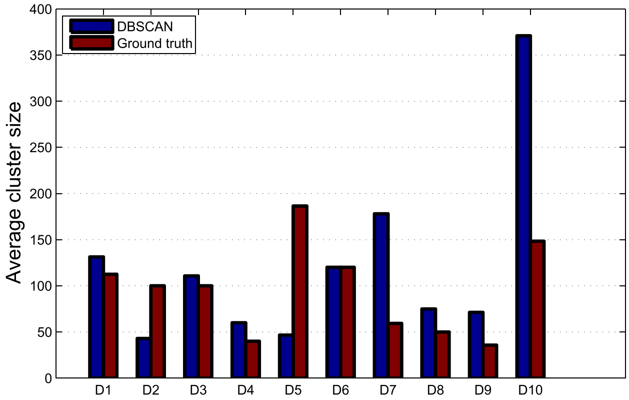


FIGURE 10. The average cluster sizes from DBSCAN and comparison with the ground truth.

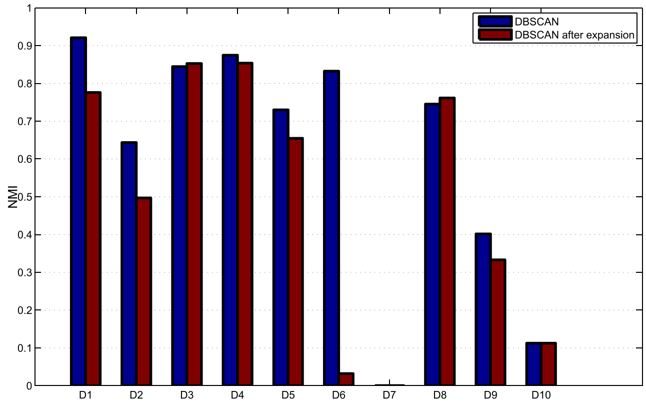


FIGURE 11. The clustering results of DBSCAN before and after cluster expansion.

results of 2D datasets can be illustrated in figures, in the following we limit out attention to the six 2D datasets, namely D1 (Aggregation), D2 (Pathbased), D3 (D31), D4 (R15), D5 (Jain) and D6 (Flame), and the observations on the remaining four non-2D datasets can be explained similarly. The clustering results on D31 and R15 with DSets-histeq and our algorithm are shown in Figure 8. Here we see that in both datasets the majority of all the clusters have been grouped correctly with DSets-histeq, and only some neighboring data are misclassified. In other words, the DSets-histeq clustering results are already quite close to ground truth, as also shown in Figure 7. In this case, our imperfect cluster expansion method fails to identify the cluster borders accurately, and some of obtained clusters cover the data of multiple real clusters (the bottom row of Figure 8), resulting in a decrease in clustering quality. The remaining four 2D datasets are shown in Figure 9. It is evident that the Aggregation, Pathbased and Flame datasets are mainly composed of spherical clusters. Correspondingly, Figure 7 shows that our algorithm performs better than DSets-histeq significantly on these three datasets. In contrast, the Jain dataset consists of non-spherical clusters, and the advantage of our algorithm over DSets-histq on this dataset is quite small. This difference on datasets with spherical and non-spherical clusters confirms that our cluster expansion method is more effective for spherical clusters. Although on some datasets cluster expansion degrades the clustering results, we also notice that the decrease in clustering quality is quite small. This shows the effectiveness of our cluster expansion method from a different perspective as it is able to limit the possible negative effective to a small level.

In uur algorithm, DSets-histeq as the first step generates clusters sequentially, and we then expand the generated (small) clusters to improve the clustering results. The majority of existing algorithms, including k-means, NCuts, AP and spectral clustering, are partitioning-based and all the clusters are obtained simultaneously from the partitioning process. As a result, their clustering results cannot be improved with cluster expansion. While with DBSCAN the clusters are obtained sequentially, this algorithm depends on local density and requires one data to be similar to its nearest neighbors only. In other words, DBSCAN has a relatively low requirement on the internal similarity of a cluster, and tends to generate large clusters. This observation is validated by the comparison of the average cluster sizes of DBSCAN with the ground truth in Figure 10, where in most cases the clusters from DBSCAN are greater than real ones. Correspondingly, expanding the clusters further is likely to degrade the clustering results, as shown in Figure 11.

VI. CONCLUSION

A cluster expansion algorithm is presented to reduce the dependence on parameters on the basis of the dominant sets algorithm. We firstly transform similarity matrices by histogram equalization to remove the influence from the similarity parameter and generate small clusters. Then a cluster expansion step is used to improve clustering results by maximizing the ratio of internal and external similarity. Experiments on then datasets show that our algorithm performs comparably to or better than some other algorithms with carefully selected parameters.

REFERENCES

[1] W. Bi, M. Cai, M. Liu, and G. Li, "A big data clustering algorithm for mitigating the risk of customer churn," *IEEE Trans. Ind. Informat.*, vol. 12, no. 3, pp. 1270–1281, Jun. 2016.

[2] L. Bai, X. Cheng, J. Liang, H. Shen, and Y. Guo, "Fast density clustering strategies based on the k-means algorithm," *Pattern Recognit.*, vol. 71, pp. 375–386, Nov. 2017.

[3] G. Guo, L. Chen, Y. Ye, and Q. Jiang, "Cluster validation method for determining the number of clusters in categorical sequences," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 2936–2948, Dec. 2017.

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. W. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.

[5] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 1999, pp. 49–60.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[7] X. Zhu, C. C. Loy, and S. Gong, "Constructing robust affinity graphs for spectral clustering," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1450–1457.

[8] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[9] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.

[10] A. B. Waluyo, D. Taniar, W. Rahayu, and B. Srinivasan, "Clustering-based index and data broadcasting for mobile nearest neighbor query processing," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 1964–1974, Nov. 2013.

[11] M. Gong, Y. Liang, J. Shi, W. Ma, and J. Ma, "Fuzzy C-means clustering with local information and kernel metric for image segmentation," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 573–584, Feb. 2013.

[12] S. Yin, H. Gao, J. Qiu, and O. Kaynak, "Descriptor reduced-order sliding mode observers design for switched systems with sensor and actuator faults," *Automatica*, vol. 76, pp. 282–292, Feb. 2017.

[13] K. Yu, S. Yu, and V. Tresp, "Soft clustering on graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1553–1560.

[14] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 167–172, Jan. 2007.

[15] A. Torsello, S. R. Bulo, and M. Pelillo, "Beyond partitions: Allowing overlapping groups in pairwise clustering," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[16] S. R. Bulò, A. Torsello, and M. Pelillo, "A game–theoretic approach to partial clique enumeration," *Image Vis. Comput.*, vol. 27, no. 7, pp. 911–922, 2009.

[17] J. Hou and M. Pelillo, "A simple feature combination method based on dominant sets," *Pattern Recognit.*, vol. 46, no. 11, pp. 3129–3139, 2013.

[18] X. Yang, H. Liu, and L. J. Latecki, "Contour-based object detection as dominant set computation," *Pattern Recognit.*, vol. 45, no. 5, pp. 1927–1936, 2012.

[19] J. Hou and W. Liu, "A parameter-independent clustering framework," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1825–1832, Apr. 2017.

[20] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, and C. Isbell, "A novel sequence representation for unsupervised analysis of human activities," *Artif. Intell.*, vol. 173, no. 14, pp. 1221–1244, 2009.

[21] A. Chakeri and L. O. Hall, "Large data clustering using quadratic programming: A comprehensive quantitative analysis," in *Proc. IEEE Int. Conf. Data Mining Workshop*, Jul. 2015, pp. 806–813.

[22] J. Hou and W. Liu, "Parameter independent clustering based on dominant sets and cluster merging," *Inf. Sci.*, vol. 405, pp. 1–17, Sep. 2017.

[23] S. Vascon, E. Z. Mequanint, M. Cristani, H. Hung, M. Pelillo, and V. Murino, "Detecting conversational groups in images and sequences: A robust game-theoretic approach," *Comput. Vis. Image Understand.*, vol. 143, pp. 11–24, Feb. 2016.

[24] R. Tripodi and M. Pelillo, "A game-theoretic approach to word sense disambiguation," *Comput. Linguistics*, vol. 43, no. 1, pp. 31–70, 2017.

[25] E. Z. Mequanint and M. Pelillo, "Interactive image segmentation using constrained dominant sets," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 278–294.

[26] J. Hou, Q. Xia, and N.-M. Qi, "Experimental study on dominant sets clustering," *IET Comput. Vis.*, vol. 9, no. 2, pp. 208–215, 2015.

[27] T. Acharya and A. K. Ray, *Image Processing: Principles and Applications*. Hoboken, NJ, USA: Wiley, 2005.

[28] J. Hou, E. Xu, L. Chi, Q. Xia, and N.-M. Qi, "Robust clustering based on dominant sets," in *Proc. Int. Conf. Pattern Recognit.*, 2014, pp. 1466–1471.

[29] S. R. Bulò, M. Pelillo, and I. M. Bomze, "Graph-based quadratic optimization: A fast evolutionary approach," *Comput. Vis. Image Understand.*, vol. 115, no. 7, pp. 984–995, 2011.

[30] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Trans. Knowl. Discovery Data*, vol. 1, no. 1, pp. 1–30, 2007.

[31] H. Chang and D. Y. Yeung, "Robust path-based spectral clustering," *Pattern Recognit.*, vol. 41, no. 1, pp. 191–203, 2008.

[32] C. J. Veenman, M. J. T. Reinders, and E. Backer, "A maximum variance cluster algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1273–1280, Sep. 2002.

[33] L. Fu and E. Medico, "Flame, a novel fuzzy clustering method for the analysis of DNA microarray data," *BMC Bioinform.*, vol. 8, no. 1, pp. 1–17, 2007.

[34] A. K. Jain and M. H. C. Law, "Data clustering: A user's dilemma," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.*, 2005, pp. 1–10.

[35] M. Daszykowski, B. Walczak, and D. L. Massart, "Looking for natural patterns in data: Part 1. Density-based approach," *Chemometrics Intell. Lab. Syst.*, vol. 56, no. 2, pp. 83–92, 2001.

JIAN HOU (M'12) received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2007. He is currently a Professor with the College of Engineering, Bohai University, Jinzhou, China. His research interests include pattern recognition, machine learning, computer vision, and image processing.

AIHUA ZHANG is currently a Professor with the College of Engineering, Bohai University, China. Her research interests lie in the field of fault diagnosis and machine learning.

• • •