Mining Quality-Aware Subspace Clusters

Ying-Ju Chen, Yi-Hong Chu, and Ming-Syan Chen

Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan, ROC
{yjchen, yihong}@arbor.ee.ntu.edu.tw, mschen@cc.ee.ntu.edu.tw

Abstract. In this paper, we study the quality issue of subspace clusters, which is an important but unsolved challenge in the literature of subspace clustering. After binning the data set into disjoint grids/regions, current solutions of subspace clustering usually invoke a grid-based apriori-like procedure to efficiently identify dense regions level by level according to the monotonic property in so defined subspace regions. A cluster in a subspace is intuitively considered as a set of dense regions that each one is connected to another dense region in the cluster. The measure of what is a dense region is successfully studied in recent years. However, the rigid definition of subspace clusters as connected regions still needs further justification in terms of the two principal measures of clustering quality, i.e., the intra-cluster similarity and the inter-cluster dissimilarity. A true cluster is likely to be separated into two or more clusters, whereas many true clusters may be merged into a fat cluster. In this paper, we propose an innovative algorithm, called the QASC algorithm (standing for Quality-Aware Subspace Clustering) to effectively discover accurate clusters. The QASC algorithm is devised as a general solution to partition dense regions into clusters and can be easily integrated into most of grid-based subspace clustering algorithms. By conducting on extensive synthetic data sets, the experimental results reveal that QASC is effective in identifying true subspace clusters.

1 Introduction

Clustering has been studied for decades and recognized as an important and valuable capability in various fields. Recently, instead of clustering in the full dimensions, research in data mining has been in the direction of finding clusters which are embedded in subspaces. The increase of research attention for subspace clustering comes from the recent report of "*the curse of dimensionality*" [1], which points out that the distances between data points will be indiscriminate in the high dimensional space. Due to the infeasibility of clustering in high dimensional data, discovering clusters in subspaces becomes the mainstream of cluster research, including the work of projected clustering [8] and subspace clustering [2][3][5].

The CLIQUE algorithm is one of the state-of-the-art methodology in the literature, which essentially relies on the monotonicity property in the partition of grid-based regions: *if a region/grid is called dense, i.e., its coverage (count*

T. Washio et al. (Eds.): PAKDD 2008, LNAI 5012, pp. 53–63, 2008.

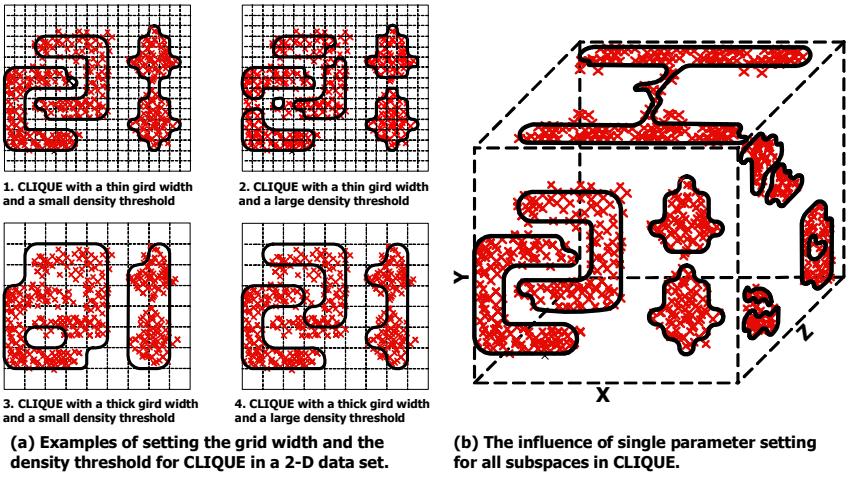

Fig. 1. Examples of quality issues in subspace clustering

of points) exceeds a specified threshold, all of its projection units will be alos dense. Therefore, after binning input data into disjointed grids according to the coordinate of each data point, Apriori-based manners are able to efficiently identify dense grids level by level.

The measure of what is a dense region and the issue of how to efficiently and precisely identify dense regions have been comprehensively studied in recent years [2][4][5]. However, identifying clusters from connected dense grids, is deemed reasonable but does not be systematically evaluated yet. In fact, the rigid definition of subspace clusters as connected grids needs further justification in terms of two general criteria of clustering quality: (1) inter-cluster dissimilarity[1]; and (2) intra-cluster similarity[2]. We note that rashly connecting dense grids as clusters inevitably faces the compromise between inter-cluster dissimilarity and intra-cluster similarity, since the naive approach will amplify the side-effect from the misadjustment of two subtle input parameters, i.e., (1) the binning width of a grid and (2) the density threshold for identifying whether a region/grid is dense. With an inappropriate parameter setting, a true cluster is likely to be separated into two or more clusters, whereas many true clusters may be merged into a fat but improper cluster.

Consider the illustrative examples shown in Figure 1(a), which contain four situations in a two-dimensional space with different input parameters in CLIQUE. It is clear to see that different parameter settings result in highly divergent results when we straightforwardly link dense grids and construct clusters. Since

[1] The inter-cluster dissimilarity is used to reflect dissimilarity between two clusters. Different clusters are generally considered with dissimilar behavior and characters.

[2] The intra-cluster similarity refers to the measure of how similar the members in a cluster are. Intuitively, data within a valid cluster are more similar to each other than to a member belonging to a different cluster.

dense grids may distribute apart from each other when the connectivity between dense grids is relatively sparse, clusters could be separated into lots of slivers in CLIQUE, such as the case in Figure 1(a).2, or the shapes of clusters could be distorted, such as the case in Figure 1(a).4. As a result, the inter-cluster dimssimilarity is strikingly sacrificed. On the other hand, true clusters could be merged into a few fat clusters when we have the crowded connection between dense grids, such as the result in Figure 1(a).1 and Figure 1(a).3. In such cases, we have the undesired loss of intra-cluster similarity in the clustering result.

Figure 1(b) illustrates another critical limitation in current subspace clustering algorithms. Essentially, users could identify a set of parameters which is able to precisely discover all clusters embedded in a subspace, such as the result in the 2-dimensional subspace $\{X, Y\}$ shown in Figure 1(b). However, there are numerous subspaces and using the same parameter setting is difficult to capture the best clustering characteristics for different subspaces due to variety of their distributions. Consider the 2-D subspaces $\{X, Z\}$ and $\{Y, Z\}$ in Figure 1(b) as examples, where the result in $\{X, Z\}$ is expected to have two separated clusters without linkages, and the result in $\{Y, Z\}$ is expected to have three clusters with near-circular shapes instead of a set of small clusters with irregular shapes.

As a result, we present in this paper an approach, called *QASC* (standing for **Q**uality-**A**ware **S**ubspace **C**lustering) to accurately construct subspace clusters from dense grids. Specifically, in order to conserve data characteristics within each subspace clusters, *QASC* takes the data distribution into consideration. Given a set of dense grids, *QASC* is devised as a two-phase algorithm to merge dense grids: (1) dense grids are partitioned into numerous small groups, where neighbor grids are located in the same group iff they are identified belonging to the same area influenced by a density function; (2) deliberately merge these small groups according to their distances and density functions by a hierarchical clustering manner.

The remaining of the paper is organized as follows. In Section 2, related works on subspace clustering are presented. In Section 3, we give the model and algorithm of QASC. Section 4 presents the experimental results. The paper concludes with Section 5.

2 Related Works

Without loss of generality, previous works on density-based subspace clustering for high dimensional data can be classified into two categories according to whether the grid structure is applied or not. Most of these algorithms utilize the grid structure, and the CLIQUE algorithm is the representative of such grid-based algorithms. On the other hand, a few works, e.g., the SUBCLU algorithm, can identify subspace clusters without use of grids.

Specifically, in the first step of CLIQUE, the data space is binned into equi-sized and axis-parallel units, where the width ξ of each dimension of a unit is one user-specified parameter. Afterward, the second step of CLIQUE exploits an apriori-like method to recursively identify all dense units in a bottom-up

way, where a dense unit is a unit whose density exceeds another user-specified threshold τ.

The use of grids can greatly reduce the computational complexity [6]. However, CLIQUE inevitably incurs many limitations from (1) using the support as a measure of interesting grids and (2) setting the subtle grid width. Consequently, the SUBCLU algorithm [3] and its extension utilize the idea of density-connected clusters from the DBSCAN algorithm without the use of grids. Giving two parameters ϵ and m in SUBCLU, the core objects are defined as the data points containing at least m data points in their ϵ-neighborhood. Since the definition of core objects also has the monotonicity property, clusters can be considered as a number of density-connected core objects with their surrounding objects, and identified in a bottom-up manner like CLIQUE. In general, SUBCLU can avoid the limitations of grids. However, the computation is higher than grid-based solutions. In addition, it also leaves the user with the responsibility of selecting subtle parameters. Even though users can empirically set parameter values that will lead to the discovery of acceptable clusters in a subspace, SUBCLU also has the problem illustrated in Figure 1(b) that clustering quality in other subspaces may be strikingly unsatisfactory.

Several variants of CLIQUE have been proposed to resolve the limitation of using the support as the measure of interesting grids. The ENCLUS algorithm in [2] utilizes entropy as a measure for subspace clusters instead of using support. The basic idea behind ENCLUS is that entropy of any subspace with clusters is higher than that of any subspace without clusters. The SCHISM algorithm is proposed to discover statistically "interesting" subspace clusters, where a cluster is interesting if the number of points it contains is statistically significantly higher than the number in the uniform distribution according to Chernoff-Hoeffding bound [7]. In addition, the MAFIA algorithm solves another limitation in CLIQUE. It uses adaptive, variable-sized grids whose widths are determined according to the distribution of data in each dimension [5]. As such, the side-effect from the rigid setting of grid widths in CLIQUE can be minimized. However, these new algorithms all merge dense/interesting grids as the same as CLIQUE. Depending on the connectivity between dense grids, they will face the same trade-off between inter-cluster dissimilarity and intra-cluster similarity in different subspaces as we show in Figure 1.

3 Quality Aware Subspace Clustering

3.1 Problem Description

We first introduce the notations used hereafter and then formalize the problem. Without loss of generality, we formalize the grid-based model by following the definition in CLIQUE. Specifically, let $S = A_1 \times A_2 \times ... \times A_d$ be the d-dimensional data space formed by the d data attributes. A k-dimensional subspace is the space with the k dimensions drawn from the d attributes, where $k \leq d$.

In the grid-based subspace clustering, the data space S is first partitioned into a number of non-overlapping rectangular units by dividing each attribute

into δ intervals, where δ is an input parameter. Consider the projection of the dataset in a k-dimensional subspace. A *"k-dimensional grid", u* is defined as the intersection of one interval from each of the k attributes, and the *density,* or said support*,* of u is defined as the number of data points contained in u. In CLIQUE, a grid is said a *dense grid* if its density exceeds a threshold τ, where τ is called *"the density threshold"*. Note that the definition of density grids is different between various subspace clustering algorithms, but subspace cluster is generally considered as disjointed sets of *dense grids* in CLIQUE and all its successors.

3.2 The QASC Algorithm

We then describe our algorithm, called QASC (the **Q**uality-**A**ware **S**ubspace **C**lustering algorithm), to deliver high-quality subspace clusters while considering the generality of the proposed model. We aim at improving the strategy of merging grids for the generality issue while conserving two general criteria of clustering quality, i.e., inter-cluster dissimilarity and intra-cluster similarity. To achieve this, the data distribution is taken into account. The basic idea behind our model is to construct small and disjointed groups of dense grids initially, where grids in each group are influenced by the same density function. Therefore, we are able to guarantee the intra-cluster similarity in the first phase. Afterward, we merge groups for improving the inter-cluster dissimilarity. We then formally present these two steps in the following sections, respectively.

Phase I of QASC: Identify Seed Clusters. The first step of QASC is to identify highly condensed group of dense grids, called seed clusters in this paper. We first have to present necessary definitions before introducing the solution to identify seed clusters.

Definition 1 (Grid Distance): *Suppose that* $V_y = [a_1, a_2, ..., a_k]$ *represents the center vector of grid* y *in the* k*-dimensional space* S^k*, where* a_i *denotes its index in the* i*-th dimension in the grid coordinates. The grid distance between grid* y *and grid* y' *in the* k*-dimensional space* S^k *is defined as the normalized Manhattan distance in the grid coordinates:*

$$Dist(y, y') = |V_y - V_{y'}| .$$

According to the definition, a grid y' is said a neighbor grid of y in S^k if $Dist(y, y') = 1$.

Definition 2 (Seed Gird): *Given the set of dense grids* D *in the* k*-dimensional space* S^k*, a grid* g *is called a seed grid iff its density* $sup(g)$ *is larger than the density of any of its neighbor grids in* D*.*

Essentially, a seed grid is a local maximum in terms of the density in the k-dimensional space, and we are able to identify the set of seed grids in each subspace by a hill-climbing procedure.

Definition 3 (Density Function): *A density function of a grid* y *wrt a seed grid* y_{sg} *in the* k*-dimensional space* S^k *is a function* $f(y_{sg}, y) : S^d \rightarrow R_0^+$ *which shows the degree of the influence from* y_{sg} *in* y*, and*

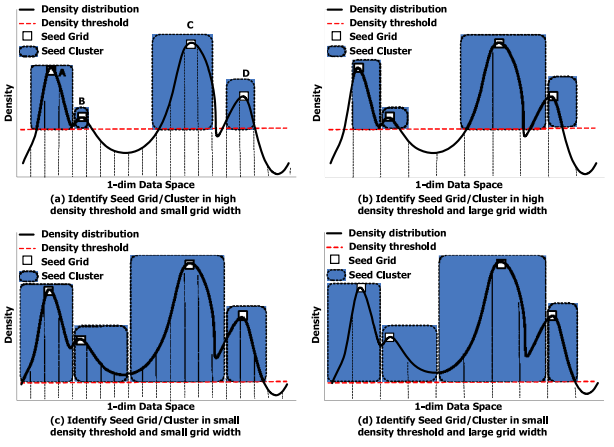

Fig. 2. Illustration of identifying seed grids and seed clusters in the 1-dim space with different parameters

$$f(y_{sg}, y) = \begin{cases} x,\ x > 0, & \text{if } y \text{ is influenced by } y_{sg} \\ 0, & \text{if } y \text{ is not influenced by } y_{sg} \end{cases}.$$

In principle, the density function can be arbitrary. However, to conserve the nature characteristics in the input data without the assumption of the data distribution, the density function is specified according to the support distribution:

$$f(y_{sg}, y) = \begin{cases} \frac{sup(y)}{sup(y_{sg})}, & \text{if } sup(y_{sg}) > sup(y), \\ & \exists y' \in D : sup(y') \geq sup(y), f(y_{sg}, y') > 0, Dist(y, y') = 1 \\ 0, & \text{else} \end{cases}.$$

Based on the foregoing, we can define the seed cluster, which is used to denote the region influenced by a density function:

Definition 4 (Seed Cluster): *Given the set of dense grids D in the k-dimensional space S^k, a seed cluster c_i wrt a seed grid y_{sg} is the maximum set of dense grids in which each grid y has $f(y_{sg}, y) > 0$, i.e., $c_i = y_{sg} \cup \{\forall y \in D \,|\, f(y_{sg}, y) > 0\}$.*

Figure 2 shows the identification of seed grids and seed clusters, where these sets of dense grids in Figures 2(a)˜Figure 2(d) are discovered with different parameters in CLIQUE. Clearly, a seed grid, e.g., grid A, B, C, or D, in Figure 2(a), has a local maximum density in the density distribution. In addition, a seed cluster wrt a seed grid y_{sg} covers a set of grids surrounding y_{sg} which are with the same distribution trend, indicating that grids within a seed cluster are highly condensed. Clearly, seed clusters can be considered as a set of most strictly defined clusters and the intra-cluster similarity can be entirely conserved in seed clusters.

Note that seed clusters inherently cannot contain grids with the density smaller than the density threshold in CLIQUE even though these grids may

satisfy the definition of density function. It is the natural limitation from the process of identifying dense grids. Nevertheless, as can be seen in Figure 2, four major seed clusters are all identified in various situations, showing the identification of seed clusters can robustly distinguish characteristics in groups of grids. On the other hand, clusters cannot be separated in Figure 2(c) and Figure 2(d) if we rashly connect dense grids into subspace clusters. The intra-cluster similarity is inevitably sacrificed.

The whole procedure to identify seed clusters in a subspace is outlined in Procedure *Seed_Identify()*. Specifically, the given set of dense grids should be sorted in order of decreasing grid density. Therefore, we can identify the seed grid from the root of the list and utilize a hill-climbing manner to search if a grid belongs to the generated seed grid. If a connected grid y_i is identified satisfying Definition 4, we set $y_i.cluster$ pointed to the corresponding seed cluster. The next grid in the sorted list is skipped if it has been identified belonging to a seed cluster. Otherwise, the procedure is iteratively executed until we have identified the cluster index for each grid. Finally, the set of seed clusters is returned.

Procedure: *Seed_ Identify()*:
Input:
dense grids $D = \{y_1, y_2, ..., y_m\}$
Output:
seed cluster $C = \{c_1, c_2, .., c_n\}$
1. $S\_D := Sort(D)$; /*sort dense grids according to the density*/
2. *for* each dense grid $sy_i \in S\_D$ *do*
3. *if* $sy_i.cluster = NULL$ *then*
4. let c_j be a new seed cluster;
5. $c_j.seed\_grid = sy_i$;
6. $sy_i.cluster = c_j$;
7. $hill\_climbing(sy_i, c_j, sy_i.density)$;
8. $C = C \cup c_j$;
9. *end if*
10. *end for*

Procedure: *hill_ climbing()*:
Input:
Dense grid y_i; Seed Cluster c_j; Integer *count*;
1. *if* ($y_i.density \leq count$ = true) *then*
2. $y_i.cluster = c_j$; /*identify that y_i belongs to seed cluster c_j*/
3. *for* each dimension a_t of grid y_i *do*
4. y_{left} = Left_Neighbor(y_i, a_t); /*return the left grid wrt the dimension a_t*/
5. *if* ($y_{left} \neq NULL$) and ($y_{left}.cluster = NULL$)
6. hill_climbing(y_{left}, c_j, $y_i.density$);
7. y_{right} = Right_Neighbor(y_i, a_t); /*return the right grid wrt the dimension a_t*/
8. *if* ($y_{right} \neq NULL$) and ($y_{right}.cluster = NULL$)
9. hill_climbing(y_{right}, c_j, $y_i.density$);
10. *end for*
11. *end if*

Phase II of QASC: Merge Seed Clusters. In essence, the seed clusters conserve the intra-cluster similarity. The inter-cluster dissimilarity is not considered yet. As shown in Figure 2(a), it is expected that seed clusters A and B belong to the same cluster because they have the same trend of distribution and

are quite close to each other. Similarly, seed clusters C and D follow the same distribution. Note that the gap with the grid distance equal to one between seed clusters C and D may occur due to noise and the choice of the grid cutting-line. It is desired to have the clustering result with only two clusters in terms of both the intra-cluster similarity and the inter-cluster dissimilarity.

The second step of QASC is thus to deliberately merge seed clusters by a hierarchical clustering manner, where the distance between clusters is taken into consideration. Here we define the cluster grid distance first.

Definition 5 (Cluster Grid Distance): *Given two clusters c_i and c_j, the cluster grid distance between c_i and c_j is defined as*

$$CDist(c_i, c_j) = \min \{Dist(y_i, y_j) \,|y_i \in c_i, y_j \in c_j \} .$$

The general criterion to merge two seed clusters is that they should be close to each other, i.e., they have the small $CDist(c_i, c_j)$. In addition, their seed grids should also be close to each other and the difference of the cluster sizes should be significantly large, meaning that they are likely to belong to the same distribution. As such, we build a global heap for merging clusters. The heap is sorted by the weight defined as:

$$weight(c_i, c_j) = \frac{size\_ratio(c_i, c_j)}{MinSeedGridDist(c_i, c_j)} \times \frac{1}{CDist(c_i, c_j)},$$

where $size\_ratio(c_i, c_j) = \max\{\frac{|c_i|}{|c_j|}, \frac{|c_j|}{|c_i|}\}$, and $|c_i|$ and $|c_j|$ are the number of points in clusters c_i and c_j, respectively. In addition, $MinSeedGridDist(c_i, c_j) = \min\{Dist(y_i, y_j)\}$, where y_i is a seed grid in c_i and y_j is a seed grid in c_j.

Importantly, clusters with a quite large $CDist(c_i, c_j)$ are not permitted to be merged even though $weight(c_i, c_j)$ is large. Note that it is reasonable to consider merging clusters with a small distance gap such as the example of seed clusters C and D in Figure 2(a). It is sufficient to avoid the influence from noise or the choice of the grid cutting-line if we permit a tolerant grid distance equal to one. As such, the prerequisite to insert the cluster pair into the heap is $CDist(c_i, c_j) \leq 2k$,where k is the number of dimensions in the subspace S^k.

The procedure in QASC to hierarchically merge clusters is outlined in Procedure *Seed_Merge()*, where the input is the set of seed clusters in S^k. Note that while two clusters c_i and c_j are merged, all information of cluster pairs in the heap related to c_i and c_j should be updated according to their new weight value.

Another criterion to determine if two clusters should be merged is shown in Line 11 in *Seed_Merge()*. Essentially, it is not desired to merge two clusters if they have similar cluster sizes because they are difficult to follow a single distribution trend. We set $\delta = 1.2$ in default to ensure the merged clusters are of variant sizes. Finally, the set of remaining clusters are returned when the heap is empty.

Procedure: *Seed_ Merge():*
Input:
Seed clusters $C = \{c_1, c_2, ..., c_m\}$ in the subspace S^k
Output:
Subspace clusters in the subspace S^k
1. *for* each seed cluster $c_i \in c$ *do*
2. *for* each seed cluster $c_j \in c, c_j \neq c_i$ *do*
3. *if* $CDist(c_i, c_j) \leq 2 \times k$ *then*
4. $weight = \frac{size\_ratio(c_i,c_j)}{MinSeedGridDist(c_i,c_j)} \times \frac{1}{CDist(c_i,c_j)}$;
5. $insertHeap(c_i, c_j, weight)$; /*insert the pair c_i, c_j in the heap sorted by the weight value*/
6. *end if*
7. *end for*
8. *end for*
9. *while* $(Heap \neq NULL)$ *do*
10. $\{c_i, c_j\} = popHeapHead(Heap)$;
11. *if* $size\_ratio(c_i, c_j) \geq \delta$ *then*
12. $c_i = c_i \cup c_j$;
13. remove c_j from C;
14. $QueuesUpdate(c_i, c_j)$;
15. *end if*
16. *end while*

4 Experimental Studies

We assess the result of QASC in Windows XP professional platform with 1Gb memory and 1.7G P4-CPU. In this section, we call the methodology to rashly merge dense grids as the *naive* approach, which is used in CLIQUE and all grid-based subspace clustering algorithms. For fair comparison, we generate dense grids by the first step in CLIQUE for QASC and the naive approach. Note that our goal is to provide an effective approach for merging grids, and the current grid-based subspace algorithms all utilize the naive approach. The benefit from QASC for these variant algorithms is expected if we can gain good clustering quality for CLIQUE. All necessary codes are implemented by Java and complied by Sun jdk1.5.

Note that various approaches to identify dense grids in subspaces introduce various parameters which would affect the clustering quality. We study the sensitivity of the QASC algorithm and the naive algorithm in various parameter setting of CLIQUE. For visualization reasons, the sensitivity analysis is studied in two dimensional spaces as the evaluation method used in traditional clustering algorithms. The result of the first study is shown in Figure 3, where a synthetic data with 6,547 points is used. Note that CLIQUE introduces two parameters, i.e., (1) the number of grids in each dimensions and (2) the density threshold, which are specified as "grid" and "minsup" in figures. Clearly, two clusters with similar diamond-like shapes are expected in the clustering results. However, the naive approach cannot capture the best result in this datasets wrt different parameter setting of CLIQUE. Figure 3(a) shows that the naive approach tends to merge clusters in high connectivity between dense grids, whereas Figures 3(b) and 3(c) show that many clusters are reported if dense grids distribute sparsely.

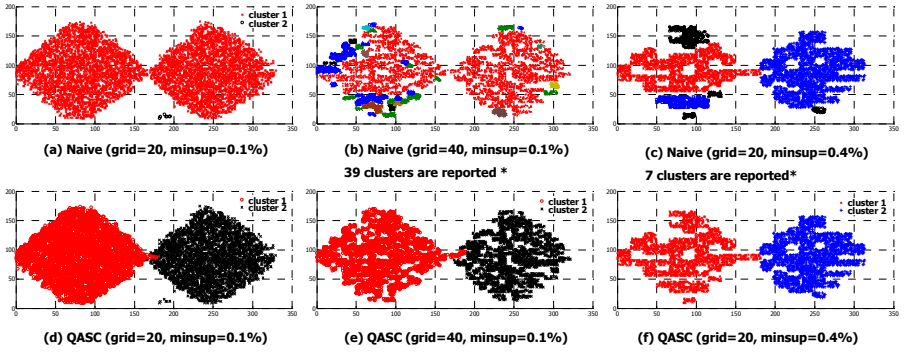

Fig. 3. The sensitivity studies on different parameter setting of CLIQUE

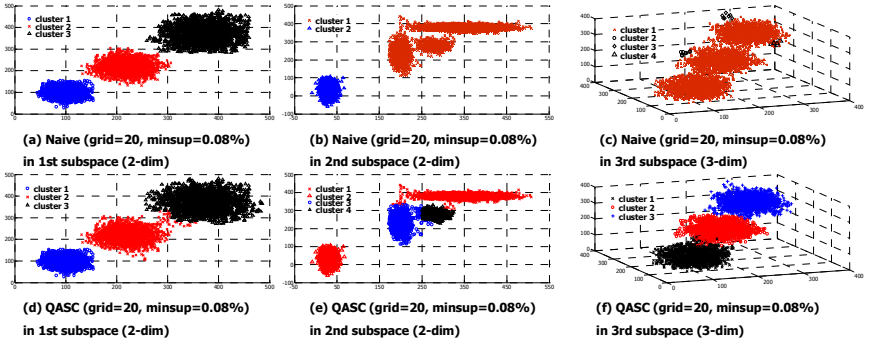

Fig. 4. Results of subspace clusters in different subspaces

On the other hand, Figures 3(d)˜(f) show that QASC results in acceptable results with two expected clusters. Specifically, the two clusters are separated in Figure 3(d) because QASC does not merge two clusters with similar sizes. In addition, it is worth mentioning that QASC permits the combination of clusters when they are distributed with a gap equal to one. Therefore, QASC can report two acceptable clusters, as shown in Figure 3(f), to avoid the side-effect from the improper parameter setting of subspace clustering algorithms, indicating the robustness of QASC.

We study the sensitivity issue in another 7-dimensional synthetic data with 6,500 points. The data is generated by embedding clusters in two 2-dimensional spaces and a 3-dimensional space. The clustering results in these subspaces are shown in Figure 4. In this case, we set grid=20 and the density threshold equal to 0.08%, which is able to correctly retrieve three clusters in the first subspace for the naive approach. However, similar to the example illustrated in Figure 1(b), this parameter setting is difficult to make correct clustering result for other subspaces. In contrast, QASC can retrieve accurate subspace clusters in other subspaces since the data distribution is taken into consideration.

5 Conclusions

In this paper, we proposed an effective algorithm, QASC, to merge dense grids for generating high-quality subspace clusters. QASC is devised as a two-step method, where the first step generates seed clusters with high intra-cluster similarity and the second step deliberately merges seed clusters to construct subspace clusters with high inter-cluster dissimilarity. QASC is devised as a general approach to merge dense/interesting grids, and can be easily integrated into most of grid-based subspace clustering algorithms in place of the naive approach of rashly connecting dense grids as clusters. We complement our analytical and algorithmic results by a thorough empirical study, and show that QASC can retrieve high-quality subspace clusters in various subspaces, demonstrating its prominent advantages to be a practicable component for subspace clustering.

References

1. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is Nearest Neighbor Meaningful? In: Proc. of ICDT Conference (1999)
2. Cheng, C.H., Fu, A.W., Zhang, Y.: Entropy-Based Subspace Clustering for Mining Numerical Data. ACM SIGKDD (1999)
3. Kailing, K., Kriegel, H.-P., Kroger, P.: Density-Connected Subspace Clustering for High-Dimensional Data. In: SDM (2004)
4. Kriegel, H.-P., Kroger, P., Renz, M., Wurst, S.: A generic framework for efficient subspace clustering of high-dimensional data. In: IEEE ICDM (2005)
5. Nagesh, H., Goil, S., Choudhary, A.: Adaptive grids for clustering massive data sets. In: Proc. of the SIAM Intern'l Conference on Data Mining (SDM) (2001)
6. Parsons, L., Haque, E., Liu, H.: Subspace Clustering for High Dimensional Data: A Review. ACM SIGKDD Explorations Newsletter (2004)
7. Sequeira, K., Zaki, M.: Schism: A new approach for interesting subspace mining. In: Proc. of the IEEE 4th Intern'l Conf. on Data Mining (ICDM) (2004)
8. Yip, K.Y., Cheung, D.W., Ng, M.K.: Harp: A practical projected clustering algorithm. IEEE Trans. Knowl. Data Eng. 16(11) (2004)