Subspace Clustering with Distance-density Function and Entropy in High-dimensional Data

Jiwu Zhao and Stefan Conrad
Institute of Computer Science, Databases and Information Systems,
Heinrich-Heine University, Universitaetsstr. 1, 40225 Duesseldorf, Germany

Keywords: Subspace Clustering, Density, High dimensionality, Entropy.

Abstract: Subspace clustering is an extension of traditional clustering that enables finding clusters in subspaces within a data set, which means subspace clustering is more suitable for detecting clusters in high-dimensional data sets. However, most subspace clustering methods usually require many complicated parameter settings, which are almost troublesome to determine, and therefore there are many limitations in applying these subspace clustering methods. In our previous work, we developed a subspace clustering method Automatic Subspace Clustering with Distance-Density function (ASCDD), which computes the density distribution directly in high-dimensional data sets by using just one parameter. In order to facilitate choosing the parameter in ASCDD we analyze the relation of neighborhood objects and investigate a new way of determining the range of the parameter in this article. Furthermore, we will introduce here a new method by applying entropy in detecting potential subspaces in ASCDD, which evidently reduces the complexity of detecting relevant subspaces.

1 INTRODUCTION

We usually need to investigate unknown or hidden information from raw data. Clustering techniques help us to discover interesting patterns in the data sets. Clustering methods divide the observations into groups (clusters), so that observations in the same cluster are similar, whereas those from different clusters are dissimilar. The clustering is important for data analysis in many fields, including market basket analysis, bio science, and fraud detection.

Unlike traditional clustering methods, which seek clusters only in the whole space, subspace clustering enables clustering in particular projections (subspaces) within a data set. Subspace clustering is usually applied in high-dimensional data sets.

Many famous subspace clustering algorithms can find clusters in subspaces of the data set. However, the effectivity is a problem of these algorithms. For instance, it is commonly known that the majority of the algorithms usually demand many parameter settings for subspace clustering. In addition, the determination of the input parameters is not simple. Furthermore, varying many sensitive parameters often cause very different clustering results.

In our former work, we introduced a density-based subspace clustering algorithm ASCDD (Automatic Subspace Clustering with Distance-Density function) (Zhao and Conrad, 2012). With its density function, the distribution of data is calculated directly in any subspace, and clusters are automatically explored according to the sizes of clusters. The method can be applied for differently scaled data. Moreover, the algorithm uses one parameter, which simplifies the application process. In this paper, we investigate the range of the parameter, in order to set the parameter in a proper range. Another important improvement of ASCDD is that we introduce an entropy based subspace search.

The remainder of this paper is organized as follows: In section 2, we present related work in the area of subspace clustering and some ideas from other algorithms. Section 3 describes the subspace clustering method ASCDD and presents our new ideas about choosing the parameter and subspace detection with entropy. Section 4 presents experimental studies for verifying the proposed method. Finally, section 5 is the conclusion of the paper.

2 RELATED WORK

In recent years, there has been an increasing amount of literature on subspace clustering. Surveys con-

Zhao J. and Conrad S..
Subspace Clustering with Distance-density Function and Entropy in High-dimensional Data.
DOI: 10.5220/0004486600140022
In *Proceedings of the 2nd International Conference on Data Technologies and Applications* (DATA-2013), pages 14-22
ISBN: 978-989-8565-67-9

ducted by (Parsons et al., 2004) and (Kriegel et al., 2009) have divided subspace clustering algorithms into two groups: top-down and bottom-up. Top-down methods (e.g. PROCLUS (Aggarwal et al., 1999), ORCLUS (Aggarwal and Yu, 2000), FINDIT (Woo et al., 2004), COSA (Friedman and Meulman, 2004)) use multiple iterations for improving the clustering results. Bottom-up methods (e.g. CLIQUE (Agrawal et al., 1998), ENCLUS (Cheng et al., 1999), MAFIA (Goil et al., 1999), CBF (Chang and Jin, 2002), DOC (Procopiuc et al., 2002)) firstly find clusters in low-dimensional subspaces, and then expand the searching into high dimensions. Other surveys from (Müller et al., 2009) and (Sim et al., 2012) categorize the basic subspace clustering methods generally into grid-based, clustering-oriented and density-based approaches.

The grid-based subspace clustering algorithms partition the data space into cells with grids, and generate subspace clusters by combining dense cells with big amount of objects. CLIQUE (Agrawal et al., 1998) is a typical representation of grid-based subspace clustering algorithms. It detects firstly one-dimensional subspace clusters, and combines them to find high-dimensional subspace clusters. CLIQUE has many extensions, one of them is ENCLUS (Cheng et al., 1999), which measures the entropy values for detecting potential subspaces with clusters, namely a subspace with clusters has lower entropy than a subspace without clusters. The entropy calculation requires density which is calculated as follows: Each dimension is divided into cells, and the density is the proportion of objects contained in a cell to all objects. After detecting all the subspace candidates, the clustering process is similar to CLIQUE.

A clustering-oriented subspace clustering method assigns objects to k medoids (similar to k-means (MacQueen, 1967)) to form clusters with corresponding subspace. Representations of clustering-oriented subspace clustering methods are PROCLUS and its extensions, such as ORCLUS, FINDIT.

Many density-based subspace clustering approaches are based on the technique of DBSCAN (Ester et al., 1996). For example, SUBCLU (Kröger et al., 2004) as an extension of DBSCAN is intended for subspace clustering. The density of an object is counted by the number of objects in a ε-neighborhood. A cluster in a relevant subspace satisfies two properties: All objects within a cluster are density-connected with each other; If an object is density-connected to any object of a cluster, it belongs to the cluster as well.

Another density-based clustering technique such as DENCLUE (Hinneburg et al., 1998) and DENCLUE 2.0 (Hinneburg and Gabriel, 2007) use Gaussian kernel function as the abstract density function and apply hill climbing method to detect cluster centers. It is unnecessary to estimate numbers or positions of clusters, because clustering is based on the density of each point. However, the estimation of parameters such as mean and variance in DENCLUE or the iteration threshold and the percentage of the largest posteriors in DENCLUE 2.0 is still necessary.

Almost all the mentioned subspace clustering methods suffer from serious limitations of determining appropriate values of parameters. For instance, the parameters such as the numbers of clusters and subspaces of top-down methods; the bottom-up method's parameters, e.g. density, grid interval, and size of clusters. These parameters influence the iterations or clustering results, but the parameters are difficult to be determined. In order to make the subspace clustering task more practical, it is necessary to simplify the parameters.

With the motivation of facilitating the determination of parameters, a subspace clustering method ASCDD (Automatic Subspace Clustering with Distance-Density function) was introduced in our previous work. ASCDD can be applied directly in any subspace for searching clusters. Based on the density values calculated with its density function, the centers of clusters can be found easily. The idea of using a density function is inspired by DENCLUE. However, the definitions of the density functions are different. ASCDD's density function can be applied directly on any subspace. A cluster in ASCDD is explored by expanding neighbors of an object with high density. Nevertheless, the definition and searching "density-connected" neighbors are totally different from DBSCAN. The clustering process in ASCDD needs just one parameter called *DDT* (distance-density threshold) with the function of determining whether two objects are neighbors (belong to the same cluster). Since choosing a proper *DDT* is important for ASCDD, in this paper we investigate thoroughly the relation between setting the parameter *DDT* and the clustering results and develop a way to set the range of *DDT*.

Although ASCDD can be applied on any subspace directly, it is still required an effective way of choosing the right subspaces with potential clusters instead of searching each subspace. Our solution is to apply entropy on detecting the potential subspaces and to reduce the subspace searching complexity. Unlike ENCLUS, ASCDD's entropy is not calculated by applying grids, but with the help of ASCDD's density function. The "interesting subspaces" in ENCLUS are the subspaces with entropy that exceeds a parameter ω. Meanwhile, interest gain more than a threshold

ε. However, the difficulty is to choose the proper parameters for an unfamiliar data set. In order to apply entropy more simply we use another technique to locate significant subspace. The extension of entropy makes ASCDD more efficient by detecting clusters directly in subspace candidates. More details are shown in the following sections.

3 AUTOMATIC SUBSPACE CLUSTERING WITH DISTANCE-DENSITY FUNCTION (ASCDD)

Generally, a data set could be considered as a pair $(\mathcal{A}, O)$, where $\mathcal{A} = \{a_1, a_2, \cdots\}$ is a set of all attributes (dimensions) and $O = \{o_1, o_2, \cdots\}$ is a set of all objects. $o_i^{a_j}$ denotes the value of an object o_i on dimension a_j.

A subspace cluster S is also a data set and can be defined as follows:

$$S = (\widetilde{\mathcal{A}}, \widetilde{O})$$

where the subspace $\widetilde{\mathcal{A}} \subseteq \mathcal{A}$ and $\widetilde{O} \subseteq O$, and S must satisfy a particular condition $\mathcal{C}$, which is defined differently in each subspace clustering algorithm. However, a general principle of $\mathcal{C}$ is that objects in the same cluster are similar, meanwhile the objects from different clusters are dissimilar. $S^{\widetilde{\mathcal{A}}}$ indicates all subspace clusters that refer to $\widetilde{\mathcal{A}}$.

Suppose S_1, S_2 are two subspace clusters, where $S_1 = (\mathcal{A}_1, O_1)$ and $S_2 = (\mathcal{A}_2, O_2)$, the intersection of two subspace clusters is defined as follows: $S_1 \cap S_2 = (\mathcal{A}_1 \cup \mathcal{A}_2, O_1 \cap O_2)$

The subspace, objects and subspace clusters have following relations:

- If $\mathcal{A}_1 \neq \mathcal{A}_2 \vee O_1 \neq O_2 \Longrightarrow S_1 \neq S_2$, the subspace clusters that have different subspaces or objects are considered as different ones.
- If $\mathcal{A}_1 \supseteq \mathcal{A}_2 \wedge O_1 = O_2$ or $\mathcal{A}_1 = \mathcal{A}_2 \wedge O_1 \supseteq O_2 \Longrightarrow S_1 > S_2$. So if $S_1 > S_2 > \cdots > S_n$, normally only the largest subspace cluster S_1 is considered as a clustering result.

The Automatic Subspace Clustering with Distance-Density function (ASCDD) is based on its density function. The following definitions are important for the density function. The distance-density of objects o_i and o_j with regard to subspace $\widetilde{\mathcal{A}}$ is defined as follows:

$$d_{o_i,o_j}^{\widetilde{\mathcal{A}}} = \frac{1}{(r_{o_i,o_j}^{\widetilde{\mathcal{A}}}{}^2 \cdot |O| + 1)^2} \qquad (1)$$

where $r_{o_i,o_j}^{\widetilde{\mathcal{A}}}$ is the normalized Euclidean distance, which is calculated as follows: $r_{o_i,o_j}^{\widetilde{\mathcal{A}}} = \sqrt{\sum_{\forall a \in \widetilde{\mathcal{A}}} (\bar{o}_i^a - \bar{o}_j^a)^2}$. The normalization of an object o_i in one dimension a is defined as $\bar{o}_i^a = \frac{o_i^a - min(o^a)}{max(o^a) - min(o^a)}$, so $\bar{o}_i^a \in [0,1]$.

The density of an object o_i relating to all objects in subspace $\widetilde{\mathcal{A}}$ is defined as follows:

$$D_{o_i}^{\widetilde{\mathcal{A}}} = \sum_{\forall o_j} d_{o_i,o_j}^{\widetilde{\mathcal{A}}} = \sum_{\forall o_j} \frac{1}{\left(\left(r_{o_i,o_j}^{\widetilde{\mathcal{A}}}\right)^2 \cdot |O| + 1\right)^2} \qquad (2)$$

The density function of ASCDD can be considered as a distribution function, which describes the distribution smoothly. The characters of clusters are shown through the density evidently, namely the cluster center has higher density than objects at edge, and therefore the position and size of the clusters can be indicated easily. Another important feature is that the algorithm can be executed in any subspace, which is simple and convenient for clustering particular subspace. *Figure 1* shows an example of density for one dimensional subspace. The peaks are possible centers of clusters, which are the key targets of our study.

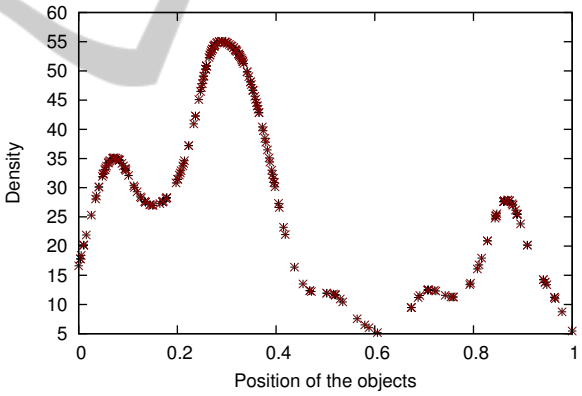

Figure 1: An example of density function.

3.1 Distance-density Threshold

Clustering is the next step after the density values are calculated. The objects in a cluster are considered as "connected" or "neighbors". A threshold for choosing proper neighbors called *DDT* (Distance-Density Threshold) is introduced in ASCDD and is important to the clustering step. The neighbors of an object o_i are defined as follows:

$$Neighbor(o_i) = \{o_j |\, d_{o_i,o_j}^{\widetilde{\mathcal{A}}} > DDT\} \qquad (3)$$

An object and its neighbors are considered in the same cluster.

Choosing a proper *DDT* is important, because *DDT* can affect the size of clusters. Only the neighbors with distance-density to the center object higher than *DDT* meet the condition. It is apparent that the larger *DDT* is chosen, the fewer neighbors will be selected. Since $d^{\widetilde{\mathcal{A}}}_{o_i,o_j}$ has a value between 0 and 1, the parameter *DDT* has also to be determined within the range $(0,1)$. However, an improper *DDT* (too small or a too big) can cause that all objects belong to one cluster or there is no cluster. So a proper value for *DDT* should be found in $(0,1)$.

We notice these two values $T^{\widetilde{\mathcal{A}}}_{min} = \min_{\forall i}\left(\max_{\forall j}(d^{\widetilde{\mathcal{A}}}_{o_i,o_j})\right)$ and $T^{\widetilde{\mathcal{A}}}_{max} = \max_{\forall i}\left(\max_{\forall j}(d^{\widetilde{\mathcal{A}}}_{o_i,o_j})\right)$ are important for the determination of *DDT*. $\max_{\forall j}(d^{\widetilde{\mathcal{A}}}_{o_i,o_j})$ is the maximum distance-density of o_i with regard to the subspace $\widetilde{\mathcal{A}}$. Each object o_i has its maximum distance-density value with an object o_j in $\widetilde{\mathcal{A}}$. Obviously, o_j has the minimum Euclidean distance to o_i. $T^{\widetilde{\mathcal{A}}}_{min}$ is the smallest maximum distance-density of all objects, and $T^{\widetilde{\mathcal{A}}}_{max}$ is the largest distance-density of all objects. If $DDT \geq T^{\widetilde{\mathcal{A}}}_{max}$, there will be no cluster result, because no object has a neighbor. If $DDT < T^{\widetilde{\mathcal{A}}}_{min}$, then all objects will be clustered as one cluster, since all objects are connected through the neighborhood. Obviously, *DDT* should be set between $T^{\widetilde{\mathcal{A}}}_{min}$ and $T^{\widetilde{\mathcal{A}}}_{max}$ to get a clustering result so *DDT* is defined as follows:

$$DDT = q \cdot T^{\widetilde{\mathcal{A}}}_{min} + (1-q) \cdot T^{\widetilde{\mathcal{A}}}_{max},\ 0 < q < 1 \qquad (4)$$

Figure 2 illustrates an example of values $T^{\widetilde{\mathcal{A}}}_{min}$ and $T^{\widetilde{\mathcal{A}}}_{max}$. We notice that *DDT* should near $T^{\widetilde{\mathcal{A}}}_{min}$ for getting a complete result. In *Figure 2*, o_{min} is the object with

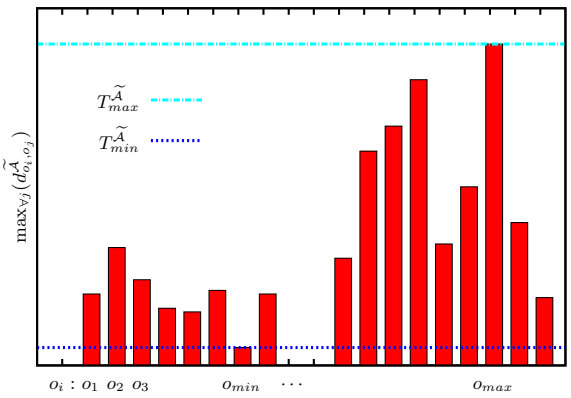

Figure 2: An example of $T^{\widetilde{\mathcal{A}}}_{min}$ and $T^{\widetilde{\mathcal{A}}}_{max}$.

distance-density $T^{\widetilde{\mathcal{A}}}_{min}$, if *DDT* is close to $T^{\widetilde{\mathcal{A}}}_{min}$, many objects with distance-density values bigger than minimum have the chances to be clustered in the next step. Conversely, if *DDT* is close to $T^{\widetilde{\mathcal{A}}}_{max}$, the amount of selected objects will be much smaller. So by setting q close to 1, ASCDD can get a relative complete result in most cases.

Notice that $T^{\widetilde{\mathcal{A}}}_{min}$ and $T^{\widetilde{\mathcal{A}}}_{max}$ are different according to $\widetilde{\mathcal{A}}$, so *DDT* has normally also different values in diverse $\widetilde{\mathcal{A}}$.

3.2 Applying Entropy for Finding Potential Subspace

Another issue is choosing the potential subspace with clusters. Our solution is to apply entropy on detecting subspaces. The authors of ENCLUS (Cheng et al., 1999) introduced a method of applying entropy for subspace clustering. However, ASCDD calculates and applies entropy in subspace clustering with a different way.

Entropy is a measure of the amount of uncertainty regarding a random variable. For a discrete random variable X with n possible outcomes $\{x_i : i = 1,\cdots,n\}$, the Shannon entropy is defined as follows: $H(X) = -\sum_{\forall i=1}^{n} p(x_i)\log p(x_i)$, where $p(\cdot)$ is the probability mass function. Obviously $H(X) > 0$. Entropy has an important property, that the variables with more uncertainty have lower entropy than the variables with less uncertainty. For the clustering purpose, we can say that a subspace with many clusters has a low entropy.

The entropy reaches maximum if all outcomes are equal.

$$H(p_1,\cdots,p_n) \leq H(\frac{1}{n},\cdots,\frac{1}{n}) = -\sum_{i=1}^{n}\frac{1}{n}\log\frac{1}{n} = \log n$$

Sometimes normalized entropy is much more convenient, because it has a range $[0,1]$ for any n. The normalized entropy is then defined as follows:

$$E(X) = \frac{H(X)}{\log n} = -\sum_{\forall i=1}^{n} p(x_i)\log p(x_i)/\log n$$

Unlike ENCLUS we apply the probability of an object o_i in $\widetilde{\mathcal{A}}$ with

$$p(o_i) = \frac{D^{\widetilde{\mathcal{A}}}_{o_i}}{\sum_{\forall i} D^{\widetilde{\mathcal{A}}}_{o_i}}$$

Obviously $0 \leq p(o_i) \leq 1$, and $\sum_{\forall i} p(o_i) = 1$, the o_i with high density has also a big value $p(o_i)$, which corresponds to the property of probability mass function.

We apply the normalized entropy $E(\widetilde{\mathcal{A}})$ in ASCDD in order to facilitate the measurement and comparison of entropy values for any subspace. As introduced above, $E(\widetilde{\mathcal{A}})$ is defined as follows:

$$E(\widetilde{\mathcal{A}}) = -\sum_{\forall i=1}^{n} p(o_i)\log p(o_i)/\log n$$

The $E(\widetilde{\mathcal{A}})$ is applicable for any subspace $\widetilde{\mathcal{A}}$. A small $E(\widetilde{\mathcal{A}})$ value indicates more uncertainties in $\widetilde{\mathcal{A}}$, which means there is more chance to detect clusters in $\widetilde{\mathcal{A}}$. A big $E(\widetilde{\mathcal{A}})$ shows that the objects distribute more uniformly. The maximum value of $E(\widetilde{\mathcal{A}})$ should be 1. However, the objects with uniform distribution do not have the same density values in ASCDD, because the densities of objects in the middle are little bigger than the densities at edge, but the difference is not large, so in this situation $E(\widetilde{\mathcal{A}})$ is smaller than 1 but very close to 1.

The entropy of low dimensional subspace and high dimensional subspace has some relations, which helps us to speculate about the potential subspaces. If the entropy of a subspace $\widetilde{\mathcal{A}}$ and a higher dimensional subspace $\widetilde{\mathcal{A}} \cup \{a_i\}$ have a relation $E(\widetilde{\mathcal{A}} \cup \{a_i\}) < E(\widetilde{\mathcal{A}})$, then the subspace $\widetilde{\mathcal{A}} \cup \{a_i\}$ has more clearly separated clusters than $\widetilde{\mathcal{A}}$. Conversely, if $E(\widetilde{\mathcal{A}} \cup \{a_i\}) > E(\widetilde{\mathcal{A}})$ it is likely that the subspace $\widetilde{\mathcal{A}} \cup \{a_i\}$ has more uniform objects than the ones in the $\widetilde{\mathcal{A}}$.

Our aim is exploring potential subspaces through the property of entropy in order to reduce the complexity. The exploring of high-dimensional potential subspace from low-dimensional subspace uses this principle: If $E(\widetilde{\mathcal{A}} \cup \{a_i\})$ is not bigger than the entropy of any subspace of $\widetilde{\mathcal{A}} \cup \{a_i\}$, we say subspace a_i can be integrated to subspace $\widetilde{\mathcal{A}}$, which is described as follows.

$$E(\widetilde{\mathcal{A}} \cup \{a_i\}) \leq \min(\{E(X) \,|\, \forall X \in \widetilde{\mathcal{A}} \cup \{a_i\}\})$$

The process of searching potential subspaces starts from one-dimensional subspace with low entropy, for instance, a_1 is a subspace candidate, if the entropy $E(a_1,a_2) < min(E(a_1),E(a_2))$ then the subspace candidate will expand from a_1 to the new subspace $\{a_1,a_2\}$. Suppose a subspace candidate $\widetilde{\mathcal{A}}$ satisfies the condition: $\forall a_i$, $E(\widetilde{\mathcal{A}}) < E(\widetilde{\mathcal{A}} \cup \{a_i\})$, then $\widetilde{\mathcal{A}}$ reaches its maximum dimension. The expansion stops when the subspace candidate reaches the maximum dimension.

Figure 3 is a simple example for subspace clustering. It is not straightforward to cluster directly in the three dimensional space, but if the objects are projected into any two dimensional subspace the clustering process will be more effective because in each

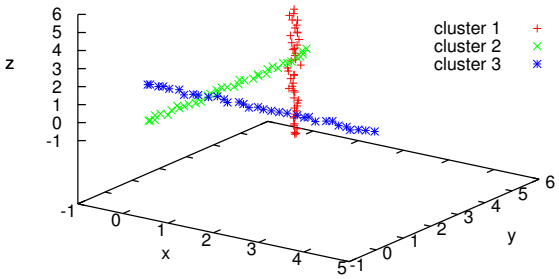

Figure 3: An example of three dimensional subspace clustering.

two dimensional subspace one cluster is much tighter than the other two clusters. Obviously, the two dimensional subspaces $\{x,y\}$, $\{y,z\}$, $\{x,z\}$ are subspace candidates. This result can also be verified through the subspace searching method. The entropy of different subspaces has relations as the follows:

$$E(x,y); E(y,z); E(x,z) < E(x); E(y); E(z) < E(x,y,z)$$

The entropy of two dimensional subspace is smaller than the entropy of one or three dimensional subspace. So each two dimensional subspace reaches the maximum dimension. The subspace searching process starts from one dimension and stops at two dimensional subspace, whereas the three dimensional space will not be considered because it has a bigger entropy value than two dimensional subspaces.

3.3 Algorithm

The clustering process of ASCDD consists of two steps. The first step is searching the potential subspaces and the second step is exploring clusters from the potential subspaces.

We use greedy strategy to search the potential subspace, which is shown in *Algorithm 1*.

Searching the potential subspaces starts from one-dimensional subspace with low entropy. A high-dimensional subspace is considered as a subspace candidate only with the principle that it has a lower entropy than all its subspaces.

Algorithm 2 illustrates the clustering process of ASCDD. The clustering process for a subspace candidate $\widetilde{\mathcal{A}}$ is divided into four steps.

I. $\forall i$, calculate $D_{o_i}^{\widetilde{\mathcal{A}}}$.

II. Take the starting object o_s that has the maximum density of current set of objects $O_{current}$.

III. Find all neighbors from o_s, and set them as a cluster S, then expand S by finding new neighbors of objects in S until no new neighbor is found.

Algorithm 1: Searching subspace.

Input: $(\mathcal{A}, O)$
Output: Subspace Candidate Set: SCS

ascending sort $E(a_i)$: $E(a_i) \leq E(a_j)$ when $i < j$
$SCS = \emptyset$
for $i = 1$ *to* $|\mathcal{A}|$ **do**
  $\mathcal{C} = \{a_i\}$
  for $j = i+1$ *to* $|\mathcal{A}|$ **do**
    $minEntropy = \min(E(\mathcal{C}), E(a_j))$
    if $E(\mathcal{C} \cup \{a_j\}) < minEntropy$ **then**
      $\mathcal{C} = \mathcal{C} \cup \{a_j\}$
  $SCS = SCS \cup \{\mathcal{C}\}$

Algorithm 2: Clustering.

Input: $(\mathcal{A}, O), SubspaceCandidateSet$
Output: Set of all clusters $\hat{S}$

$\hat{S} = \emptyset$
foreach $\widetilde{\mathcal{A}} \subseteq SubspaceCandidateSet$ **do**
  $O_{current} = O$
  $\forall i$, calculate $D_{o_i}^{\widetilde{\mathcal{A}}}$
  while $O_{current} \neq \emptyset$ **do**
    o_s has max $(D_{o_i}^{\widetilde{\mathcal{A}}}), \forall o_i \in O_{current}$
    $\widetilde{O} = Neighbor(o_s)$
    Iteration: $\forall o_i \in \widetilde{O}, Neighbor(o_i) \subseteq \widetilde{O}$
    $O_{current} = O_{current} - \widetilde{O}$
    $S = (\widetilde{\mathcal{A}}, \widetilde{O})$, $\hat{S} = \hat{S} \cup S$

IV. Remove objects in S from $O_{current}$, repeat step II until no more new cluster is found.

ASCDD could find arbitrary (convex or concave) shaped clusters through extending the neighborhood. For example, a cluster with a concave form is found as follows: ASCDD may find an object with the highest density in the cluster as the center object. Then the process of searching and adding the new neighbors to this cluster connects the objects together to reach its original concave shape. The object with highest density in a cluster is chosen as the "center" object. However, this "center" object is possibly not the geometric center of the cluster. *Figure 4* shows an example of two-dimensional clustered objects (marked with different colors) and corresponding density values of the objects. In this example, some center objects are at edges of the clusters. The objects in one cluster are all the extensions of neighborhoods from its center object.

The clusters are detected according to the order of density values of center objects one by one (from the highest density to the lowest density), which does not depend on the input order of the objects. Therefore it is not necessary to estimate the quantity of clusters in ASCDD.

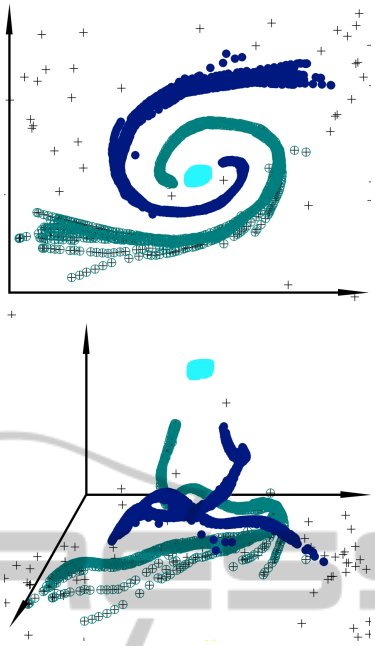

Figure 4: Two-dimensional clustered objects & Three-dimensional view of objects density.

The time complexity of ASCDD depends on the numbers of objects $|O|$ and dimensions $|\mathcal{A}|$ and subspace candidates $|SCS|$. The run-time of density calculation is $O(|O|^2)$ and the run-time of searching subspace depends on the subspace candidates, which can between $O(|\mathcal{SCS}|)$ and $O(2^{|\mathcal{A}|})$.

4 EMPIRICAL EXPERIMENTS

A set of experiments was performed to observe the effectiveness and efficiency of ASCDD, particularly, focusing on the accuracy and run-time of clustering for large quantities of data on high-dimensional spaces and the ability for searching subspaces. All experiments were carried out on a PC with 800MHz dual-core processor, 4GB RAM, Linux operating system and Java environment.

4.1 Synthetic Data

Firstly, we use synthetic data as experimental data in order to make the experiment controllable and to measure the accuracy easily. The data sets consist of 10000 objects and 100 dimensions. 20 simulated clusters are hidden in 10 different subspaces. The clusters have different forms, e.g. convex and con-

cave forms. The subspaces without clusters are filled with random objects.

We compare the results of ASCDD with different settings of the parameter *DDT*. As we discussed above, *DDT* depends on q, which is defined in *Equation 4*. So the problem of determining the *DDT* is transformed to choose a $q \in (0,1)$. Because the two extreme situations $q = 0$ and $q = 1$ cause two results respectively: no cluster object and all objects belong to one cluster. When q is close to 1, almost all cluster centers are taken into account by ASCDD, and the clustering result is more complete than the results with a small q; When q approximates to 0, some small clusters disappear, and the big clusters shrink to small ones. However, the computation time will be reduced with a small q. Generally speaking, altering q between 1 and 0 could adjust between details of clusters and run-time. *Figure 5* presents the run-

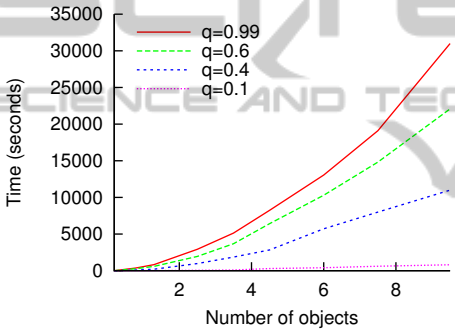

Figure 5: Run-time with different q.

time with four arbitrarily chosen q values. It is worth mentioning that the clustering results do not change much in a small range of q. In order to acquire complete clustering results, we choose $q = 0.96$ in the following experiments, where 0.96 is just a discretionary choice close to 1. Nevertheless, q can also be another value $\in (0.95, 0.98)$ because the clustering results are almost equal.

Since ENCLUS is one of the most famous subspace clustering method applying entropy, we compare ASCDD with ENCLUS in the next experiment with regard to potential subspaces and clustering results. ASCDD starts searching with the subspace with lowest entropy, and expands the subspace in higher dimensions by calculating and comparing the entropy values. Finally all expected subspaces are obtained correctly. We apply ENCLUS by setting the number of units to 285 in order to keep averagely 35 objects in each cell as the authors suggest. ENCLUS uses "entropy $< \omega$" and "interest_gain $> \varepsilon$" as the thresholds for detecting subspace candidates. However, choosing proper values for these two parameters is a challenge. We choose $\omega = 8.5$, $\varepsilon = 1$ as described in the article. ENCLUS does not find all the same subspace candidates as ASCDD, ENCLUS finds a part of expected subspaces and many non-expected subspaces, where no clusters exist. Even by altering the two parameters with different combinations in ENCLUS, the results of subspace candidates are still mixed with non-expected subspaces.

Next we compare the clustering results between ASCDD and ENCLUS. In this step ASCDD finds the defined clusters in both convex and concave forms with high precision. ENCLUS uses grid-based method by searching the clusters firstly through the grids in one dimensional subspace, and combines the clusters in high-dimensional subspace to search more clusters. In this experiment the result of ENCLUS includes just some simple convex clusters correctly. Some concave clusters are bound together as one cluster and some are separated to small clusters. Unlike ENCLUS, who has to search each low-dimensional subspace of a subspace candidate, ASCDD can directly focus on the subspace candidates for searching clusters.

The efficiency evaluation of ASCDD and ENCLUS are illustrated in *Figure 6*. This evaluation is based on subsets of the synthetic data set. ASCDD and ENCLUS use the same parameter settings as in the former experiment.

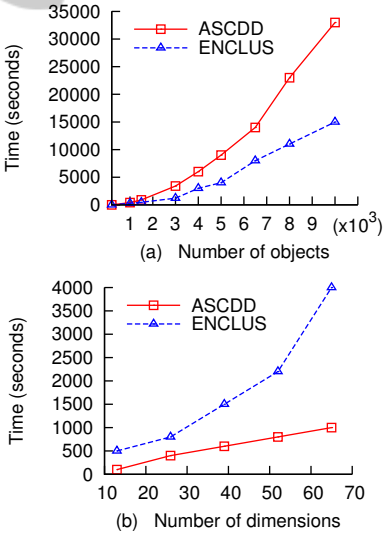

Figure 6: Run-time compared with ENCLUS.

ASCDD scales very well with an increasing dimensionality. As we can see, the run-time of ASCDD increases linearly if the number of dimensions grows. The reason is that ASCDD searches firstly only the subspace candidates, and the clustering process executes directly on high dimensional subspace candidates. ASCDD has almost the same run-time for a

Table 1: Results of ASCDD and ENCLUS on "Gas Sensor Array Drift".

Cluster	ASCDD Accuracy	Subspace	ENCLUS Accuracy	Subspace
1	68%	76, 113, 17, 4, 79, 70, 14, 68, 121, 57, 15, 6, 7, 53, 118, 12, 54, 62, 127	41%	113, 4, 79, 70, 68, 57, 15, 54, 7, 14, 53, 118, 83, 14, 73
2	67%	15, 6, 78, 49, 7, 12, 55, 63	55%	20, 6, 78, 30, 19, 7, 66, 23, 11, 50, 93
3	39%	47, 24, 107, 111, 88, 97, 99, 105	31%	88, 40, 26, 113, 105, 95, 33, 28, 16
4	68%	44, 108, 39, 47, 24, 103, 111, 88, 97, 99, 105	52%	111, 23, 108, 75, 39, 94, 47, 85
5	34%	112, 56, 120, 122, 98, 16, 35, 106, 43, 80, 36, 108, 24, 107, 88, 97, 99, 105	19%	112, 43, 106, 16, 80, 24, 74, 87, 86, 98, 19, 108, 58
6	88%	65, 9, 76, 4, 79, 70, 14, 68, 15, 6, 78, 7, 12, 39, 47, 103	59%	65, 83, 4, 68, 70, 6, 81, 14, 7, 103, 79

clustering within a subspace with no matter high or low dimension.

With increasing number of objects the run-time of ASCDD grows quadratically, which is longer than ENCLUS in this situation. The reason is that the calculation of density for one object in ASCDD involves all objects and ENCLUS works similar to CLIQUE that separates the objects into grids, which is not sensitive to amount of objects. Although the scalability of ASCDD related to the size of objects is not linear, the complexity ensures getting a complete clustering result. Of course the run-time with regard to the number of objects depends also on the parameter setting because choosing a *DDT* that yields many objects in the clustering result takes more time than with a *DDT* that involves fewer objects.

ENCLUS finds almost the same low dimensional subspace candidates, but ENCLUS is slower than ASCDD for high dimensional subspace, because ENCLUS does clustering only from low to high dimensional subspace, which takes much time than direct clustering in high dimensional subspace as ASCDD.

4.2 Real Data

The data set "Gas Sensor Array Drift" has been obtained from the UC Irvine Machine Learning Repository (Frank and Asuncion, 2010). This data set corresponds to the measurements of 16 chemical sensors utilized in simulations for drift compensation in discriminating six gas types (Ammonia, Acetaldehyde, Acetone, Ethylene, Ethanol, and Toluene) at various concentrations. The data is prepared for the chemosensor research community and artificial intelligence to develop strategies to cope with sensor/concept drift. The dataset contains 128 dimensions, 13910 measurements with six clusters (six gas types), we applied ASCDD and ENCLUS on the data without cluster labels, the results were then compared with the cluster labels. The clusters are located in different subspaces, which means the particular subspaces can specialize detecting the gas types. We illustrate some examples of the clustering result and the accuracies of data related to months one and two in *Table 1*. The accuracy is defined as the proportion of the number of correctly clustered objects to the number of objects in that cluster.

This clustering process takes 1440 seconds with ASCDD and 4410 seconds with ENCLUS. Compared with ENCLUS, ASCDD is more efficient on high-dimensional subspace and is able to detect the clusters directly on these subspaces with higher precision.

5 CONCLUSIONS

Departing from the traditional clustering methods, ASCDD is suitable for complex data with arbitrary forms. It provides useful distribution information and can be applied easily with just one simple parameter *DDT* by clustering. The clusters are detected according to their densities, which does not depend on the input order. The results of ASCDD in our experiments show high accuracy.

In this paper we improve the methods of subspace detection and parameter determination in the subspace clustering method ASCDD for high-dimensional data set. By adhibiting entropy, ASCDD is able to detect high-dimensional subspace candidates easily, where a subspace with low entropy is considered as a potential subspace. We develop a way to detect subspace candidates to reach its maximum dimensions. ASCDD can directly find clusters within the located subspace candidates. Since the clustering result and quality depend on choosing the parameter *DDT*, we investigate the *DDT* and introduce a method of choosing this parameter. The *DDT* can be chosen in accordance with the tendencies to complete clustering results or short run-time. One of our future works will be reducing the calculation time with very high number of objects.

REFERENCES

Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., and Park, J. S. (1999). Fast algorithms for projected clus-

tering. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, SIGMOD '99, pages 61–72. ACM.

Aggarwal, C. C. and Yu, P. S. (2000). Finding generalized projected clusters in high dimensional spaces. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 70–81. ACM.

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, SIGMOD '98, pages 94–105. ACM.

Chang, J.-W. and Jin, D.-S. (2002). A new cell-based clustering method for large, high-dimensional data in data mining applications. In *Proceedings of the 2002 ACM symposium on Applied computing*, SAC '02, pages 503–507. ACM.

Cheng, C.-H., Fu, A. W., and Zhang, Y. (1999). Entropy-based subspace clustering for mining numerical data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 84–93. ACM.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, volume 1996, pages 226–231. AAAI Press.

Frank, A. and Asuncion, A. (2010). UCI machine learning repository.

Friedman, J. H. and Meulman, J. J. (2004). Clustering objects on subsets of attributes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, pages 815–849.

Goil, S., Nagesh, H., and Choudhary, A. (1999). Mafia: Efficient and scalable subspace clustering for very large data sets. Technical Report CPDC-TR-9906-010, Northwestern University.

Hinneburg, A. and Gabriel, H.-H. (2007). Denclue 2.0: fast clustering based on kernel density estimation. In *Proceedings of the 7th international conference on Intelligent data analysis*, IDA'07, pages 70–80. Springer-Verlag.

Hinneburg, A., Hinneburg, E., and Keim, D. A. (1998). An efficient approach to clustering in large multimedia databases with noise. In *Proc. 4rd Int. Conf. on Knowledge Discovery and Data Mining*, pages 58–65. AAAI Press.

Kriegel, H.-P., Kröger, P., and Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3:1:1–1:58.

Kröger, P., Kriegel, H.-P., and Kailing, K. (2004). Density-connected subspace clustering for high-dimensional data. In *Proc. SIAM Int. Conf. on Data Mining (SDM'04)*, pages 246–257.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

Müller, E., Günnemann, S., Assent, I., and Seidl, T. (2009). Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB Endowment*, 2(1):1270–1281.

Parsons, L., Haque, E., and Liu, H. (2004). Subspace clustering for high dimensional data: A review. *SIGKDD Explor. Newsl.*, 6:90–105.

Procopiuc, C. M., Jones, M., Agarwal, P. K., and Murali, T. M. (2002). A monte carlo algorithm for fast projective clustering. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, SIGMOD '02, pages 418–427. ACM.

Sim, K., Gopalkrishnan, V., Zimek, A., and Cong, G. (2012). A survey on enhanced subspace clustering. *Data Mining and Knowledge Discovery*, pages 1–66.

Woo, K.-G., Lee, J.-H., Kim, M.-H., and Lee, Y.-J. (2004). Findit: a fast and intelligent subspace clustering algorithm using dimension voting. *Information and Software Technology*, 46(4):255–271.

Zhao, J. and Conrad, S. (2012). Automatic subspace clustering with density function. In *International Conference on Data Technologies and Applications*, DATA 2012, pages 63–69. SciTePress Digital Library.