Front. Electr. Electron. Eng. China 2008, 3(1): 44–48
DOI 10.1007/s11460-008-0010-x

RESEARCH ARTICLE

Kun NIU, Shubo ZHANG, Junliang CHEN

Subspace clustering through attribute clustering

Abstract Many recently proposed subspace clustering methods suffer from two severe problems. First, the algorithms typically scale exponentially with the data dimensionality or the subspace dimensionality of clusters. Second, the clustering results are often sensitive to input parameters. In this paper, a fast algorithm of subspace clustering using attribute clustering is proposed to overcome these limitations. This algorithm first filters out redundant attributes by computing the Gini coefficient. To evaluate the correlation of every two non-redundant attributes, the relation matrix of non-redundant attributes is constructed based on the relation function of two dimensional united Gini coefficients. After applying an overlapping clustering algorithm on the relation matrix, the candidate of all interesting subspaces is achieved. Finally, all subspace clusters can be derived by clustering on interesting subspaces. Experiments on both synthesis and real datasets show that the new algorithm not only achieves a significant gain of runtime and quality to find subspace clusters, but also is insensitive to input parameters.

Keywords subspace clustering, high dimensional data, attribute clustering

1 Introduction

With the rapid development of data collection technology, there are many high dimensional and large scale datasets in different areas such as biology, geography, finance and telecommunication. The dimensionality of these datasets may be over 10 or even over thousands of dimensions, bringing in a lot of difficulties to traditional clustering methods. Most clustering methods encounter challenges when the dimensionality of the dataset grows high. This is because only a small number of dimensions are usually relevant to certain clusters when the dimensionality increases. Data in the irrelevant dimensions may produce much noise and mask the real clusters from being discovered. Moreover, when dimensionality increases, data usually become increasingly sparse because the data points are likely to be located in different dimensional subspaces, which may result in the curse of dimensionality. As the number of dimensions in a dataset increases, distance measure becomes increasingly meaningless. Additional dimensions spread out the data points until they are almost equidistant from each other at very high dimensions. As an extension to feature selection, subspace clustering searches for groups of clusters within different subspaces of the same data set.

Translated from *Journal of Beijing University of Posts and Telecommunications*, 2007, 30(3): 1−5 [译自: 北京邮电大学学报]

Kun NIU (✉), Junliang CHEN
State Key Laboratory of Networking and Switching Technology, Beijing 100876, China
E-mail: niukun2006@gmail.com

Shubo ZHANG
Department of Strategy Research, China Telecom Beijing Research Institute, Beijing 100035, China

2 Subspace clustering

CLIQUE [1,2] is one of the first algorithms proposed that attempted to find clusters within subspaces of the datasets. It uses an a priori technique to find subspace clusters. ENCLUS [3] and MAFIA [4] are two extensions of CLIQUE. DOC [5] is somewhat of a hybrid method that blends the grid-based approach used by the bottom-up approaches. W-k-means [6] proposes a weight estimation method and detects subspace clusters based on current data segmentation to specify attribute weight.

The existing subspace clustering algorithms usually have two drawbacks [7]. First, all methods usually scale exponentially with the number of attributes and/or the dimensionality of the subspace clusters. Second, most subspace clustering approaches use a global density threshold for performance reasons. However, it is quite questionable that a global density threshold is applicable on clusters in subspaces of considerably different dimensionality, since density naturally decreases with increasing dimensionality.

In this paper, we present subspace clustering via attribute clustering (SCA), a novel algorithm for mining subspace clusters. It first filters out redundant attributes by

computing the Gini coefficients. To evaluate the correlation of every two non-redundant attributes, the relation matrix of non-redundant attributes is constructed based on the relation function of two dimensional united Gini coefficients. After applying overlapping clustering algorithm on the relation matrix, the candidate of all interesting subspaces is achieved. Finally, all subspace clusters can be obtained by clustering on interesting subspaces.

3 The new algorithm for subspace clustering-SCA

3.1 Problem statement

Let DB be a dataset of n objects with dimensionality d. All feature vectors have normalized values, that is, all values fall into [0,1]. Let $A=\{a_1, a_2, \ldots, a_d\}$ be the set of all attributes a_i of DB. Any subset $S \subseteq A$ is called a subspace. Different from traditional search strategy, SCA takes the method of clustering on attributes. It generates the candidate set of interesting subspaces by a clustering method which allows overlapping clusters. Moreover, a specially designed relation function is used to evaluate the relevance of every two non-redundant attributes. We introduce the steps in the following.

3.2 Filtering out redundant attributes

In the first step of SCA, an entropy-based method is proposed to filter out redundant attributes. Attributes for which it is impossible to compose any interesting subspace are called redundant attributes. According to the downward closure property of dense units [1], if there are dense units in k dimensions, there are dense units in all $(k-1)$ dimensional projections, that is, if an attribute is sparse, we can conclude that it is impossible to be included in interesting subspaces. Taking each attribute as an independent information source, SCA denotes its density as the entropy of the attribute.

Similar to the method in Ref. [3], we divide each attribute into ω intervals of equal length, so that the high-dimensional space is partitioned to form a grid. Let $X(A)$ be the set of all units in attribute A, and the density of unit $x \in X(A)$ be the percentage of data contained in x. We define the Gini value of attribute A to be $\mathrm{Gini}(A) = 1 - \sum_{i=1}^{\omega} P_i^2$, where P_i is the percentage of data contained in the ith unit. SCA takes each attribute as a discrete random variable and computes its Gini value. If the value is above the threshold $\delta = 1 - k/\omega$, then the attribute is a redundant attribute. Here k is the aggregation constant of data points, which denotes the density distribution multiple of data points with respect to uniform distribution. For example, $k = 2$ means that only 50% of grid units have data points. Obviously, ω denotes the number of intervals of equal length.

Figure 1 shows the data distributions of two attributes. In Fig. 1(a), the data points are almost uniformly distributed, and it is very uncertain where a point would lie in. The entropy is high. When the data points are closely packed in a small cluster, we know that a point is likely to fall within the small area of the cluster, and so the uncertainty and entropy will be low as shown in Fig. 1(b).

3.3 Computing the relation matrix

The key point of subspace clustering is to find all possible interesting subspaces. SCA searches for all possible interesting subspaces with the relation matrix.

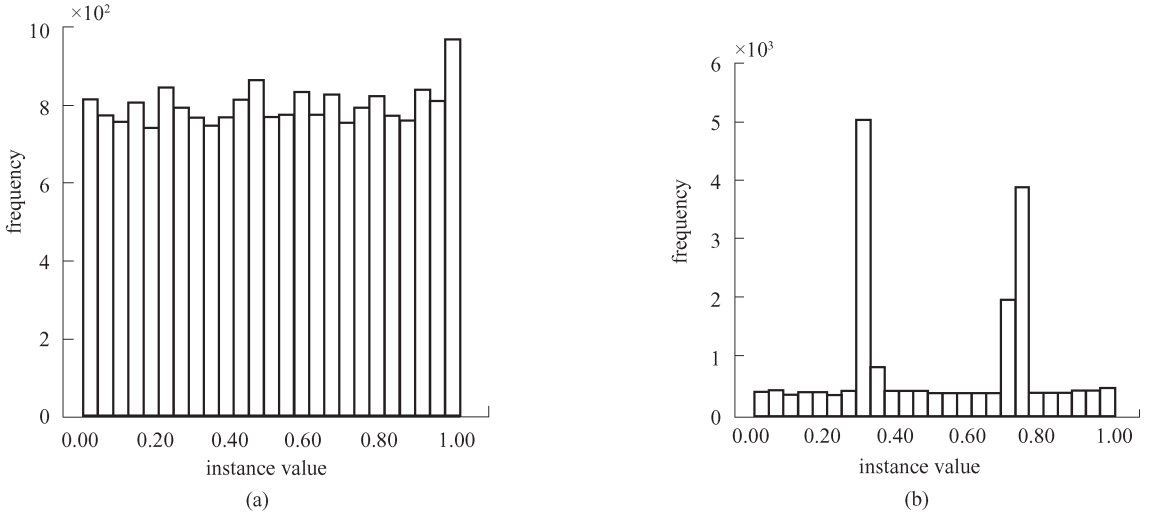

Fig. 1 Data distribution of different attributes.
(a) Data distribution of redundant attribute; (b) Data distribution of interesting attribute

After attribute filtering, we acquire all 1-dimensional (1-D) interesting subspaces. For detecting interesting subspaces within higher dimensions, we define the correlation function to evaluate the relevance of every two non-redundant attributes. If two attributes lie in the same interesting subspace, they are called "relative". We find an observation in the experiments as follows.

Let C_A be the number of clusters in attribute A and C_B be the number of clusters in attribute B, then the number of clusters in subspace $\{A, B\}$, denoted as C_{AB}, lies in [max (C_A, C_B), C_AC_B]. Generally, when A and B are in the same interesting subspace, the number of C_{AB} tends to be max (C_A, C_B). Otherwise, it tends to be C_AC_B.

The clusters in low dimensional subspace scatter with growing dimensionality. We compose two non-redundant attributes to form 2-D subspaces. If the number of clusters in a 2-D subspace is nearly the same as that in each 1-D subspace, we say that the two attributes have rather high relevance. Figure 2 illustrates the observation. In Fig. 2(a), attribute A and attribute B have two clusters each. After composition, there are still two clusters in the subspace $\{A, B\}$. In Fig. 2(b), attribute A and attribute C also have two clusters each. However, there are four clusters in the subspace $\{A, C\}$, which means that attribute A is relevant to attribute B but irrelevant to attribute C.

Therefore, we consider the data distribution in units to evaluate the joint distribution of two non-redundant attributes in 2-dimensional space. In 2-D subspaces, we also take the percentage of data contained in a unit as its density. To evaluate the relevance of two attributes, SCA measures the joint Gini value of a 2-dimensional subspace. When two attributes are highly relevant, the value is large. On the contrary, when two attributes have little relevance, the value is small because there is scarce statistical correlation in the distribution of these two attributes. The relation function is defined as follows: correlation $(A, B) = \text{Gini}(A, B)$.

After getting the relation matrix composed of relation function values of non-redundant attributes, we select values larger than the threshold $\delta = 1 - k/\omega^2$ as relative attributes. Then, we set their flags in the matrix as 1 and others as 0. Here k is still the aggregation constant of data points. In general, when computing a two-dimensional joint Gini value, we take $k = 20$.

3.4 Attribute clustering

According to the relation matrix, SCA takes attribute clustering to generate a candidate set of interesting subspaces in high dimension. Traditional clustering algorithms often deny overlapping clusters. However, in the problem of subspace clustering, attributes often belong to more than one interesting subspace. Accordingly, we need to design a special clustering method to accomplish attribute clustering. The attribute clustering process is given as follows:

AttributeClustering().

Input : relationmatrix $\boldsymbol{R}_A$ and the set

of all non-redundant attributes A

Output : the clusters in set A

(1) take the first object a_1 as the center of the first cluster, i.e., $c_1 = a_1$;

(2) For $j = 2$ *to* $|A|$ do

If $\boldsymbol{R}_{(aj,ck)} = 1$ for any existing cluster centers c_k

If $\boldsymbol{R}_{(aj,ak)} = 1$ for all objects a_k in C_k

add a_j into C_k

(3) Select an object $a' \in A$ C_1 *as* c_2

(4) Go to step (2)

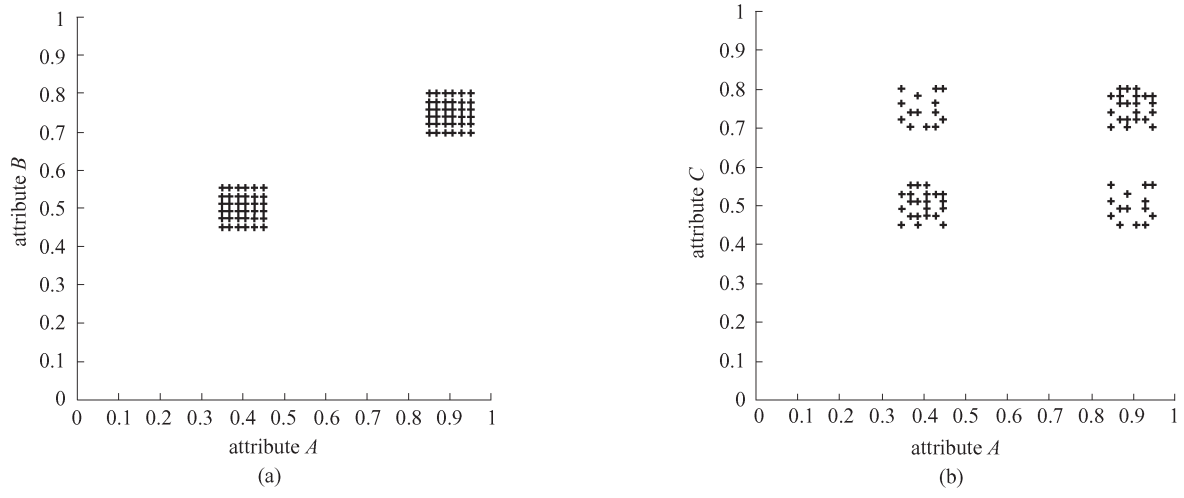

Fig. 2 Relevance of two attributes in 2-D subspace. (a) Two relevant attributes; (b) two irrelevant attributes

3.5 Generating subspace clusters

After obtaining the candidate set of interesting subspaces, the post-processing of SCA is constructed on a backtracking search. If there is no cluster in a k-D subspace S, we detect all of the $(k-1)$-D subsets of S. We should note that when detecting subspaces containing clusters, any clustering method can be taken according to different applications. In this paper, we take DBSCAN according to the experimental results in Ref. [5].

4 Experimental results

We adopt a self-implemented data generator to produce datasets with clusters of high density in specific subspaces. The data generator allows control over the structure and size of datasets through parameters such as the number of instances, the number of dimensions, and the range of values for each dimension. To discover the ability of SCA to detect subspace clusters, there are no clusters in the full feature space. All values are set between 0 to 1. In the following experiments, we take $\omega = 100$, $k = 2$ in general.

4.1 Subspace detection and accuracy

To observe the ability of detecting subspace clusters, we compare SCA, CLIQUE and ENCLUS in a 20 dimensional dataset with 100000 objects. The results are also presented as a confusion matrix that lists the relevant attributes of the input clusters as well as those outputs by the algorithm, as shown in Table 1.

Table 1 Confusion matrix of clustering results of SCA, CLIQUE and ENCLUS

clusterID	1	2	3	4
input	{1,3,5,7}	{2,3,6,8,10}	{5,10,12,13}	{9,10,11,12,13,14,15}
SCA	{1,3,5,7}	{2,3,6,8,10}	{5,10,12,13}	{9,10,11,12,13,14,15}
CLIQUE	{1,3,5,7}	{2,3,6,8,10}	{5,10,12}	{9,10,11,12}
ENCLUS	{1,3,5,7}	{2,3,6,8,10}	{5,10,12}	{9,10,11,12,13}

Table 1 shows the best case input and output clusters for SCA, CLIQUE and ENCLUS. The parameters for each method are optimized to achieve a fair comparison. SCA discovered all of the clusters while both CLIQUE and ENCLUS missed one or more dimensions in cluster 3 and cluster 4.

4.2 Scalability

We ran SCA, CLIQUE and ENCLUS with the same synthetic datasets. The purpose of these experiments was to present the scalability of the three algorithms. The experiments were run with $\omega = 100$.

Figure 3(a) shows the scalability as the size of the database is increased from 100000 to 1000000 instances. The data space had 20 attributes and there were 10 clusters, each in a different 5-dimensional subspace. As expected, the running time of these three algorithms scale linearly with the size of the database because of the rather simple but efficient grid-based clustering model. Moreover, SCA takes the method of attribute clustering instead of an a priori-like search, which makes it faster.

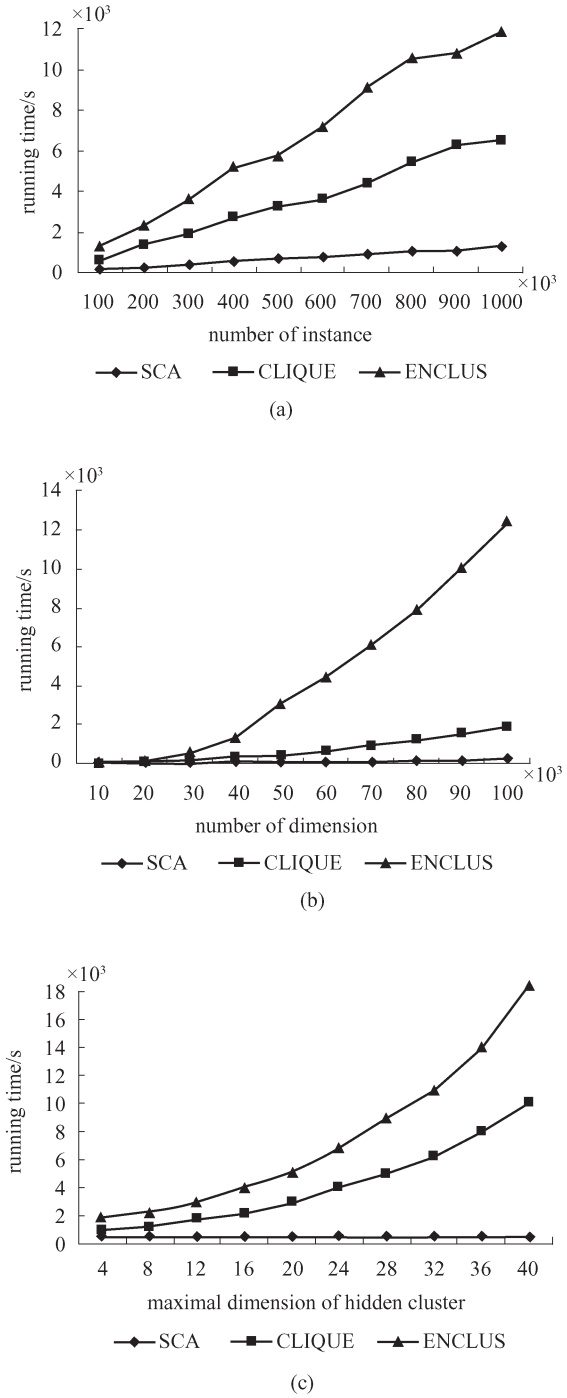

Fig. 3 (a) Running time vs. number of instances; (b) running time vs. number of instances dimensions; (c) running time vs. the dimensionality of the hidden cluster

Figure 3(b) shows the scalability as the dimensionality of the data space is increased from 10000 to 100000. The database had 100000 instances and there were 10 clusters, each in a different 5-dimensional subspace. As can be seen from Fig. 3(b), the running time of both CLIQUE and ENCLUS clearly increases non-linearly, while the curve of SCA exhibits just quadratic behavior. Our experiments show that SCA is the only method that can be efficiently applied to the datasets with more than 100000 dimensions.

Figure 3(c) shows the impact of the highest dimensionality of a subspace cluster on running time. We observe again that SCA clearly outperforms CLIQUE and ENCLUS. SCA abandons a traditional search mode and takes attribute clustering, which results in its independence of the highest dimensionality of subspace clusters.

4.3 Impact of parameterization on accuracy

We evaluated the impact of ω on 20-dimensional datasets with 100000 instances. The results show that the curve exhibits quadratic behavior, since ω decides the complexity to compute the attribute relevance. At the same time, all clusters are still found with high precision during the experiments. Moreover, since a larger value of k only means more redundant attributes, running time is almost the same with different values of k. It should be noted that k has no influence on the clustering accuracy.

5 Conclusions

In this paper, we discuss the problem of automatic subspace clustering. The solution we proposed, SCA, has been designed to find clusters embedded in subspaces of high dimensional datasets. SCA employs a relation function to evaluate the relevance of every two attributes. It is insensitive to the order of input records and does not presume some canonical data distribution. A thorough experimental evaluation has shown that the effectiveness of SCA is significantly better than that of well-known algorithms such as CLIQUE and ENCLUS. In addition, SCA clearly outperforms CLIQUE and ENCLUS in terms of scalability and running time with respect to data dimensionality and subspace dimensionality. An approach for future work is the development of an efficient subspace clusters generating method.

Acknowledgements This work was supported by the National Basic Research Program of China (No. 2007CB307100) and the National Natural Science Foundation of China (Grant No. 60432010).

References

1. Agrawal R, Gehrke J, Gunopulos D, et al. Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of ACM SIGMOD International Conference on Management of Data. Washington: ACM Press, 1998: 94–105
2. Agrawal R, Gehrke J, Gunopulos D, et al. Automatic subspace clustering of high dimensional data. Data Mining and Knowledge Discovery, 2005, 11(1): 5–33
3. Cheng C H, Fu A W, Zhang Y. Entropy-based subspace clustering for mining numerical data. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. USA: ACM Press, 1999: 84–93
4. Goil S, Nagesh H S, Choudhary A. MAFIA: efficient and scalable subspace clustering for very large data sets. Technique Report No. CPDC-TR-9906-010. Center for Parallel and Distributed Computing, Dept. of Electrical and Computer Engineering, Northwestern University: Evanston, IL, 1999
5. Procopiuc C M, Johes M, Agarwal P K, et al. A Monte Carlo algorithm for fast projective clustering. In: Proceedings of ACM SIGMOD International Conference on Management of Data. Madison: ACM Press, 2002: 418–427
6. Huang Z, Ng M, Rong H. Automated variable weighting in k-means type clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(5): 657–668
7. Kriegel H, Kröger P, Renz M, et al. A generic framework for efficient subspace clustering of high-dimensional data. In: Proceedings of 5th IEEE International Conference on Data Mining. New Orleans: IEEE Press, 2005: 250–257