Soft Comput (2011) 15:845–853
DOI 10.1007/s00500-010-0552-8

FOCUS

Efficient approaches for summarizing subspace clusters into *k* representatives

Guanhua Chen · Xiuli Ma · Dongqing Yang · Shiwei Tang · Meng Shuai · Kunqing Xie

Published online: 9 March 2010

Abstract A major challenge in subspace clustering is that subspace clustering may generate an explosive number of clusters with high computational complexity, which severely restricts the usage of subspace clustering. The problem gets even worse with the increase of the data's dimensionality. In this paper, we propose to summarize the set of subspace clusters into *k* representative clusters to alleviate the problem. Typically, subspace clusters can be clustered further into *k* groups, and the set of representative clusters can be selected from each group. In such a way, only the most representative subspace clusters will be returned to user. Unfortunately, when the size of the set of representative clusters is specified, the problem of finding the optimal set is NP-hard. To solve this problem efficiently, we present two approximate methods: PCoC and HCoC. The greatest advantage of our methods is that we only need a subset of subspace clusters as the input instead of the complete set of subspace clusters. Precisely, only the clusters in low-dimensional subspaces are computed and assembled into representative clusters in high-dimensional subspaces. The approximate results can be found in polynomial time. Our performance study shows both the effectiveness and efficiency of these methods.

Keywords Subspace clustering · Representative clusters · High-dimensional data · Data summarization · Data mining

G. Chen · X. Ma (✉) · D. Yang · S. Tang · M. Shuai · K. Xie
School of Electronics Engineering and Computer Science,
Peking University, Beijing 100871, China
e-mail: maxl@cis.pku.edu.cn

G. Chen
e-mail: ghchen@cis.pku.edu.cn

S. Tang
e-mail: tsw@pku.edu.cn

M. Shuai
e-mail: shuaimeng@cis.pku.edu.cn

K. Xie
e-mail: kunqing@cis.pku.edu.cn

G. Chen · X. Ma · S. Tang · M. Shuai · K. Xie
Key Laboratory of Machine Perception (Peking University),
Ministry of Education, Beijing, China

D. Yang
Key Laboratory of High Confidence Software Technologies
(Peking University), Ministry of Education, Beijing, China
e-mail: dqyang@pku.edu.cn

1 Introduction

In recent years, many real-world datasets such as web logs and gene samples consist of complex data objects, where a large number of attributes are involved. These datasets are often referred to as high-dimensional data. Clustering in high-dimensional spaces is often difficult as theoretical results questioned the meaning of closest matching in high-dimensional spaces. Subspace clustering has attracted great attention due to its capability of discovering salient patterns in high-dimensional data. Subspace clustering aims at searching for clusters in subspaces formed by any possible subset of relevant attributes. A subspace cluster can be represented as $\langle C, S\rangle$, where C is the set of objects in the cluster and S is the set of attributes that constitute the underlying subspace.

Subspace clustering has been studied extensively on scalable methods for mining various kinds of subspace clusters. However, the major challenge of subspace clustering is not only the efficiency but also the interpretability.

Most subspace clustering algorithms depend on a global density threshold, if the density threshold is high, subspace clustering may generate only commonsense clusters, while a low density threshold may generate an explosive number of results. This has severely restricted the usage of subspace clustering. As long as the number of discovered clusters is beyond tens or hundreds, it becomes difficult for an analyst to examine them directly.

This paper aims at solving the interpretability and redundancy issue by summarizing the set of subspace clusters into k representative clusters. A naive brute-force solution can be implemented by posing an enumeration process after the subspace clustering process. However, it may encounter two major challenges in many real-world applications. First, for any dataset containing potentially subspace clusters in a high-dimensional subspace (say, 20+ dimensions), the subspace clustering process is not only computationally expensive but also leaving a huge input for the enumeration process. Second, finding the set of k representative subspace clusters which represents the set of subspace clusters best is a special case of the maximum coverage problem, which is NP-hard.

There are two main contributions of this work. First, we examine the relationship between low-dimensional subspace clusters and high-dimensional subspace clusters which makes it possible that the representative subspace clusters can be assembled from low-dimensional subspace clusters. Second, we develop two approaches for discovering the set of k representative subspace clusters: PCoC (partition-based clustering on subspace clusters) and HCoC (hierarchical clustering on subspace clusters). PCoC applies a k-medoids style clustering approach with fast grouping so that it can be used to deal with a large number of subspace clusters, while HCoC can return a dendrogram which allows a user to explore the cluster structure from high level to detailed level. Even in cases that the complete set of subspace clusters is computationally infeasible, it is also possible to have a global view of subspace clusters through the set of k representative subspace clusters.

The rest of the paper is organized as follows. We first introduce related work in Sect. 2. In Sect. 3, the problem of mining k representative subspace clusters is defined and related concepts are introduced. In Sect. 4, we introduce our methods, PCoC and HCoC, step by step. Section 5 reports the performance evaluation of our methods both on real and synthetic datasets. We summarize our work and discuss some future research directions in Sect. 6.

2 Related work

Subspace clustering has been studied extensively on scalable methods for mining various kinds of subspace clusters, such as grid-based (Agrawal et al. 1998), entropy-based (Cheng et al. 1999), and density-based (Baumgartner et al. 2004; Bohm et al. 2004; Kriegel et al. 2005; Assent et al. 2007) approaches. The above approaches may suffer from the problem of a sheer size of redundancy results.

The introduction of the skyline of subspace clusters (Chen et al. 2008) and clusters in maximal dimensional subspace (Kriegel et al. 2005) can partially alleviate this redundancy problem. The skyline of subspace clusters is the complete set of dominating subspace clusters which cannot be covered by any other cluster. A subspace cluster is a cluster in maximal dimensional subspace if and only if its object set does not clustering in any super-space that contains its attribute set. Unfortunately, for any subspace cluster $\langle C, S\rangle$, as long as there is a small disturbance on the object set, it may generate hundreds of sub-clusters in different subspaces. In fact, these sub-clusters overlap greatly on their object sets, thus a representative subspace cluster can be a more general representation of these clusters which may represent users' interest more accurately.

Another related research work is the problem of summarization of frequent itemsets. To achieve a concise summary or representatives of frequent itemsets, recent works such as the closed pattern (Pasquier et al. 1999), the spanning set approach (Afrati et al. 2004), the clustering-based approach (Xin et al. 2005), the profile-based approach (Yan et al. 2005), the Markov random field approach (Wang and Parthasarathy 2006) and the regression-based approach (Jin et al. 2008) are developed. We share some ideas from these works; however, there are two main differences between our work and them. First, the compact set of frequent itemsets only involves the set of items and its support, while the representative subspace clusters should consider both the object set and attribute set with its density. Second, the methods developed in this paper are built on the set of low-dimensional subspace clusters, while the work on the representatives of frequent itemsets need the whole set of frequent itemsets being computed at first.

3 Problem statement

Given a set of n objects with d attributes, Let $D = \{o_1, o_2,\ldots, o_n\}$ be the set of objects and $A = \{a_1, a_2,\ldots, a_d\}$ be the set of attributes. An object $o_i = (v_{i1}, v_{i2},\ldots, v_{id})$ $(1 \le i \le n)$, v_{ik} $(1 \le i \le n,\ 1 \le k \le d)$ is the value of o_i on attribute a_k, which can also be denoted by dot notation $o_i{\cdot}a_k$.

Attributes can have different meaning, ranges and even data types, thus there can be different (dis)similarity measures on each attribute. Without restricting the generality, we assume that there are d similarity check functions $f_1, f_2, \ldots, f_d$ defined on each attribute, $f_k(o, o')$ returns true if o and o' is similar on the attribute a_k, otherwise false. Further, we assume the similarity check functions have the following properties:

1. $f_k(o, o)$ is true.
2. $f_k(o, o') = f_k(o', o)$ (referred to as symmetry property).
3. If both $f_k(o, o')$ and $f_k(o', o'')$ return true, then $f_k(o, o'')$ returns true (referred to as transition property).

Now we define the subspace cluster based on the set of similarity check functions, which can be regarded as the clustering structure on a single attribute.

Definition 1 (*Subspace cluster*) A subspace cluster $\langle C, S\rangle$ is a set of objects C with a set of attributes S that satisfies all of the following conditions:

1. $C \subseteq D$, $S \subseteq A$, $|C| > 0$, $|S| > 0$.
2. For any two objects $o_i, o_j \in C$, for any attribute $a_k \in S$, $f_k(o_i, o_j)$ returns true.
3. $|C| \geq \text{coverage}\cdot n$, where coverage ($0 < \text{coverage} < 1$) is a threshold specified by user to guarantee the number of objects exceeding a portion of the whole set of objects, in other words, to make sure that the object set is not too small.

Condition 2 is referred to as Similarity Condition by which we can define different similarity measures for each attribute, and a subspace cluster can be interpreted easily as a set of objects that are similar on every attribute of its attribute set. Condition 3 is denoted as Density Condition which is also used in CLIQUE (Agrawal et al. 1998) for identifying dense units. Conceptually, this definition is similar to those in grid-based clustering and density-based clustering. The main difference is that we treat each attribute as a separate distribution while other approaches put all attributes together and create a multinomial model.

Theorem 1 (Subspace clusters have the downward closure property on the attribute set) *Theorem 1 ensures that any subspace cluster is still a subspace cluster on the subset of its attribute set. Formally, for a subspace cluster $\langle C, S\rangle$, $\langle C, S'\rangle$ is still a subspace cluster for any $S' \subseteq S$ and $|S'| > 0$. Proving theorem 1 is intuitive but this downward closure property is useful in the A priori style subspace search as a pruning criterion. We refer a subspace cluster $\langle C, S\rangle$ to as a subspace cluster with a maximal attribute set if there does not exist an attribute set S', such that $S \subset S'$ and $\langle C, S'\rangle$ is a subspace cluster.*

Theorem 2 (Subspace clusters have the downward closure property on the object set) *Similar to Theorem 1, for a subspace cluster $\langle C, S\rangle$, $\langle C', S\rangle$ is still a subspace cluster for any $C' \subseteq C$* if *$|C'| > coverage\cdot n$.*

Usually, we are interested in subspace clusters with maximal object set. Formally, for a subspace cluster $\langle C, S\rangle$, C is a maximal object set if and only if there does not exist an object set C', such that $C \subset C'$ and $\langle C', S\rangle$ is a subspace cluster. This is referred to as Maximal Condition. From now on, if not specified explicitly, we refer a subspace cluster to as a subspace cluster with maximal object set.

Theorem 3 *Given a set of subspace clusters $G = \{\langle C_i, S_i\rangle\}$ ($1 \leq i \leq m$), construct an object set C and an attribute set S by $C = \bigcap_{i=1}^{m} C_i$ and $S = \bigcup_{i=1}^{m} S_i$, then $\langle C, S\rangle$ is a subspace cluster if $|C| \geq coverage\cdot n$.*

Proof For $|C| \geq coverage\cdot n$, the density condition is already satisfied. To prove $\langle C, S\rangle$ is a subspace cluster with maximal object set, we need to prove $\langle C, S\rangle$ satisfies both the similarity condition and the maximal condition.

First, we prove $\langle C, S\rangle$ satisfies the similarity condition. For any two objects $o_i, o_j \in C$, for any attribute $a_k \in S$, there must exist a subspace cluster $\langle C', S'\rangle \in G$, such that $a_k \in S'$ because $S = \bigcup_{i=1}^{m} S_i$, which means each attribute in S belongs to at least one attribute set in $\{S_i\}$. From $C = \bigcap_{i=1}^{m} C_i$, we have $o_i, o_j \in C''$ because $C \subseteq C''$. Then we have $f_k(o_i, o_j)$ is true because $\langle C'', S''\rangle$ itself is a subspace cluster and satisfies the similarity condition on attribute a_k.

Second, we prove $\langle C, S\rangle$ satisfies the maximal condition. Suppose there exists a subspace cluster $\langle C'', S\rangle$ such that $C \subset C'$, then there is an object o that $o \in C'$ but $o \notin C$ (i.e. $o \in C'/C$). As an object in C', o is similar to objects in C over every attribute in S. From $o \notin C$, we have there exists at least one subspace cluster $\langle C'', S''\rangle \in G$ such that $o \notin C''$. We separate C'' into two set of objects, one is C and the other is C''/C. For $S'' \subseteq S$, o is similar to objects in C over every attribute in S''. If C''/C is not empty, objects in C''/C are similar to objects in C over every attribute in S''. Due to the transition property of similarity check functions, we have that o is also similar to objects in C''/C over every attribute in S''. With above conclusions, we have o is similar to all objects in C'' over every attribute in S'', then $\langle C''\cup\{o\}, S''\rangle$ is a subspace cluster, which conflicts with $\langle C'', S''\rangle$ is a subspace cluster with maximal object set. From the conflict, we conclude that $\langle C, S\rangle$ is a subspace cluster with the maximal object set. □

According to Theorem 3, we have following implications:

1. The subspace clusters in high-dimensional subspaces can be constructed from low-dimensional subspace clusters directly. Especially, a subspace cluster with k attributes can be constructed easily by k subspace clusters with one attribute.
2. Low-dimensional subspace clusters with similar object sets can be grouped together for indicating the clustering structure over high-dimensional subspaces.

On the other hand, a significant cluster in high-dimensional subspace has more sub-clusters in lower dimensional subspace sharing similar object sets.

3. For each group of low-dimensional subspace clusters sharing similar object sets, a representative subspace cluster that covers most low-dimensional subspace clusters in the group also covers high-dimensional subspace clusters, which can be constructed from this group with high probability.

Inspired by these observations, we find that it is very important to group low-dimensional subspace clusters with similar object sets together to eliminate redundant clusters. We can merge the low-dimensional subspace clusters in each group to form the representative subspace clusters. The representative subspace cluster is composed by the maximal set of most sharing objects clustering over a maximal set of attributes within the group. Formally, given a group of subspace clusters $G = \{\langle C_i, S_i\rangle\}$, the representative subspace clusters can be defined as follows:

Definition 2 (*Representative subspace cluster*) Subspace cluster $\langle C, S\rangle$ is a representative subspace cluster for G satisfying following conditions:

1. $\langle C, S\rangle$ is a subspace cluster can be constructed (according to Theorem 3) from a subset of subspace clusters in G
2. For all subspace clusters that can be constructed in G, S has the largest attribute set.

We use *derive*(G) to denote the complete set of subspace clusters that can be generated from any subset of subspace clusters in G. For two subspace clusters $\langle C, S\rangle$, $\langle C', S'\rangle \in derive(G)$, if $C \subseteq C'$ and $S' \subseteq S$, we say $\langle C, S\rangle$ represents $\langle C', S'\rangle$. We use $rep(\langle C, S\rangle)$ to denote the complete set of subspace clusters that can be represented by $\langle C, S\rangle$.

Theorem 4 *For all subspace clusters in derive(G), the representative subspace cluster $\langle C, S\rangle$ of G has the largest set of rep($\langle C, S\rangle$).*

Proof For the downward closure property on the attribute set, we have $rep(\langle C, S\rangle) = P(\langle C, S\rangle)$, where $P(\langle C, S\rangle)$ is the subspace clusters containing C over every attribute set S' in the power set of S. Thus, we have $|rep(\langle C, S\rangle)| = |P(\langle C, S\rangle)| = 2^{|S|} - 1$. As long as the representative subspace cluster has the largest attribute set, that is, has the maximal $|S|$, we conclude that it has the largest set of $rep(\langle C, S\rangle)$. □

Based on above discussion, we define our problem of mining representative subspace clusters as finding k representative subspace clusters that represent as many as possible subspace clusters in the whole set of subspace clusters, that is, finding a set of k subspace clusters $\{\langle C_1, S_1\rangle, \langle C_2, S_2\rangle, \ldots, \langle C_k, S_k\rangle\}$ that maximizes $\left|\bigcup_{i=1}^{k} rep(\langle C_i, S_i\rangle)\right|$.

Theorem 5 *Given a set of subspace clusters $G = \{\langle C, S\rangle\}$, the problem of finding k representative subspace clusters from G is NP-hard.*

Proof We regard G as a set of ground elements, that is, each subspace cluster $\langle C, S\rangle$ in G is an ground element. For $rep(\langle C, S\rangle) \subseteq G$, we denote $R = \{rep\langle C, S\rangle\}$. Then we have $|G|$ ground elements (subspace clusters) and $|R|$ sets of ground elements. The problem of finding k subspace clusters in G such that maximizes $\left|\bigcup_{i=1}^{k} rep(\langle C_i, S_i\rangle)\right|$ is equivalent to finding k sets in R that maximizes $\left|\bigcup_{i=1}^{k} rep(\langle C_i, S_i\rangle)\right|$, which can be reduced to the unweighted maximal coverage problem; therefore, the problem is NP-hard. □

4 Efficient mining of the representative subspace clusters

In this section, we study efficient mining of the representative subspace clusters. In Sect. 4.1 we first present an overview of our methods.

4.1 General idea

According to Theorem 5, finding the set of k representative subspace clusters that represent maximal number of subspace clusters in the whole set of subspace clusters is equivalent to the unweighted maximal coverage problem. However, we can use a simple heuristics to achieve an approximate result in polynomial time. First, we group the set of low-dimensional subspace clusters into k disjoint groups based on their similarity. In this step, we implemented two clustering-based approaches to maximize inner similarity and minimize inter-similarity among the groups: a k-medoids style clustering method can achieve a local optimization on similarity within each group, while a hierarchical agglomerative clustering method has the global optimization results. After that, a representative subspace cluster is assembled from each group by a greedy algorithm.

Based on Theorem 3, we can generate the representative subspace clusters from a set of low-dimensional subspace clusters. Typically, we only compute subspace clusters with the set of attributes no more than *small_d* attributes. For *small_d* is a small number (say, 2 or 3), the whole running time of mining representative subspace clusters is polynomial. The whole process consists of the following steps:

4.1.1 Mining of low-dimensional subspace clusters

First, all clusters on each single attribute are computed. Then, all of low-dimensional subspace clusters having no more than *small_d* attributes can be generated according to Theorem 3.

4.1.2 Partition subspace clusters

The discovered low-dimensional clusters are partitioned into k disjoint groups based on their similarity. To achieve this, we develop two approaches: PCoC and HCoC. PCoC is a k-medoids style approach that achieves local optimal result with fast convergence, while HCoC is a hierarchical agglomerative clustering approach having global optimal result. In addition, HCoC can return a dendrogram which allows a user to explore the cluster space in a top-down manner.

4.1.3 Generating representative subspace clusters

After the grouping process, a representative subspace cluster is generated from each group by a greedy algorithm.

4.2 Mining of low-dimensional subspace clusters

For each attribute, we discover its clustering structure based on the definition that a cluster (dense region) is a set of objects spanning over a region having significant higher density than expectation. By assuming a uniform distribution, we define the expected density of attribute a_i as $density_i = Dom(a_i)/n$, where $Dom(a_i)$ is the range of values on a_i, i.e. $Dom(a_i) = \max\{v_{ji}\} - \min\{v_{ji}\}$ $(1 \le j \le n)$. Given a region of length l on a_i, the expected number of objects in the region is $l/density_i$. A dense object on a_i is defined as having more objects in its neighborhood. In our method, we define the region $(v_{ji} - density_i, v_{ji} + density_i)$ as the neighborhood of o_j and the expected number of objects in its neighborhood is 2. o_j is a dense object if there are no less than $2x$ objects in its neighborhood, where x is a dense threshold and $x > 1$. In our experiments, $x = 2$ (twice to the expected density) is sufficient for tight clusters.

As long as the dense object is defined, we develop a variant of DBSCAN (Ester et al. 1996) to find all density connected clusters having no less than $coverage{\cdot}n$ objects over each attribute. For attribute a_i, the set of clusters found on a_i is denoted as $G_{ai} = \{\langle C_j, \{a_i\}\rangle\}$. The similarity check function f_i can be defined on G_{ai} as follows: if o, o' belong to the same cluster in G_{ai}, $f_i(o, o')$ returns true, otherwise false. The similarity check functions defined in this way have the symmetry property and transition property, thus they are valid similarity check functions.

After clusters on every single attribute are found, we generate all of subspace clusters with no more than *small_d* attributes in the A priori style search based on the downward closure property on the attribute set. The cost of this step is $O(m^{small\_d})$ in the worst case, where m is the number of all clusters on every single attributes. Obviously, m is linear with d.

With the complete set of low-dimensional subspace clusters (denoted as G), we first present a partition-based and hierarchical clustering approaches to separate G into k disjoint groups, then generate the representative subspace clusters from these groups.

4.3 PCoC

First we define the measures used in the clustering process, then the algorithm is presented step by step, with possible optimizations.

4.3.1 Definitions

First, we define the similarity between two subspace clusters as follows:

$$Sim(\langle C, S\rangle, \langle C', S'\rangle) = |C \cap C'|/|C \cup C'|.$$

In this similarity measure, only the similarity between the object sets is considered. We do not need to consider the similarity between their attribute sets, because if two subspace clusters share no attribute with each other but have similar object set, they can be merged into a subspace cluster with union of their attribute set according to Theorem 3, thus they can be represented by the same subspace cluster and should be grouped together. On the other hand, if there are two subspace clusters with identical attribute set (i.e. in the same subspace), they could have no sharing object (otherwise the Maximal Condition is violated), and they should not be grouped together.

Second, we define the measure of cluster quality, which is used as the stop condition in the iteration process.

We define quality measure $Q = \sum_{i=1}^{k} (1/|G_i| \sum_{\langle C,S\rangle \in G_i} Sim\langle C, S\rangle, \langle C_i, S_i\rangle))$, where G_i is the i-th group of subspace clusters and $\langle C_i, S_i\rangle$ is the center of the i-th group. The quality measure Q is the sum of average similarity to the center in each group. The larger Q, the better grouping quality is.

4.3.2 The algorithm

As shown in Algorithm 1, we present PCoC step by step. The algorithm is a variant of k-medoids grouping similar subspace clusters. First, k subspace clusters are selected randomly as centers of k groups. Then each of the remaining subspace clusters is assigned to be a member of the group having the largest similarity with the center. After the assignment, a center can be replaced by any random selected subspace cluster if the new center set with their members can achieve better quality Q. The algorithm stops when tried sufficient times of replacing centers without having a better quality. Instead of the typical k-means clustering algorithm, we choose k-medoids algorithm for following reasons:

1. k-Medoids is more robust to outliers.
2. k-Medoids usually gets a faster convergence than k-means.
3. The center of a group can be a subspace cluster directly.
4. The final k center subspace clusters found by k-medoids can be regard as core clusters and used in the following greedy search of representative subspace clusters.

Algorithm 1 Grouping subspace clusters: PCoC

Input: set of subspace clusters $G = \{\langle C, S\rangle\}$, number of groups K

Output: K set of subspace clusters G_1, G_2,..., G_K and their centers

```
1: randomly select $K$ subspace clusters as the initial center of $K$
   groups;
2: for each subspace cluster <$C$, $S$>$\in G$ do
     assigns its membership to the group that has the largest
     similarity with the center;
3: calculate the cluster quality $Q$
4: randomly select subspace clusters to replace one of the group
   centers and form a new list of centers and new group structure
   with cluster quality $Q'$
5: if ($Q' > Q$), then $Q = Q'$, repeat 2
6: if not tried sufficient times, repeat 4
7: else return the $K$ groups with their centers;
```

4.3.3 *Optimizations*

There are heuristics can be used to improve the performance of Algorithm 1. As discussed in Sect. 4.3.1, two subspace clusters with identical attribute set have no sharing object, thus should not be grouped together. Using this observation, we can select those subspace clusters with identical attribute set as the initial group centers. This optimization provides an acceleration of the convergence in the process of iteration.

Another optimization is that we can calculate a similarity matrix composing similarity between any two subspace clusters at the start of the algorithm so that the similarity to the centers need not to be re-calculated in each iteration. This optimization avoids many repeating similarity computation between subspace clusters.

4.4 HCoC

As shown in Algorithm 2, we present HCoC step by step. First, each of subspace cluster forms a group containing itself. Then, two groups with largest similarity are merged until there are no more than k groups. The similarity between groups is formulated by the average similarity between all pairs of their subspace clusters. Formally, for two groups of subspace clusters G_1 and G_2, we define

$$sim(G_1, G_2) = 1/(|G_1| \cdot |G_2|) \sum_{\langle C,S\rangle \in G_1, \langle C',S'\rangle \in G_2} sim(\langle C, S\rangle, \langle C', S'\rangle).$$

Hierarchical clustering produces a dendrogram where two groups are merged together at each level. The dendrogram allows a user to explore the cluster space in a top-down manner and provides a global view of subspace clusters. With the dendrogram, user can be released from finding the optimal k representative subspace clusters.

Algorithm 2 Grouping subspace clusters: HCoC

Input: set of subspace clusters $G = \{\langle C, S\rangle\}$, number of groups K

Output: K set of subspace clusters G_1, G_2,..., G_K and their centers

```
1: initialize $k = |G|$ groups, each of which has one subspace
   cluster, it is also the center of the group;
2: calculate $sim(G_i, G_j)$ for all pairs of groups $G_i$, $G_j$
3: while ($k > K$)
4:   select $G_i$, $G_j$ having largest $sim(G_i, G_j)$
5:   merge $G_i$, $G_j$ to a new group $G_{new}$;
6:   calculate the similarity between $G_{new}$ and the remaining groups;
7:   $k = k - 1$;
8: end while
9: select the center cluster for each group such that the sum of
   similarity between the center and the other clusters in the
   group is maximized.
10: return the $K$ groups with their centers;
```

4.5 Generating representative subspace cluster

For each group of subspace cluster, a representative subspace cluster can be generated by merging a subset of subspace clusters. Intuitively, a brute-force method can try all combinations of subspace clusters to find the one has largest attribute set; however, this method is infeasible for computation. We propose a greedy search method that can find an approximate result in polynomial time. As shown in Algorithm 3, we start the search from the center of the group (step 1), each time we add in a subspace cluster which has the largest similarity with the representative subspace cluster found so far, until there does not exist any subspace cluster that can be added (step 2–6).

Algorithm 3 Find a representative subspace cluster

Input: A group of subspace clusters $G = \{\langle C, S\rangle\}$, the center of G: $\langle C_i, S_i\rangle$

Output: a representative subspace cluster $\langle C_r, S_r\rangle$

```
1: $C_r = C_i,\ S_r = S_i,\ G=G-<C_i,S_i>;$
2: while $(|G|>0)$
3:   find $<C, S>$ in $G$ that has largest similarity with $<C_r, S_r>$
4:   if $<C\cap C_r, S\cup S_r>$ is a valid subspace cluster
          $C_r=C\cap C_r,\ S_r= S\cup S_r$
          $G=G-<C, S>;$
5:   else 7
6: end while
7: return $<C_r, S_r>$
```

5 Experiment study

We evaluate the effectiveness and efficiency of our methods on both synthetic and real dataset. All the experiments are performed on a 2 GHz Pentium 4 PC with 1 GB main memory, running on Windows XP. All the programs are written in Java. In the experiments, we set *small_d* = 2, that is, only the set of subspace clusters with no more than 2 attributes are computed.

5.1 Synthetic datasets

We implement a synthetic data generator to produce datasets with clusters of high density in specific subspaces. The data generator allows controlling over the structure and the size of datasets through parameters such as the number of records, the number of dimensions, and the range of values for each dimension. Initially, each dimension of the dataset is generated in a random style (equivalent to a uniform distribution on every dimension). Then, different subspace clusters are inserted into the dataset, replacing parts of the original random values. We can design the subspace clusters by their dimensionality and size.

5.2 Real datasets

A real sales dataset is organized into a data cube with dimensions such as products, locations, time and the amount of product sales. The fact table of the data cube is composed of the daily sales of over a hundred of products for 3 years. The clustering process is equivalent to answer the questions like *Find the similar product groups according to their weekly sales during last n weeks*, with *n* varying from 20 to 60. It is a dense dataset with many subspace clusters embedded in subspaces with 3–20 attributes. We use this dataset to test the efficiency of our algorithms. For the sake of privacy, details of the products and sales data are neglected.

5.3 Effectiveness

First, we examine the effectiveness of our method on the synthetic data. We designed 6 datasets with 200 objects and 100 attributes. Each dataset may have 3–8 groups of subspace clusters. Subspace clusters in each group are sharing similar object set. The typical subspace cluster contains 20% objects and is embedded in the subspace with 10–30 attributes. Subspace clusters may overlap with each other both on object set and attribute set. We apply PCoC and HCoC on these datasets, respectively. By varying *k*, we count the valid representative subspace clusters found by PCoC and HCoC. We report our result in Tables 1 and 2.

As shown in Tables 1 and 2, we can see that as long as the optimal *k* is selected, the valid representative subspace clusters can provide a consistent summary of the groups of subspace clusters. When *k* is less than the actual number of groups of subspace clusters, smaller groups are missed in the result. When *k* is more than the actual number of groups of subspace clusters, bigger groups are divided into smaller groups. Although the perfect value of *k* is hard to guess beforehand, the value of *k* other than the optimal one also returns informative results through which a user may have a global view of data distribution.

Table 1 Number of valid representative subspace clusters discovered by PCoC

Dataset no.	Number of groups	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$
1	3	3	4	4	4	4	4	5
2	4	3	4	5	6	5	6	4
3	5	3	4	5	6	6	6	6
4	6	3	4	5	6	6	8	7
5	7	3	3	4	6	7	6	8
6	8	2	2	3	6	7	8	8

Table 2 Number of valid representative subspace clusters discovered by HCoC

Dataset no.	Number of groups	$K = 3$	$K = 4$	$K = 5$	$K = 6$	$K = 7$	$K = 8$	$K = 9$
1	3	3	4	5	6	7	8	8
2	4	3	4	4	5	6	7	7
3	5	3	4	5	6	6	7	7
4	6	3	4	5	6	7	8	9
5	7	3	3	4	6	7	6	8
6	8	3	4	4	6	7	8	8

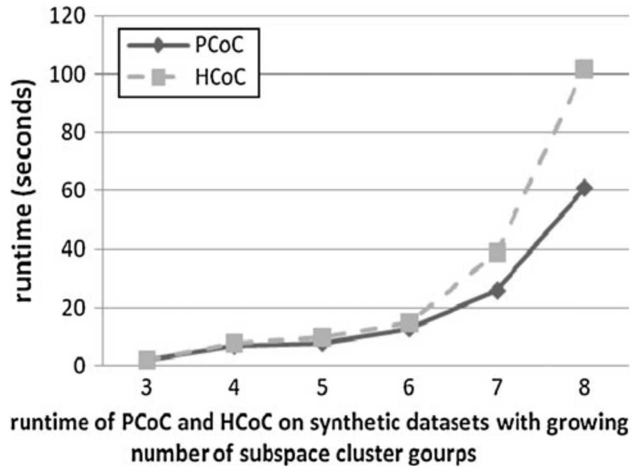

Fig. 1 Runtime of PCoC and HCoC on synthetic datasets with growing number of groups of subspace cluster

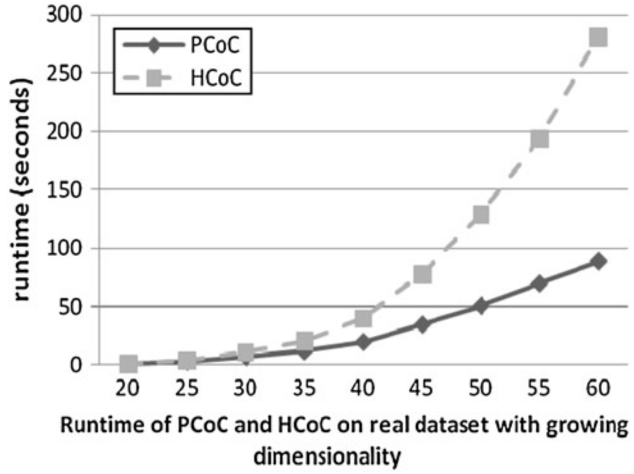

Fig. 2 Runtime of PCoC and HCoC on real dataset with growing dimensionality

5.4 Efficiency

In Fig. 1, we test PCoC and HCoC on different synthetic datasets with increasing number of groups of subspace clusters. More groups of subspace clusters means more subspace clusters are needed to be processed. The runtime of PCoC and HCoC grows in polynomial with the number of groups of subspace clusters.

In Fig. 2, we test PCoC and HCoC on the real dataset with increasing number of attributes by querying weekly sales data from 20 to 60 weeks of over a hundred of products. Unlike the typical subspace clustering algorithms that have exponential cost on dimensionality of data, HCoC and PCoC both finished in acceptable time and grows polynomial with the dimensionality. For each case, we tried different k from 3 to 10. Basically, the value of k has little influence on the runtime and the maximal runtime of each case is selected.

From the result of above experiment, we can find out that PCoC has better performance than HCoC both on the synthetic datasets and the real dataset because the major computation cost in HCoC is the pair-wise similarity calculation. In total, Algorithm 2 has $O(|G|^2)$ similarity computation. PCoC is an approach to eliminate the quadratic cost, which can achieve very fast clustering for a large number of subspace clusters.

6 Conclusion

In this paper, we have studied how to summarize subspace clusters into k representatives efficiently. The representative subspace clusters can solve the interpretability issue caused by the huge number of subspace clusters in high-dimensional dataset. We examined the relationship between low-dimensional subspace clusters and high-dimensional subspace clusters so that we can assemble the representatives from the set of low-dimensional subspace clusters. Without generating the complete set of subspace clusters, the problem can be approximately solved in polynomial time. Empirical studies indicate that we can obtain very compact set of representatives with high cover ratio in real datasets. Other than the post-processing style approaches, our work can be directly applied on the input dataset as a mining process which achieves much better effectiveness and efficiency. To improve the effectiveness of our wok, finding the optimal k automatically that is as small as possible without degrading the set of representative subspace clusters is an open issue.

Acknowledgments This work was supported by the National High-Tech Research and Development Plan of China (863) under Grant No. 2007AA120502; NSFC 60874082.

References

Afrati F, Gionis A, Mannila H (2004) Approximating a collection of frequent sets. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining (KDD'04), pp 12–19

Agrawal R, Gehrke J, Gunopulos D, Raghavan P (1998) Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of international conference on management of data (SIGMOD'98), pp 94–105

Assent I, Krieger R, Müller E, Seidl T (2007) DUSC: dimensionality unbiased subspace clustering. In: Proceedings of the seventh IEEE international conference on data mining (ICDM'07), pp 409–414

Baumgartner C, Kailing K, Kriegel H-P, Kroger P, Plant C (2004) Subspace selection for clustering high dimensional data. In: Proceedings of the fourth IEEE international conference on data mining (ICDM'04), pp 11–18

Bohm C, Kailing K, Kriegel H-P, Kroger P (2004) Density connected clustering with local subspace preferences. In: Proceedings of the fourth IEEE international conference on data mining (ICDM'04), pp 27–34

Chen G, Ma X, Yang D, Tang S (2008) Discovering the skyline of subspace clusters in high-dimensional data, In: Proceedings of the fifth international conference on fuzzy systems and knowledge discovery (FSKD '08), vol 2, pp 439–443

Cheng CH, Fu AC, 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), pp 84–93

Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the second ACM SIGKDD international conference on knowledge discovery and data mining (KDD'96), pp 226–231

Jin R, Abu-Ata M, Xiang Y, Ruan N (2008) Effective and efficient itemset pattern summarization: regression-based approaches, In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'08), pp 399–407

Kriegel HP, Kroger P, Renz M, Wurst S (2005) A generic framework for efficient subspace clustering of high-dimensional data. In: Proceedings of the fifth IEEE international conference on data mining (ICDM'05)

Pasquier N, Bastide Y, Taouil R, Lakhal L (1999) Discovering frequent closed itemsets for association rules. In: Lecture notes in computer science: database theory, ICDT, pp 398–416

Wang C, Parthasarathy S (2006) Summarizing itemset patterns using probabilistic models. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'06), pp 730–735

Xin D, Han J, Yan X, Cheng H (2005) Mining compressed frequent-pattern sets. In: Proceedings of the 31st international conference on very large data bases (VLDB'05), pp 709–720

Yan X, Cheng H, Han J, Xin D (2005) Summarizing itemset patterns: a profile-based approach. In: Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery and data mining (KDD'05), pp 314–323