Data Min Knowl Disc (2013) 26:332–397
DOI 10.1007/s10618-012-0258-x

A survey on enhanced subspace clustering

Kelvin Sim · Vivekanand Gopalkrishnan · Arthur Zimek · Gao Cong

Received: 26 May 2011 / Accepted: 6 February 2012 / Published online: 24 February 2012
© The Author(s) 2012

Abstract Subspace clustering finds sets of objects that are homogeneous in subspaces of high-dimensional datasets, and has been successfully applied in many domains. In recent years, a new breed of subspace clustering algorithms, which we denote as *enhanced subspace clustering* algorithms, have been proposed to (1) handle the increasing abundance and complexity of data and to (2) improve the clustering results. In this survey, we present these enhanced approaches to subspace clustering by discussing the problems they are solving, their cluster definitions and algorithms. Besides enhanced subspace clustering, we also present the basic subspace clustering and the related works in high-dimensional clustering.

Keywords Subspace Clustering · High-Dimensional Clustering · Projected Clustering · Survey

Responsible editor: Charu Aggarwal.

K. Sim (✉)
Data Mining Department, Institute of Infocomm Research, A*STAR, Singapore, Singapore
e-mail: shsim@i2r.a-star.edu.sg

V. Gopalkrishnan
IBM Research, Singapore, Singapore
e-mail: vivek@sg.ibm.com

A. Zimek
Institute for Informatics, Ludwig-Maximilians-Universität München, Munich, Germany
e-mail: zimek@dbs.ifi.lmu.de

G. Cong
School of Computer Engineering, Nanyang Technological University, Singapore, Singapore
e-mail: gaocong@ntu.edu.sg

1 Introduction

Clustering finds groups of similar objects, and due to its usefulness, it has been successfully applied in many domains such as biology, finance, marketing, etc. Details of its applications can be found in (Jain et al. 1999; Kriegel et al. 2009). Traditional clustering accounts the full data space to partition objects based on their similarity. Recently, advances in data collection and management have led to large amount of data being collected, particularly dataset with a large number of attributes. Traditional clustering, although it is a rather mature research field, is not always appropriate to handle data sets with a larger number of attributes.

In high dimensional data, several problems occur, attributed to the so called curse of dimensionality. Interesting characteristics of this curse are discussed e.g. in (Beyer et al. 1999; Bennett et al. 1999; Francois et al. 2007; Houle et al. 2010). With respect to clustering, we can summarize the curse of dimensionality as a twofold problem (Kriegel et al. 2009)

First, several attributes may not be relevant to define a certain cluster properly and may, thus, distort the distance computations usually performed in full-dimensional space to discern similar from dissimilar points. The cluster may be present in some subset of the attributes (i.e., some subspace), but it may not be possible to identify the cluster properly in the full dimensional space.

Second, the subset of attributes relevant to some cluster (i.e., the subspace where this cluster is discernible) may be different from the subset of attributes relevant for a second cluster, both may differ from the relevant subspace for a third cluster. As a consequence, there may be no global feature reduction procedure able to identify a common subspace to derive all clusters in the data set. This second observation leads to an important property of clusters in high dimensional data. It may be meaningful to define clusters in an overlapping way, i.e., one data point can belong to cluster C_1 in a certain subspace but to cluster C_2 in another subspace. This is a possibility that is not accounted for by traditional clustering methods such as density based or partitioning approaches.

Subspace clustering has been proposed to overcome these two problems traditional clustering faced in datasets with a large number of attributes. Let $\mathbb{D} = \mathbb{O} \times \mathbb{A}$ be a dataset presented in the form of a matrix, where $\mathbb{O}$ is the set of objects and $\mathbb{A}$ is the set of attributes. A subspace cluster C is a submatrix $O \times A$, where the set of objects $O \subseteq \mathbb{O}$ is homogeneous in the set of attributes (also known as subspace) $A \subseteq \mathbb{A}$.

Research on subspace clustering has been gathering momentum for the past decade. Basic subspace clustering focuses on objects which are closely together in their subspaces, and three major variants of it have been developed, viz. the grid based, window based and density based. Although these approaches are efficient and effective in solving their clustering problems, their limitations are exposed by the recent proliferation of complex data and the need for higher quality clustering results. These fuel the research in *enhanced* subspace clustering, which is the main focus of this survey.

Enhanced subspace clustering can be broadly classified into two groups, namely *handling complex data* and *improving clustering results*.

– *Handling complex data* The basic subspace clustering approaches only handle quantitative two-dimensional (2D) data (*object* $\times$ *attribute*), and do not handle

complex data of higher order 3D data (*object* × *attribute* × *time*), infinite streaming data, noisy data or categorical data.

– *Improving clustering results* Improving clustering results derives from the shortcomings of basic subspace clustering approaches, which can be broadly categorized into three types.
First, overlapping of subspace clusters may, in certain situations, lead to an explosion of too many clusters, which is undesirable as the user will be overwhelmed by the huge numbers. Hence, instead of enumerating all subspace clusters that satisfy the definition of the cluster, *significant subspace clusters* that are intrinsically prominent in the data should be mined.
Second, the current algorithms require the user to set *tuning parameters*, and clusters that satisfy these parameters will be mined. These algorithms are generally sensitive to these tuning parameters, and given that tuning parameters are non-meaningful and non-intuitive, it is difficult for the user to set the right parameter settings. This situation is exacerbated as existing algorithms are abounded with tuning parameters, thereby complicating the clustering task. Therefore, it is desirable to overcome this *parameter-sensitivity* problem of the existing algorithms.
Third, the user may have useful information such as constraints or domain knowledge that can help to improve the quality of the clusters, but the basic subspace clustering approaches cannot incorporate these extra information. Incorporating these extra information can be viewed as a form of *semi-supervised* subspace clustering, as they are used to guide the clustering process.

In this survey, we present enhanced subspace clustering approaches that handle complex data and/or improve the clustering results, which to the best of our knowledge, is an important data mining topic that has not yet been given a comprehensive coverage.

The usual surveys give a systematic presentation of how the algorithms work. Our survey style is different from them as we decouple the subspace clustering problems and the solutions; the reader will first understand the problems and what desired properties they seek in their clustering solutions. Next, the reader will understand the solutions and their properties, and if the properties of the solutions do satisfy the desired properties that the problems seek.

1.1 Related surveys

There are a number of surveys on traditional clustering (Jain et al. 1999; Berkhin 2006; Xu and Wunsch 2005). The style of these surveys is similar, in which the clustering algorithms are organized into different categories, and explained in detail. However, the clustering algorithms are presented in different perspective in these surveys. Jain et al. (1999) present the clustering algorithms from a statistical pattern recognition perspective, and important applications of traditional clustering are also given. Berkhin (2006) presents the clustering algorithms from a data mining perspective. He also briefly touched on the topic of clustering high-dimensional data, particularly subspace clustering and co-clustering. Xu and Wunsch (2005) present the clustering algorithms

from three different perspectives, namely statistical, computer science and machine learning, and some applications of traditional clustering are also given.

There are several surveys on high-dimensional clustering (Jiang et al. 2004b; Madeira and Oliveira 2004; Parsons et al. 2004; Tanay et al. 2004; Parsons et al. 2004; Patrikainen and Meila 2006; Moise et al. 2009; Müller et al. 2009d; Kriegel et al. 2009), and we can roughly categorize them into two types: theoretical and experimental surveys.

For the theoretical surveys, they conduct in-depth analysis and discussions on the clustering algorithms. Madeira and Oliveira (2004), Jiang et al. (2004b), and Tanay et al. (2004) give surveys on pattern based clustering or biclustering algorithms, and their applications in microarray gene expression data. Kriegel et al. (2009) give a comprehensive survey in clustering high-dimensional data, which is categorized into three main family of algorithms: subspace clustering, pattern based clustering and correlation clustering.

For the experimental surveys (Parsons et al. 2004; Patrikainen and Meila 2006; Moise et al. 2009; Müller et al. 2009d), they describe subspace clustering, projected clustering and pattern based clustering algorithms, and at the same time, conduct experimentations to evaluate the scalability, efficiency and accuracy of the algorithms.

In spite of the fact that there are a considerable number of existing surveys, none of them discuss about enhanced subspace clustering.

1.2 Layout of the survey

The layout of the survey is as follows:

1. Introduction
   (a) Related Surveys
   (b) Layout of the Survey
2. Subspace Clustering: Problems
   (a) Preliminaries
   (b) Basic Subspace Clustering Problems
   (c) Enhanced Subspace Clustering Problems
      i. Handling Complex Data
      ii. Improving Clustering Results
3. Related High Dimensional Clustering Techniques
   (a) Attribute Selection and Reduction
   (b) Projected Clustering
   (c) Maximal Bicliques and Frequent Patterns
   (d) Pattern based Clustering
   (e) Correlation Clustering
   (f) Co-Clustering
   (g) Summary
4. Basic Subspace Clustering: Approaches and their Definitions
   (a) Grid based Subspace Clustering
   (b) Window based Subspace Clustering
   (c) Density based Subspace Clustering

2 Subspace clustering: problems

2.1 Preliminaries

We present some notations that will be used in the rest of this survey. We adopt the definition of tensor by Sun et al. (2007) to describe our dataset. Let our dataset be a kth order tensor $\mathbb{D} \in \mathbb{R}^{N_1 \times \ldots \times N_k}$, where N_i is the dimensionality of the ith mode, for $1 \leq i \leq k$.

In this survey, we focus on 2nd and 3rd order tensors, as they are the most common types of dataset used in clustering. For convention's sake, we denote a 2nd order tensor as a two-dimensional (2D) dataset which has dimension *objects* and *attributes*, and a 3rd order tensor as a three-dimensional (3D) dataset which has dimension *objects*, *attributes* and *timestamps*.

In the literature, the term dimension is often used interchangeably with attribute. Indeed, attributes should be a dimension of the dataset. However, to cover terminology used in the literature, we refer to a 2D dataset with a large number of attributes as high dimensional 2D dataset.

Let $\mathbb{O}$ be a set of objects and $\mathbb{A}$ be a set of attributes. We use $|.|$ to denote the cardinality of a set. The attributes can be binary, categorical, discrete or continuous. Let x_a be a value of an attribute a and the domain of a be $D(a)$. Let $max(D(a))$ and $min(D(a))$ be the maximum and minimum values in domain $D(a)$ respectively, and we denote the range of the domain of attribute a as $R_a = max(D(a)) - min(D(a))$.

We represent a 2D dataset as a matrix $\mathbb{D} = \mathbb{O} \times \mathbb{A}$, with the value of object $o \in \mathbb{O}$ on attribute $a \in \mathbb{A}$ be denoted as x_{oa}. Let $O \subseteq \mathbb{O}$ be a set of objects and $A \subseteq \mathbb{A}$ be a set of attributes. The set of attribute A is also known as the subspace of the dataset. A 2D subspace cluster $C = (O, A)$ is a submatrix $O \times A$.

We denote $D_O(a) \subseteq D(a)$ as the domain of attribute a projected on a set of objects O, i.e. $\forall x_a \in D_O(a) : \exists o \in O$ such that $x_{oa} = x_a$.

Let $ALL = \{C_1, \ldots, C_m\}$ be the set of possible subspace clusters from a dataset, and let $M = \{C_1, \ldots, C_n\} = \{(O_1, A_1), \ldots, (O_n, A_n)\} \subseteq ALL$ be a set of subspace clusters.

2.2 Basic subspace clustering problems

We introduce the basic subspace clustering problem. Subspace clustering finds clusters where sets of objects are homogeneous in sets of attributes. We can characterize subspace cluster C by the following two functions:

Definition 1 (Homogeneous function $h(C)$) The homogeneous function $h(C)$ measures the homogeneity in the matrix $O \times A$. We say that the matrix is homogeneous when $h(C)$ is satisfied, e.g. a threshold is met.

Definition 2 (Support function $\pi(C)$) The support function $\pi(C)$ measures the size of the matrix $O \times A$. We say that the size of the matrix is significant when $\pi(C)$ is satisfied, e.g. a threshold is met.

$C = (O, A)$ is a subspace cluster when (1) the set of objects O are *homogeneous* in the set of attributes A, and (2) the size of the matrix $O \times A$, is *significant*. The homogeneity of the set of objects O in the set of attributes A is measured by the homogeneous function $h(C)$, and the size of the cluster C is measured by the support function $\pi(C)$. Hence, these two functions are the basis of defining a subspace cluster.

Since different subspace clustering approaches solve different problems, they have their own homogeneous and support functions, and it is possible that some only have one of the functions. With the two functions, we can formally define a subspace cluster as:

Definition 3 (Subspace cluster $C = (O, A)$) Let O be a set of objects and A be a set of attributes. Matrix $O \times A$ forms a subspace cluster $C = (O, A)$ if

- the homogeneous function $h(C)$ is satisfied
- the support function $\pi(C)$ is satisfied
- $\forall o \in O : o$ is allowed to be in other subspace cluster C'

Note that we only discuss subspace clusters that are axis-parallel, as the submatrix $O \times A$ of a cluster $C = (O, A)$ is parallel to the axes of the dataset.

The third criterion of Definition 3 distinguishes subspace clusters from other high-dimensional data clustering approaches, such as projected clusters and co-clusters, which partition the objects into different clusters. Details about them will be discussed in Sect. 3.

If A is equivalent to the whole set of attributes $\mathbb{A}$ of the dataset, then the cluster is a traditional cluster.

Desired properties of the clusters:

- *Homogeneity* The homogeneity can be based on similarity, distance, density, etc, depending on the clustering problem. A commonly used metric is Euclidean distance. For example, a set of objects O can be said to be homogeneous, if their pairwise distances are relatively small in the Euclidean space of the subspace A (partitioning clustering model), or if they are density-connected within the Euclidean space of the subspace A (density based clustering model), or if they exhibit common trends within the subspace A (some pattern based clustering models). After deciding on the metric, the next problem is to determine the homogeneity of a cluster. The general solution is to set a threshold and the cluster is considered homogeneous if its homogeneous function exceeds the threshold.
- *Significant size* Similar to determining homogeneity, it is hard to decide the significant size of the cluster. The general solution is to set threshold on the size of the cluster. Setting threshold serves two purposes: first, the user may prefer large clusters as the significance of a cluster may be related to its size. Second, analysis of the result is easier as large clusters are usually fewer in numbers.
- *Maximal clusters* The concept of maximality is proposed by Pasquier et al. (1999) for frequent itemsets. A subspace cluster $C = (O, A)$ is maximal if there does not exist another subspace cluster $C' = (O', A')$, such that $O \subseteq O' \wedge A \subseteq A'$. A cluster that is a subset of a maximal cluster conveys the same information as the maximal cluster, hence mining maximal clusters ensures no redundancies in the results. As the use of maximality is prevalent, we assume that the clusters discussed in the rest of this survey have the maximality property, unless otherwise stated.

  A more relaxed version of maximality, known as *redundancy* is proposed in (Assent et al. 2007, 2008b). A subspace cluster (O', A') is redundant if there exists a subspace cluster (O, A) with $O' \subseteq O \wedge A \subset A' \wedge |O'| \geq r.|O|$, where r is a user-defined parameter.

Desired properties of the algorithm:

- *Complete result* It is desirable to mine the complete set of subspace clusters, so that important clusters are not omitted. However, the user should not be overwhelmed by a large number of clusters, which is why recent approaches focus more on omitting redundant clusters (Details are in Sect. 5.2.1).
- *Stable result* The same set of clusters should be mined from the same dataset in every execution of the algorithm. This is vital as the result is unusable if the result is unstable. For example, unstable results cannot be used in experimental studies to backup hypotheses. Ensemble approaches, however, sometimes try to establish a stable result based on a diverse set of (unstable) results (see Kriegel and Zimek 2010, for an overview).
- *Efficiency* It is important to have an efficient algorithm to handle high dimensional datasets. To this end, heuristics and assumptions are used in the algorithms.

2.3 Enhanced subspace clustering problems

The need for enhanced subspace clustering can be broadly categorized into two main parts, namely *handling complex data* and *improving the clustering result*. Enhanced

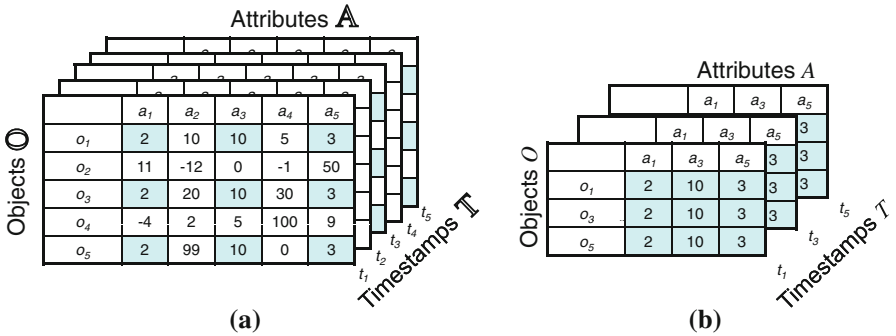


Fig. 1 **a** A 3D dataset as a cuboid $\mathbb{D} = \mathbb{O} \times \mathbb{A} \times \mathbb{T}$, **b** A 3D subspace cluster $C = (O, A, T)$ represented as a sub-cuboid $O \times A \times T$

subspace clustering is about solving problems beyond the scope of basic subspace clustering, and is not to improve existing subspace clustering algorithms (Assent et al. 2008a; Nagesh et al. 2001; Sequeira and Zaki 2004; Assent et al. 2007).

Note that the desired properties of the basic subspace clusters still hold for the enhanced subspace clusters, and each enhanced subspace cluster also has its own desired properties pertaining to the problem that it is solving.

2.3.1 Handling complex data

The basic subspace clustering problem focuses on 2D data, and not on complex data, such as 3D, categorical, stream or noisy data.

Three-Dimensional (3D) data With the advancement of data gathering over the years, more 3D datasets have been collected, such as *gene-sample-time* microarray data in biology (Madeira and Oliveira 2004), *stock-financial ratio-year* data in finance (Sim et al. 2006), *item-time-region* data in market analysis (Ji et al. 2006), etc.

Let $\mathbb{T}$ be a set of entities that corresponds to the third dimension, and the entities usually are related to time or location. In this survey, we assume that the third dimension is related to time. We represent a 3D dataset as a cuboid $\mathbb{D} = \mathbb{O} \times \mathbb{A} \times \mathbb{T}$, with the value of object $o \in \mathbb{O}$ on attribute $a \in \mathbb{A}$, at timestamp $t \in \mathbb{T}$ be denoted as x_{oat}. Figure 1a shows an example of a 3D dataset, with each cell representing a value x_{oat}.

2D subspace clusters can be mined from each time frame of the 3D dataset, but a large number of clusters will be mined, and the relation of the clusters across time is not explored. Therefore, a better solution is to extend the subspace clusters to 3D, which are subspace clusters that persist or evolve across time.

Definition 4 **(3D subspace cluster** $C = (O, A, T)$**)** Let O be a set of objects, A be a set of attributes, and T be a set of timestamps T. Cuboid $O \times A \times T$ forms a subspace cluster $C = (O, A, T)$ if

- the homogeneous function $h(C)$ is satisfied
- the support function $\pi(C)$ is satisfied
- $\forall o \in O : o$ is allowed to be in other subspace cluster C'

Figure 1b shows an example of a 3D subspace cluster $C = (O, A, T)$, which is a sub-cuboid $O \times A \times T$.

Desired properties of the clusters:

- *Homogeneity* The homogeneous function is extended to 3D, which measures the homogeneity of the values in the sub-cuboid $O \times A \times T$.
- *Significant size* The support function measures the size of the sub-cuboid $O \times A \times T$.
- *Maximal clusters* Same as basic subspace clusters.
- *Concept of subspace in the third (time) dimension* Having the concept of subspace in the time dimension means that the 3D subspace cluster exists in some timestamps of the dataset, and not across all time. It is highly unlikely to have clusters that are persistent in each timestamp of a dataset, especially in a dataset that has a large number of timestamps. Moreover, this persistency may not be important in some problems, for example, if the third dimension is location. However, in certain problems, it may be desirable to mine clusters that exist in subsets of continuous timestamps.

Desired properties of the algorithm:

- *Complete and stable result* Same as basic subspace clusters.
- *Efficiency* Efficient mining is generally more difficult to achieve in 3D data than 2D data, and using 2D subspace clustering algorithms to mine 3D subspace clusters is not an efficient solution. For example, we can mine 2D subspace clusters in each timestamp, and then use these 2D subspace clusters as candidates to mine 3D subspace clusters. This approach can be highly inefficient if a large number of redundant 2D subspace clusters are generated, which are not part of any 3D subspace clusters. Thus, 3D subspace clustering algorithm that can aggressively prune the 3D search space is needed for efficient mining of the clusters.

Categorical data In categorical data, its values have no natural order, and there is no distance information between them. These two characteristics make categorical data hard to cluster, as existing distance measures cannot be used. Hence, in a subspace cluster $C = (O, A)$ of categorical data, the set of objects O have the same value for each attribute $a \in A$. Hence, the subspace cluster is formed by a set of attribute values and a set of objects.

Desired properties of the clusters:

- *Homogeneity* The homogeneous criterion is generally the identity of the values for each attribute in the subspace cluster. Hamming distance or Jaccard index (Guha et al. 1999) can be used on the objects, if some dissimilarity in attributes are allowed.
- *Significant size, Maximal clusters* Same as basic subspace clusters.

Desired properties of the algorithm:

- *Complete and stable result, Efficiency* Same as basic subspace clusters.

Stream data Stream data is not to be confused with the 3D data (*object* $\times$ *attribute* $\times$ *time*). In the 3D data, each of its dimension is finite, while the stream data is a 2D

data (*object* × *attribute*), with one of the dimension being infinite and the other being finite.

Let us denote $\mathbb{D} = \mathbb{O} \times \mathbb{A}$ as a stream data, and there are two types of stream data. The first type has streaming objects $\mathbb{O} = \{o_1, \ldots, o_i, \ldots\}$ with a fixed set of attributes $\mathbb{A}$. The second type has a fixed set of objects (each object is still considered as a stream) but with streaming attributes. In this type, the objects only contain one attribute, but values of this attribute are streamed into the data and are indexed by timestamps. Thus, $\mathbb{A} = \{a_{t_1}, \ldots, a_{t_i}, \ldots\}$, where a_{t_i} denotes the attribute a at time t_i.

Desired properties of the clusters:

– *Homogeneity, Significant size, Maximal clusters* Same as basic subspace clusters.

Desired properties of the algorithm:

– *Complete and stable result* Same as basic subspace clusters.
– *Efficiency* Due to the streaming nature of the data, the algorithm is generally required to read the data once or the mining is restricted within a fixed window of the stream data.
– *Up-to-date clustering* The subspace clusters have to be constantly updated with respect to the continuous data stream.

Noisy data Noisy data can contain erroneous, missing or uncertain values, and real-world data are notorious for being noisy, such as microarray gene expression data in biology, sensors data in smart home systems, stock prices and financial ratios data in finance, etc. Therefore, it is vital to have subspace clustering approaches that can handle noisy data, so that useful clusters can still be found in the presence of noise.

The representation of uncertain data is different from standard matrix $\mathbb{D} = \mathbb{O} \times \mathbb{A}$. For uncertain data, each attribute value of an object is sampled multiple times, i.e., there are multiple values x_{oa} of the object o on attribute a, and so, each value is represented as a probability distribution.

Desired properties of the clusters:

– *Homogeneity, Significant size, Maximal clusters* Same as basic subspace clusters.
– *Tolerate noisy data* The subspace cluster $C = (O, A)$ should be able to tolerate missing or erroneous values. In other words, the clustering approach should be able to infer if an erroneous or missing value should be part of a cluster.
– *Handle uncertain data* The subspace cluster should be able to account for the uncertainty of the data.

Desired properties of the algorithm:

– *Complete and stable result, Efficiency* Same as basic subspace clusters.

2.3.2 Improving clustering results

The clustering result can be improved in a variety of ways, from mining *significant* subspace clusters to using parameter-insensitive clustering approaches. In parameter-insensitive clustering, the 'true' subspace clusters are discovered and they are not manifestations of skewed parameter settings.

Significant subspace clustering Significant subspace clusters are clusters that are intrinsically prominent in the data, and are more interesting or meaningful than other clusters. They are usually small in numbers, which are easier to analyze.

Desired properties of the clusters:

- *Homogeneity, Significant size, Maximal clusters* Same as basic subspace clusters.
- *Significant clusters* There is no universal accepted definition of significant subspace clusters. As a rule of thumb, a subspace cluster is significant if it is more interesting or meaningful than other clusters, based on criteria of the clustering approach.

Desired properties of the algorithm:

- *Complete and stable result, Efficiency* Same as basic subspace clusters.

Semi-supervised subspace clustering In semi-supervised subspace clustering, additional knowledge such as domain knowledge or user's preference is used in the clustering process.

Desired properties of the clusters:

- *Homogeneity, Significant size, Maximal clusters* Same as basic subspace clusters.
- *Semi-supervised clusters* The additional knowledge is used to improve the quality of the subspace clusters.

Desired properties of the algorithm:

- *Complete and stable result* Same as basic subspace clusters.
- *Efficiency* The additional knowledge is used to improve the efficiency of the algorithm, by guiding the algorithm to regions of the search space with clusters and pruning regions without clusters.

Overcoming parameter-sensitive subspace clustering The current paradigm of subspace clustering requires the user to set parameters, and clusters that satisfy these parameters are returned. We can broadly classify the parameters into *cluster parameters* and *algorithm parameters*, where cluster parameters are used in defining the clusters and algorithm parameters are used in guiding the clustering process. For example, user-selected centroid is a cluster parameter, as the cluster is formed by objects similar to the centroid. On the other hand, the parameter k in k-means clustering is an algorithm parameter, as it determines the clustering process.

In some cases, the parameter can be both cluster and algorithm parameter. For example, constraints such as must-link and cannot-link (Wagstaff et al. 2001) are cluster parameters, as they respectively indicate which objects should be clustered together, and which objects should not be clustered together. These constraints are also algorithm parameters, as they can be used in improving the efficiency of the clustering process. In another example, the minimum size threshold of the cluster is a parameter of both cluster and algorithm, as it defines the size of the cluster, and it can be used to prune regions of the search space which do not contain clusters that have the minimum size.

From another perspective, we can also classify the parameters as *tuning parameters* and *semantical parameters*, based on their usage (Kriegel et al. 2007). Tuning parameters are used to tune the efficiency of the algorithm or the quality of the clusters, and

Table 1 The parameter matrix

	Tuning parameter Tune the quality of the clustering result or efficiency of the algorithm	*Semantical parameter* Meaningful parameters and the domain knowledge or preference of the user can be incorporated into them
Cluster parameter Define the cluster	Minimum size threshold, distance threshold	Minimum size threshold, must-link and cannot-link constraints, distance measure, user-selected centroids
Algorithm parameter Guide the clustering process	Minimum size threshold, maximum number of iterations, number of clusters, sliding window size	Minimum size threshold, must-link and cannot-link constraints

A parameter can be categorized into two categories: (1) cluster or algorithm parameter, and (2) tuning or semantical parameter. The cells in the matrix are examples of the different types of parameters

semantical parameters are meaningful parameters that describe the semantics of the clusters and domain knowledge of the user can be incorporated into these parameters. For example, the maximum number of iterations allowed in an optimization algorithm (Nocedal and Wright 2006) is a tuning parameter, as it determines the accuracy and efficiency of the algorithm. On the contrary, the must-link and cannot-link constraints are semantical parameters, as they are based on the domain knowledge of the user.

In certain cases, a parameter can either be a tuning parameter or a semantical parameter, depending on the user. For example, if the user has domain knowledge, then minimum size threshold can be a semantical parameter, whereas if the user does not have the domain knowledge, then minimum size threshold is a tuning parameter.

Based on these classifications, we can present the categories of the parameters as a parameter matrix, shown in Table 1. Semantical parameters can be desirable as they enable the users to flexibly control the results, based on their knowledge. On the other hand, tuning parameters are usually non-meaningful and non-intuitive, making it difficult for the user to set the correct parameters.

Setting of tuning parameters should be avoided (Kriegel et al. 2007). If these parameters are set based on the biased assumptions of the user, highly skewed clusters will likely to be generated. Furthermore, subspace clustering algorithms are typically abound with tuning parameters, thereby increasing the burden of the user.

To relieve the user of this dilemma, these parameters should be few in numbers. If it is not possible, then the tuning parameters should be insensitive to the clustering results, so that the clusters do not change dramatically under slight variations of the tuning parameters.

In the rest of this survey, we focus on tuning parameters when we discuss on parameter-light and parameter-insensitivity properties.

Desired properties of the clusters:

- *Homogeneity, Significant size, Maximal clusters* Same as basic subspace clusters.
- *Parameter-light* The subspace cluster should be parameter-light, where only few tuning parameters are used in defining the cluster.
- *Parameter-insensitivity* The subspace cluster is parameter-insensitive when it can be obtained under a range of settings on the tuning parameters.

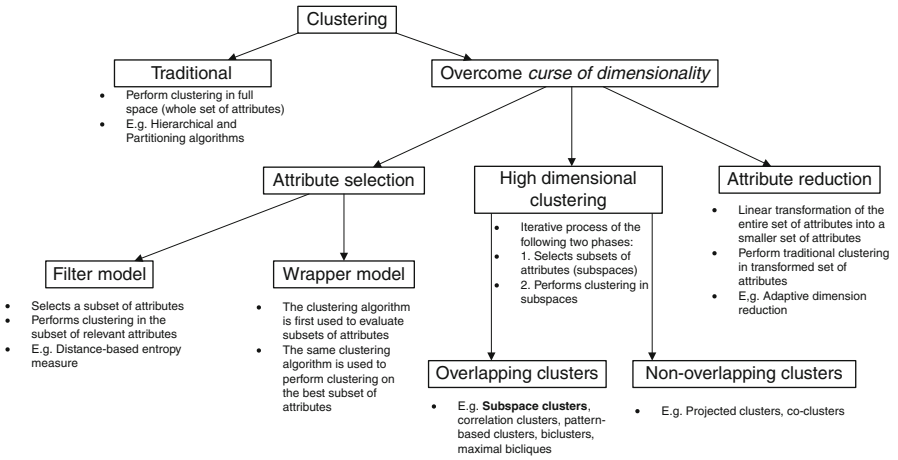


Fig. 2 Overview of the different approaches to overcome the curse of dimensionality

Desired properties of the algorithm:

– *Complete and stable result, Efficiency* Same as basic subspace clusters.
– *Parameter-light* The algorithm is parameter-light when few tuning parameters are required to control the running of the algorithm.
– *Parameter-insensitivity* The algorithm is parameter-insensitive when it is efficient and the result is stable under a range of settings on the tuning parameters.

3 Related high dimensional clustering techniques

We briefly discuss high-dimensional clustering techniques that are related to subspace clustering. Figure 2 presents an overview of the different high dimensional clustering techniques.

3.1 Attribute selection and reduction

There are two existing solutions to the curse of dimensionality problem in high dimensional clustering, namely *attribute reduction* and *attribute selection*[1].

3.1.1 Attribute reduction

In attribute reduction, the large set of attributes is transformed into a smaller set of attributes, and traditional clustering can be applied on the transformed data without suffering from the curse of dimensionality (Ding et al. 2002). The transformation is usually based on Principal Component Analysis (PCA) or Singular Value Decomposition

[1] Also known as feature reduction and feature selection.

(SVD), which results in the transformed set of attributes being a linear combination of the original set of attributes.

However, this approach itself has several weaknesses. Firstly, PCA or SVD is highly sensitive to noisy objects, and since the transformation is dependent on all objects, the transformed set of attributes may be distorted by the noisy objects, and are not helpful in distinguishing the clusters. Secondly, analysis of clusters is difficult, as the transformed set of attributes bear no semantic meanings. Thirdly, clusters may exist in different subspaces (sets of attributes $A \subseteq \mathbb{A}$), and this information is lost after attribute reduction.

3.1.2 Attribute selection

In attribute selection, a subspace $A \subseteq \mathbb{A}$ in which the objects are homogeneous, is selected for clustering, and there is no transformation of the attributes. There are two main models of attribute selection, the filter and the wrapper models. The filter model selects a set of relevant attributes (Dash et al. 2002), and then a traditional clustering algorithm is applied on them. Thus, it is independent of the clustering algorithm.

In the wrapper model, the clustering algorithm is a black box which is used to evaluate the quality of different subspaces. After the best subspace is found, the same clustering algorithm is used to mine clusters from it (Kohavi and John 1997).

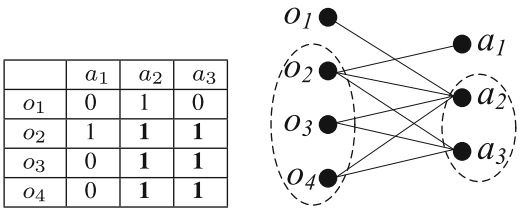
High dimensional clustering (such as subspace clustering and projected clustering) assumes that different sets of objects can be homogeneous in different subspaces; hence high dimensional clustering is an iterative process of selecting relevant subspaces and performing clustering on them, with these two phases being closely interlinked. On the other hand, both filter and wrapper models just find the best subspace and perform clustering on it.

3.2 Projected clustering

Projected clustering (Aggarwal et al. 1999) finds clusters that are similar to subspace clustering, where in a cluster, its set of objects are homogeneous in its set of attributes. As a matter of fact, some projected clustering techniques refer themselves as subspace clustering techniques (Domeniconi et al. 2004; Chan et al. 2004), which leads to confusion between these two types of clustering techniques.

The main difference of projected clustering from subspace clustering is that there is no overlapping of clusters in projected clustering, i.e. an object can only be assigned to a cluster. Hence, projected clustering can be considered as a partitioning technique.

However, partitioning of objects may be a harsh requirement, as allowing an object to be in different clusters allows the understanding of the different perspectives of the object. Details on projected clustering can be found in the survey by Kriegel et al. (2009).



	a_1	a_2	a_3
o_1	0	1	0
o_2	1	**1**	**1**
o_3	0	**1**	**1**
o_4	0	**1**	**1**

(a)

(b)

$id(T)$	Items
o_1	a_2
o_2	a_1, a_2, a_3
o_3	a_2, a_3
o_4	a_2, a_3

(c)

Fig. 3 **a** A binary dataset $\mathbb{D}$ with subspace cluster $\{o_2, o_3, o_4\} \times \{a_2, a_3\}$. **b** The subspace cluster corresponds to a maximal biclique subgraph (the pair of circled vertex sets) in the bipartite graph. **c** The subspace cluster corresponds to a frequent pattern a_2, a_3 with occurrence set $\{o_2, o_3, o_4\}$ in the transactional dataset

3.3 Maximal bicliques and frequent patterns

Let us assume that the dataset is a binary dataset, i.e. $\mathbb{D} = \mathbb{O} \times \mathbb{A} \in \{0, 1\}^{|\mathbb{O}| \times |\mathbb{A}|}$, with an example shown in Fig. 3a. $\mathbb{D}$ can be represented as a bipartite graph G, where both the set of objects $\mathbb{O}$ and the set of attributes $\mathbb{A}$ are sets of vertices respectively, and an edge $\{o, a\}$ exists in the graph if value $x_{oa} = 1$, as shown in Fig. 3b.

Let G denote a bipartite graph representing $\mathbb{D}$, which consists of two sets of disjoint vertices $V(G) = \{\mathbb{O}, \mathbb{A}\}$ and a set of edges $E(G) = \{\{o, a\} | o \in \mathbb{O} \wedge a \in \mathbb{A}\}$. A graph g is a subgraph of G if $V(g) \subseteq V(G)$ and $E(g) \subseteq E(G)$. A subgraph g with $V(g) = \{O, A\}$ is a biclique subgraph of G iff $E(g) = \{\{o, a\} | \forall o \in O \wedge \forall a \in A\}$, i.e. every vertex o in vertex set O is connected to every vertex a in vertex set A. A biclique subgraph g is maximal if there does not exist another biclique subgraph g' such that $V(g) \subseteq V(g')$ and $E(g) \subseteq E(g')$. An example of a maximal biclique subgraph is shown in the circled vertex sets of Fig. 3b.

Mining maximal biclique subgraphs from graph G is equivalent to mining maximal subspace clusters from its 2D binary dataset $\mathbb{D}$. The equivalence is obvious and is proven in (Li et al. 2005). For a maximal biclique subgraph g with $V(g) = \{O, A\}$, the submatrix $O \times A$ in its adjacency matrix are all '1's, which is equivalent to a subspace cluster (O, A). For example, the maximal biclique subgraph of Fig. 3b corresponds to the subspace cluster $(\{o_2, o_3, o_4\}, \{a_2, a_3\})$ of Fig. 3a. Thus, for binary dataset, efficient mining maximal biclique subgraphs algorithm (Liu et al. 2006) can be used to mine subspace clusters.

The dataset $\mathbb{D}$ can also be represented as a transactional dataset, as shown in Fig. 3c. Let the set of attributes $\mathbb{A}$ be a set of items, and a transaction T be a subset of $\mathbb{A}$. The transactional dataset contains rows of transactions, and $id(T)$ is denoted as the transaction identifier of transaction T. For each object $o \in \mathbb{O}$, a transaction with identifier $id(T) = o$ can be created, and the transaction contains items $a \in \mathbb{A}$ where $x_{oa} = 1 \in \mathbb{D}$.

Let $P \subseteq \mathbb{A}$ be a pattern and $occ(P) = \{id(T) | P \subseteq T\}$ be the occurrence set of the pattern P. A pattern P is closed if adding any item $\mathbb{A} \backslash P$ to it will lead to the decrease of its occurrence set. Li et al. (2005) show that a pair of closed pattern P and its occurrence set $occ(P)$ correspond to a biclique subgraph g with $V(g) = \{O, A\}$, where $P = A$ and $occ(P) = O$. Therefore, it is possible to mine closed patterns (Uno et al.

	a_1	a_2	a_3
o_1	1	5	3
o_2	1	5	3
o_3	1	5	3

(a)

	a_1	a_2	a_3
o_1	1	1	1
o_2	5	5	5
o_3	3	3	3

(b)

	a_1	a_2	a_3
o_1	1	2	3
o_2	4	5	6
o_3	7	8	9

(c)

Fig. 4 **a** A pattern based cluster with homogeneity on the attributes, **b** a pattern based cluster with homogeneity on the objects, and **c** a pattern based cluster with homogeneity on both attributes and objects

2004) and do a post-processing to retrieve their occurrence sets, to obtain the maximal biclique subgraphs, which in turn correspond to subspace clusters. For example, the frequent pattern a_2, a_3 with occurrence set $\{o_2, o_3, o_4\}$ in the transactional dataset of Fig. 3c corresponds to the subspace cluster $(\{o_2, o_3, o_4\}, \{a_2, a_3\})$ of Fig. 3a.

If discrete dataset is involved, then quantitative patterns (Srikant and Agrawal 1996; Ke et al. 2006) with their occurrence sets correspond to subspace clusters. Quantitative patterns focus on mining patterns with significant occurrences, and simple occurrence threshold (Srikant and Agrawal 1996) or mutual information (Ke et al. 2006) have been used to determine whether their occurrences are significant. This is different from subspace clusters, which are sets of objects being homogeneous in sets of attributes.

3.4 Pattern based clustering

Pattern based clustering, also known as biclustering, is originally used in analysis of microarray gene expression data (Cheng and Church 2000). In a 2D microarray dataset, the genes are the objects and the samples are the attributes. Similar to subspace clustering, pattern based clustering mines clusters where a cluster is a set of objects that are homogeneous in a set of attributes, and overlapping of clusters is allowed. However, there are two subtle differences in these two types of clusters. First, the submatrix defined by the objects (rows) and attributes (columns) of a pattern based cluster exhibits a pattern. Second, the objects and attributes of a pattern based cluster are treated equally, but this is not the case for a subspace cluster. This equal treatment of pattern based cluster encourages more flexibility in its homogeneity; its homogeneity can be on the attributes, on the objects, or on both attributes and objects.

Homogeneity on the attributes means that the objects are homogeneous in each attribute, e.g. the objects have similar values in each attribute, as shown in the pattern based cluster of Fig. 4a. Homogeneity on the objects means that each object is homogeneous in the attributes, e.g. the attributes have similar values for each object, as shown in the pattern based cluster of Fig. 4b. On homogeneity on both objects and attributes, a simple example is where all values in the cluster have similar values. A more common type of this homogeneity is shifting or scaling of values across attributes and objects. Figure 4c shows a pattern based cluster of shifting homogeneity on both attributes and objects.

The following definition is popularly used in defining the shifting or scaling homogeneity:

Definition 5 (pScore) Given a submatrix $M = \begin{bmatrix} x_{oa} & x_{oa'} \\ x_{o'a} & x_{o'a'} \end{bmatrix} \subseteq O \times A$, the pScore of the submatrix is $|(x_{oa} - x_{oa'}) - (x_{o'a} - x_{o'a'})|$.

The pattern based cluster (O, A) exhibits a shifting pattern if the pScore of any of its submatrix M is less than a user-specified parameter δ (Wang et al. 2002). The pattern based cluster of Fig. 4c has shifting homogeneity as the pScore of any of its submatrix is 0. For example, the pScore of submatrix $\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$, is $|(1-2)-(4-5)| = 0$.

Note that Definition 5 is symmetric, $|(x_{oa} - x_{oa'}) - (x_{o'a} - x_{o'a'})| = |(x_{oa} - x_{o'a}) - (x_{oa'} - x_{o'a'})|$. To find scaling patterns in the submatrix, we simply convert the values of the submatrix into their logarithm form and apply the pScore on the submatrix.

The values in a pattern based cluster (O, A) can also be expressed by the following equation, with α_o and β_a determining the type of homogeneity in the cluster.

$$x_{oa} = \mu + \alpha_o + \beta_a \tag{1}$$

μ is a typical value in the cluster, α_o is the adjustment for object $o \in O$ and β_a is the adjustment for attribute $a \in A$. For example, $\alpha_o = 0$, $\beta_a > 0$ mean that the cluster has similar values for each attribute, and $\alpha_o > 0$, $\beta_a > 0$ mean that the cluster has shifting pattern. For example, the pattern based cluster of Fig. 4c has $\mu = 1$, $\alpha_{o_3} = 6$, $\beta_{a_3} = 2$, which results in $x_{o_3 a_3} = 1 + 6 + 2 = 9$. Scaling pattern in the cluster can also be expressed by replacing the values with their logarithm form.

For more complex homogeneity, Xu et al. (2006) combine both shifting and scaling patterns by using the equation $x_{oa} = \alpha.x_{o'a} + \beta$, where α, β are the scaling and shifting factors respectively. There are also some clusters which are hybrids of subspace cluster and pattern based cluster, such as the tricluster (Zhao and Zaki 2005). Comprehensive surveys on pattern based clustering can be found in (Madeira and Oliveira 2004; Jiang et al. 2004b; Tanay et al. 2004), and a detailed comparison of subspace clustering and pattern based clustering can be found in (Kriegel et al. 2009). The close relationship between frequent pattern mining and subspace clustering has been pointed out recently (Vreeken and Zimek 2011).

3.5 Correlation clustering

Equation 1 shows that the homogeneity of pattern based clusters can be expressed as simple linear equations. Correlation clustering (Böhm et al. 2004) generalizes this concept and finds clusters that display any types of linear equations. More specifically, a correlation cluster is a set of objects whose values are positively or/and negatively correlated on a set of attributes.

Correlation clustering usually does not allow overlapping clusters, though this is based on algorithmic reasons and not in the nature of the task definition. For example, algorithm 4C (Böhm et al. 2004) does not mine overlapping correlation clusters. However, if we consider the correlation cluster model (Achtert et al. 2006b), mining overlapping clusters is not an issue.

More details on correlation clustering can be found in (Kriegel et al. 2009). A similar approach to correlation clustering is ASI clustering (Li et al. 2004), where objects are clustered based on subspaces where there are linear combinations of the attributes. However, ASI clustering is more related to projected clustering, as it partitions the objects into clusters.

3.6 Co-clustering

Co-clustering is developed primarily to cluster *word-document* dataset (Dhillon et al. 2003), which is a 2D contingency table represented as a matrix. In the matrix, the rows and columns correspond to documents and words respectively, with the value x_{oa} in the matrix representing the probability or the occurrence of the word a (column) in the document o (row).

The contingency matrix is partitioned into l sets of documents $\{O_1, \ldots, O_l\}$ and k sets of words $\{A_1, \ldots, A_k\}$, and the user is required to specify the number of partitions l, k. A set of documents O and a set of words A are then used to form a co-cluster $C = (O, A)$. Thus, there is no overlapping of clusters in co-clustering.

Most of the co-clustering techniques find co-clusters such that the occurrences or probabilities of the words in the documents are similar in a co-cluster. Graph partitioning (Dhillon 2001; Rege et al. 2006) and mutual information (Dhillon et al. 2003; Sim et al. 2009a; Gao et al. 2006; Chiaravalloti et al. 2006) are the common techniques used in co-clustering, with mutual information as the more popular one. In mutual information technique, the optimal set of co-clusters is obtained by minimizing the mutual information between $I(\mathbb{O}; \mathbb{A})$ and $I(\{O_1, \ldots, O_l\}; \{A_1, \ldots, A_k\})$.

Co-clustering is also applied in the 2D microarray dataset (Pensa and Boulicaut 2008), but co-clustering is a partitioning approach that mines non-overlapping clusters, which is different from the overlapping clusters of pattern based clustering.

3.7 Summary

Table 2 gives a comparison of the properties of the related high-dimensional techniques. On the types of data that are handled by the techniques, we denote continuous, discrete, categorical, word-document and binary as CO, D, CA, WD and B respectively. We differentiate word-document data from quantitative data, as word-document data can either be discrete (when occurrences of words in documents are used) or continuous (when probabilities of words in documents are used).

We indicate if the high-dimensional techniques mine overlapping clusters and 3D clusters. We also indicate the commonly used homogeneity function in the techniques: For subspace cluster or projected cluster (O, A), it has similar values for each attribute $a \in A$, in its submatrix $O \times A$. For maximal biclique or frequent pattern, if it is represented by a binary subspace cluster (O, A), it has value '1's for each attribute $a \in A$, in its binary submatrix $O \times A$. For pattern based cluster or correlation cluster (O, A), the values in its matrix $O \times A$ can be expressed in a linear equation. For co-cluster (O, A), the values in its matrix $O \times A$ have similar probabilities or co-occurrences.

Table 2 A comparison of the properties of different high-dimensional clustering techniques

High-dimensional clustering technique	Data type it handles	Overlapping clusters	3D clusters	Commonly used homogeneity function on their cluster (O, A)
Subspace clustering	CO, D, CA, WD, B	√	√	Similar attribute values in submatrix $O \times A$
Projected clustering	CO, D, CA, WD, B			Similar attribute values in submatrix $O \times A$
Maximal bicliques	CA, B	√	√	Same attribute values in submatrix $O \times A$
Frequent patterns	D, CA, B	√	√	Same attribute values in submatrix $O \times A$
Pattern based clustering	CO, D, WD	√	√	Attribute values expressed in simple linear relation, in submatrix $O \times A$
Correlation clustering	CO, D, WD	√		Attribute values expressed in complex linear relation, in submatrix $O \times A$
Co-clustering	CO, D, WD, B			Attribute values with similar probabilities or co-occurrences, in submatrix $O \times A$

CO, D, CA, WD and B denote continuous, discrete, categorical, word-document and binary respectively

4 Basic subspace clustering: approaches and their definitions

The approaches to solving the basic subspace clustering problem have three main characteristics. First, they handle quantitative 2D dataset. Second, their homogeneous function is distance based. Third, the significance of the size is determined by user-specified thresholds. Their main difference lies in their homogeneous and support functions. Figure 5a shows an example of objects on the hyperplane of subspace $\{a_1, a_2\}$, and Fig. 5b–c show the subspace clusters mined by the different approaches. Note that the subspace clusters $C = (O, A)$ are submatrices $O \times A$ that are axis-parallel to the dataset, and not to the hyperplane of the subspace.

4.1 Grid based subspace clustering

In grid based subspace clustering (Agrawal et al. 1998), the data space is partitioned into grids, and dense grids containing significant number of objects are used to form subspace clusters.

The domain of each attribute a, $D(a)$, is first partitioned into ξ intervals, $u_1^a, \ldots, u_\xi^a$, each of equal length $\frac{R_a}{\xi}$. Given a set of attributes A, we denote $\mathbf{u} = \{u^a | a \in A\}$ as an $|A|$-attribute unit, which is the combination of one interval from each attribute of A. Two $|A|$-attribute units $\mathbf{u} = \{u^a | a \in A\}$, $\mathbf{v} = \{v^a | a \in A\}$ have *common face* if (1) $\forall i \in \{1, \ldots, |A| - 1\} : u^i = v^i$ and (2) $u^{|A|}$ and $v^{|A|}$ are contiguous. Two units $\mathbf{u}$, $\mathbf{v}$ are *connected* if they have common face or if there exists another unit $\mathbf{w}$ such that there is a common face between $\mathbf{u}$ and $\mathbf{w}$, and $\mathbf{w}$ and $\mathbf{v}$.

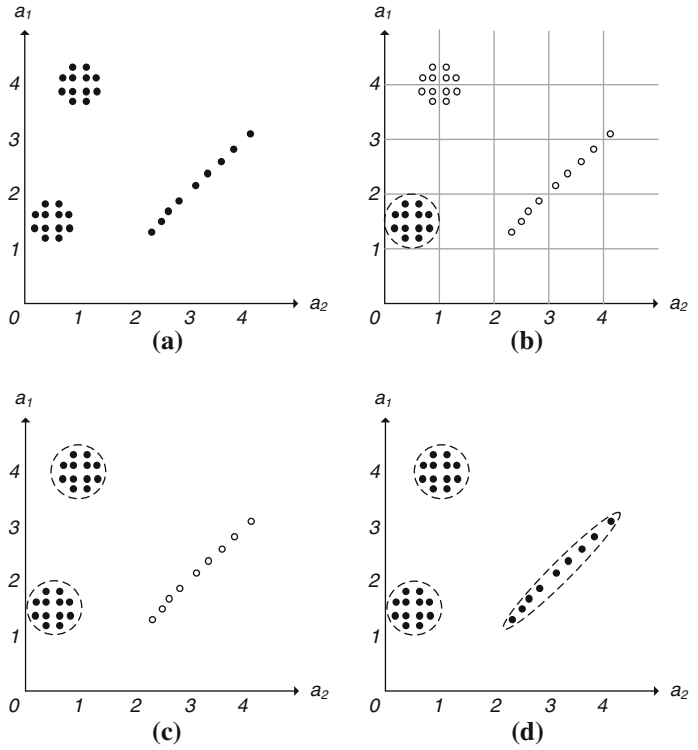


Fig. 5 **a** Objects in the hyperplane of subspace $\{a_1, a_2\}$, **b** grid based subspace clusters, **c** window based subspace clusters, and **d** density based subspace clusters

Definition 6 (Grid based subspace cluster) Given a matrix $O \times A$, we denote $U = \{\mathbf{u_1}, \ldots \mathbf{u_n}\}$ as the set of connected $|A|$-attribute units contained in matrix $O \times A$. $C = (O, A)$ is a subspace cluster if

- $h(C) = \forall \mathbf{u} \in U : \forall u^a \in \mathbf{u} : |x_{oa} - x_{o'a}| \leq \frac{R_a}{\xi}$ for any $x_{oa}, x_{o'a} \in u^a$
- $\pi(C) = \forall \mathbf{u} \in U : \frac{|\{o | \forall u^a \in \mathbf{u} : x_{oa} \in u^a \wedge o \in O|}{|\mathbb{O}|} \geq \tau$

Figure 5b shows an example of the result of grid based subspace clustering, with each cell in the grid as a unit. In this example, the unit is considered dense if it has at least 5 objects.

Properties of the clusters:

- *Homogeneity* The intervals in the units induced the homogeneity of the cluster, given that the maximum distance between objects in an interval is $\frac{R_a}{\xi}$. There are some weaknesses in using intervals. First, as the intervals are non-overlapping, and wrong positioning of the grids may lead to 'truth' subspace clusters being overlooked. Second, setting a fixed size on the intervals using ξ may result in poor clustering quality, as the distribution of the objects in each attribute is different. Nagesh et al. (2001) proposed using adaptive grids to overcome this problem, which varies the interval sizes based on the data distribution. Figure 5b shows a

subspace cluster in the region $a_1 = 4, a_2 = 1$ being overlooked due to the poor positioning of the grid.

- *Size* The support function requires the units to be dense, where the density is specified by the density threshold τ. Setting a fixed τ may degrade the clustering quality, as a high threshold leads to a small number of subspace clusters, while a low threshold leads to a large number of clusters. To circumvent this problem, Sequeira and Zaki (2004) proposed a non-linear monotonically decreasing threshold which decreases as the size of the subspace increases.
- *Number of parameters* There are two tuning parameters to set, the density threshold τ and the number of intervals ξ.
- *Parameter sensitivity* The cluster is sensitive to the tuning parameters. If wrong τ and ξ are set, actual dense units may be overlooked. For example, the subspace cluster in the region $a_1 = 4, a_2 = 1$ can be overlooked due to wrong parameter settings. The parameters are difficult to set as they are non-meaningful and non-intuitive.
  A possible remedy is to try a range of parameter settings, and check for results which are stable in a particular range of parameter settings. Another possible option is to adjust the parameter setting until a suitable number of clusters is obtained.

4.2 Window based subspace clustering

Window based subspace clustering (Liu et al. 2009) is developed primary to overcome the weaknesses of grid based subspace clustering. In window based subspace clustering, a sliding window is slided over the domain of each attribute to obtain overlapping intervals, which are then used as building blocks for subspace cluster. Thus, the chances of 'true' subspace clusters being overlooked are greatly reduced.

Definition 7 (Window based subspace cluster) $C = (O, A)$ is a subspace cluster if

- $h(C) = \forall a \in A: |x_{oa} - x_{o'a}| \leq R_a.\delta$, for any pair of objects $o, o' \in O$.
- $\pi(C) = |O| \geq min_o \wedge |A| \geq min_a$

If the values are normalized to [0, 1], then the homogeneous function is simply $|x_{oa} - x_{o'a}| \leq \delta$.

Properties of the clusters:

- *Homogeneity* The homogeneity of the cluster is based on L^∞ norm, i.e. the distance between two objects in the cluster is dependent on the attribute $a \in A$ which gives the largest distance between them. Parameter δ determines the size of the sliding window, which controls the distance between objects in the cluster.
- *Size* The size of the cluster is determined by the parameters min_o and min_a.
- *Number of parameters* There are three tuning parameters min_o, min_a and δ to set.
- *Parameter sensitivity* The cluster is sensitive to the tuning parameters.

Figure 5c shows an example of the result of window based subspace clustering. Unlike density based subspace cluster, window based subspace clustering does not

mine arbitrarily shaped clusters, hence it does have the problem of mining undesirable elongated clusters. Details of density based subspace clustering is explained in the next section.

4.3 Density based subspace clustering

Kailing et al. (2004) proposed density based subspace clustering, which overcomes the problems of grid based subspace clustering by dropping the usage of grids. Moreover, it is able to mine arbitrarily shaped subspace clusters in the hyperplane of the dataset.

Let $||o - o'||_p^A = (\sum_{a \in A} |x_{oa} - x_{o'a}|^p)^{\frac{1}{p}}$ be the distance between objects o and o' in L^p-norm, projected on the subspace A. We denote the neighborhood of object o on subspace A as $\mathcal{N}_\epsilon^A(o) = \{o' | o' \in \mathbb{O}, ||o - o'||_p^A \leq \epsilon\}$, where ϵ controls the closeness of the objects on subspace A.

Definition 8 (Density based subspace cluster) $C = (O, A)$ is a subspace cluster if

- $h(C) = \forall o, o' \in O : \exists k : \forall i = 1, \ldots, k-1 : \exists q_i \in O : \|q_i - q_{i+1}\|_p^A \leq \epsilon \wedge q_1 = o, q_k = o' \wedge \forall q_i (i = 2, \ldots, k-1) : |\mathcal{N}_\epsilon^A(q_i)| \geq m$
- $\pi(C) = \forall o \in O : |\mathcal{N}_\epsilon^A(o)| \geq m \vee (o \in \mathcal{N}_\epsilon^A(q) \wedge |\mathcal{N}_\epsilon^A(q)| \geq m)$

Properties of the clusters:

- *Homogeneity* The cluster can be seen as a chain of objects, as the homogeneity function $h(C)$ states that two objects are in a cluster in subspace A, if there is a chain of objects between them, such that each closest pair of objects q_i, q_{i+1} (these objects are also in the cluster) satisfy the distance constraint. Hence, arbitrarily shaped clusters can be found.
  The calculation of the distance between objects is in L^p-norm, and more meaningful and stable results can be obtained by setting a low p (Hinneburg et al. 2000; Aggarwal et al. 2001).
  Due to the curse of dimensionality, the distance between the objects in the subspace A increases as the size of A increases. This is a common problem that also affects grid and window based subspace clustering, but more research on mitigating this problem is done in density based subspace clustering (Assent et al. 2007; Achtert et al. 2007; Kriegel et al. 2005).
  Algorithm DUSC (Assent et al. 2007) mitigates this problem by using a density measure that is adaptive to the size of A, but it does not have monotonic properties for efficient pruning of the search space. Algorithms DiSH (Achtert et al. 2007) and FIRES (Kriegel et al. 2005) mitigate this problem by calculating the distance between objects in each attribute of the subspace A, i.e. $\forall a \in A : ||o - o'||_p^a$, instead of the whole subspace A. Thus, their main algorithmic effort is in combining clusters found in single interesting attributes.
- *Size* The size of the cluster is determined by the parameter m, where m controls the neighborhood density of an object on subspace A. It is important to set $m > 1$, so that the cluster is a m-link cluster, instead of the undesirable single-link cluster.
- *Number of parameters* There are two tuning parameters m and ϵ to set.

– *Parameter sensitivity* The cluster is sensitive to the tuning parameters.

Figure 5d shows an example of the result of density based subspace clustering.

5 Enhanced subspace clustering: approaches and their definitions

We present the existing approaches to solve the enhanced subspace clustering problems described in Sect. 2.3, and their cluster definitions.

Note that we do not dedicate a section on overcoming parameter-sensitive subspace clustering. Instead, we discuss the approaches to overcome parameter-sensitive problem in the other enhanced subspace clustering sections, as the main contributions of these approaches lie in solving other enhanced subspace clustering problems.

5.1 Handling complex data

5.1.1 3D data

Subspace clustering in binary 3D dataset In binary 3D dataset, the values x_{oat} are binary, either '0' or '1', i.e. $\mathbb{D} = \mathbb{O} \times \mathbb{A} \times \mathbb{T} \in \{0, 1\}^{|\mathbb{O}| \times |\mathbb{A}| \times |\mathbb{T}|}$.

The most common definition of binary 3D subspace cluster is as follows:

Definition 9 (Binary 3D subspace cluster) $C = (O, A, T)$ is a subspace cluster if

– $h(C) := \frac{\sum_{x_{oat} \in C} x_{oat}}{|O|.|A|.|T|} = 1$
– $\pi(C) := |O| \geq min_o \wedge |A| \geq min_a \wedge |T| \geq min_t$

Properties of the clusters:

– *Homogeneity* The binary 3D subspace cluster is a sub-cuboid $C = O \times A \times T$ which contains all '1's. This means that the set of objects O have the set of attributes A, across the set of timestamps T. This definition is also known as frequent closed cube (Ji et al. 2006), closed 3-set (Cerf et al. 2008, 2009), frequent tri-sets (Jaschke et al. 2006) and cross-graph quasi-biclique subgraphs (Sim et al. 2011).
– *Size* Three parameters min_o, min_a and min_t are used to determine the size of a cluster.
– *Number of parameters* There are three tuning parameters to set, min_o, min_a and min_t.
– *Parameter sensitivity* The cluster is sensitive to the tuning parameters.
– *Concept of subspace in the time dimension* Concept of subspace exists in the time dimension.

Cerf et al. (2008) proposed closed n-sets, which are n-dimensional binary subspace clusters, but we focus our attention to closed 3-sets, since 3D datasets are more common in the real world. Cross-graph quasi-biclique subgraphs are noise-tolerant 3D subspace clusters, and its details are in Sect. 5.1.4.

Let us assume that the values in the dataset are not binary, but are association weights in the range [0, 1], i.e. x_{oat} is the weight of the object o on attribute value a at

time t. Georgii et al. (2010) proposed mining 3D subspace clusters from this type of dataset. The clusters are based on density but the desired clusters are still sub-cuboids $C = O \times A \times T$ containing all '1's.

Definition 10 (Dense 3D subspace cluster) $C = (O, A, T)$ is a subspace cluster if

- $h(C) := \frac{\sum_{x_{oat} \in C} x_{oat}}{|O|.|A|.|T|} \geq \theta$
- $\pi(C) := |O| > 1 \vee |A| > 1 \vee |T| > 1$

Properties of the clusters:

- *Homogeneity* The homogeneity is density based and θ is the parameter controlling the density of the cluster. Setting $\theta = 1$ will lead to the same homogeneous function of Definition 9.
- *Size* There is no requirement of the size, but the set of objects, attributes or timestamps should not be singleton.
- *Number of parameters* Only one tuning parameter is needed, θ.
- *Parameter sensitivity* The cluster is sensitive to the tuning parameter θ.
- *Concept of subspace in the time dimension* Concept of subspace exists in the time dimension.

Similar to closed n-sets (Cerf et al. 2008), the approach proposed by Georgii et al can mine dense n-dimensional subspace clusters.

Subspace clustering in quantitative 3D dataset Subspace clustering in quantitative 3D dataset is more complex than in binary 3D dataset, as the homogeneity of the clusters is not just a matter of '0's and '1's.

A simple solution is proposed by Sim et al. (2011), in which the values are discretized and converted into binary dataset, and then 3D binary subspace clusters are mined from it. However, this lossy conversion of data has several weaknesses. Selecting the appropriate discretization method is non-trivial, and information may be lost during the discretization. Moreover, the binary dataset may increase exponentially if the discretization is too fine, as each attribute of the binary dataset corresponds to an interval of the discretized attribute values of the original dataset.

Jiang et al. (2004a) mine 3D subspace clusters, known as coherent gene clusters, directly from quantitative 3D dataset, but they 'flatten' the 3D dataset into 2D, which results in having the strict requirement that the clusters must be persistent in every timestamp of the dataset.

Let $\overline{x}_{oa} = \sum_{t \in \mathbb{T}} \frac{x_{oat}}{|\mathbb{T}|}$ be the average value of object o on attribute a, over time $\mathbb{T}$.

Definition 11 (Coherent gene cluster) $C = (O, A, \mathbb{T})$ is a subspace cluster iff

- $h(C) := \frac{\sum_{t \in \mathbb{T}} (x_{oat} - \overline{x}_{oa})(x_{oa't} - \overline{x}_{oa'})}{\sqrt{\sum_{t \in \mathbb{T}} (x_{oat} - \overline{x}_{oa})^2} \sqrt{\sum_{t \in \mathbb{T}} (x_{oa't} - \overline{x}_{oa'})^2}} \geq \delta$, for any $a, a' \in A$
- $\pi(C) := |O| \geq min_o \wedge |A| \geq min_a$

Properties of the clusters:

- *Homogeneity* The homogeneous function is Pearson's correlation coefficient (Yang et al. 2002). In the coherent gene cluster, the values of its objects O are linearly correlated across time in its subspace A. However, there is no requirement

that the values must be closed together on its subspace A. Thus, coherent gene cluster is more related to pattern based cluster than subspace cluster. Both scaling and shifting patterns (explained in Sect. 3.4) can be captured under Pearson's correlation coefficient.

- *Size* The size of the cluster is determined by the parameters min_o and min_a.
- *Number of parameters* There are three parameters to set tuning min_o, min_a and δ. Parameter δ controls the strictness of the correlation of the cluster across time.
- *Parameter sensitivity* The cluster is sensitive to the tuning parameters.
- *Concept of subspace in the time dimension* There is no concept of subspace in the time dimension. The cluster must be persistent in every timestamp of the dataset, which can be too strict. It is also unlikely to mine any clusters when the time dimension is large. Coherent gene cluster is more suitable for datasets that have small number of timestamps.

Sim et al. (2010b) proposed semi-supervised, quantitative 3D subspace cluster which also requires the cluster to be persistent across time in the dataset. Its details is discussed in Sect. 5.2.2.

Zhao and Zaki (2005) proposed triclusters, which is a variant of window based subspace cluster. Unlike the coherent gene cluster, it does not 'flatten' the 3D dataset into 2D and the concept of subspace exists in the time dimension. Tricluster is a highly flexible cluster model that can be morphed into a wide variety of 3D subspace clusters, such as clusters that have similar values, clusters that exhibit shifting or scaling patterns, etc. To this end, the homogeneous function of the window based subspace cluster is extended to the object and time dimension, and the pScore (Definition 5) is used to detect the shifting or scaling patterns.

Let M be an arbitrary 2×2 submatrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ of a 3D subspace cluster $C = (O, A, T)$, i.e. $M \subseteq O \times A$ for some $t \in T$, $M \subseteq O \times T$ for some $a \in A$ or $M \subseteq A \times T$ for some $o \in O$.

Definition 12 (Tricluster) $C = (O, A, T)$ is a subspace cluster if

- $h(C) := \begin{cases} 1.\ \forall x_{oat}, x_{o'a't'} \in C : |x_{oat} - x_{o'a't'}| \leq \delta \\ \qquad \text{where } \delta = \begin{cases} \delta_o \text{ if } a = a' \wedge t = t' \\ \delta_a \text{ if } o = o' \wedge t = t' \\ \delta_t \text{ if } o = o' \wedge a = a' \end{cases} \\ 2.\ \text{For any } M \text{ of } C : pScore(M) \leq \epsilon \end{cases}$
- $\pi(C) := |O| \geq min_o \wedge |A| \geq min_a \wedge |T| \geq min_t$

Properties of the clusters:

- *Homogeneity* The first criterion of the homogeneous function controls the similarities of the values in triclusters. It is similar to the homogeneous function of window based subspace cluster, except that it is extended to both attribute and time dimensions. The second criterion of the homogeneous function checks if shifting or scaling patterns exist in the triclusters, which is based on pScore (Definition 5), a common criterion used in pattern based clusters. Thus, tricluster is a hybrid of subspace and pattern based clusters.

- *Size* The size of the cluster is determined by the parameters min_o, min_a and min_t.
- *Number of parameters* The flexibility of tricluster comes at a cost. The user is required to set 7 tuning parameters $\delta_o, \delta_a, \delta_t, \epsilon, min_o, min_a, min_t$, which can be burdensome. Tweaking them to define the cluster of interest requires some effort and the details are given in (Zhao and Zaki 2005). For example, if $\delta^o \approx 0, \delta^a \approx 0, \delta^t \neq 0, \epsilon \approx 0$, then for each timestamp t of a tricluster, the submatrix $O \times A$ in timestamp t has similar values, but the values display shifting or scaling patterns across time in T.
- *Parameter sensitivity* The cluster is sensitive to the tuning parameters.
- *Concept of subspace in the time dimension* Concept of subspace exists in the time dimension.

Sim et al. (2010a) proposed quantitative 3D subspace clusters which are significant and parameter-insensitive, and the details are given in Sect. 5.2.1.

There are 3D clusters that are related to 3D subspace clusters. Xu et al. (2009) proposed mining S^2D^3 clusters, which are variations of triclusters, but they are not axis-parallel. Zhang and Wang (2007) proposed mining *F-clusters*, which are sets of 2D subspace clusters. More specifically, a *F-cluster* is a set of 2D subspace clusters $\{(O, A_t)|t \in T\}$, where A_t is the set of attributes at time t, and each 2D subspace cluster (O, A_t) satisfies the pScore criterion. The set of objects O is fixed in a *F-cluster*, but the set of attributes can change across time.

5.1.2 Categorical data

Zaki et al. (2005) proposed mining subspace clusters in categorical dataset, where the occurrences of their values are higher than expected, under the assumption that the attributes are independent and uniformly distributed.

Let V be a set of values, and $occ(V)$ be the set of objects containing V, i.e. $occ(V) = \{o|\forall x_a \in V : x_{oa} = x_a\}$. Under the assumption that the attributes are independent and uniformly distributed, we can calculate the expected size of $occ(V)$ as $E[|occ(V)|] = |\mathbb{O}| \prod_{x_a \in V} \frac{1}{|D(a)|}$. For example, if the dataset contains 10 objects and has a categorical attribute $gender$ with value $male$, $female$, then the expected number of objects containing the value $male$ is $E[|occ(\{male\})|] = 10.\frac{1}{2} = 5$. For a subspace cluster $C = (O, A)$, we can calculate the expected number of objects on the subspace A as $E[|O|] = |\mathbb{O}| \prod_{a \in A} \frac{|D_O(a)|}{|D(a)|}$.

Definition 13 (Categorical subspace cluster) $C = (O, A)$ is a subspace cluster if

- $h(C) := \forall a \in A : \forall o \in O : x_{oa} = x_a$
- $\pi(C) := \begin{cases} 1.\ |O| \geq \alpha E[|O|] \\ 2.\ \forall a, a' \in A : \forall o \in O : |occ(V)| \geq \alpha E[|occ(V)|], \\ \quad \text{such that } V = \{x_{oa}, x_{oa'}\} \end{cases}$

Properties of the clusters:

- *Homogeneity* The set of objects O have the same value for each attribute $a \in A$.

- *Size* Under the assumption that the attributes are independent and uniformly distributed, the first criterion of the support function requires the number of objects in the cluster to be more than the expected number, while the second criterion requires that the occurrences of each pair of values in the cluster are more than the expected number. The assumption of the attribute being independent and uniformly distributed can be too rigid, as it is possible that the attributes are dependent and the dataset may not be uniformly distributed.
- *Number of parameters* Only one tuning parameter α is needed, which controls the density of the cluster.
- *Parameter sensitivity* The cluster is sensitive to the tuning parameter.

Müller et al. (2009c) also adopted the concept of Definition 13 and used it in density based subspace clustering on heterogeneous dataset, which contains categorical and quantitative attributes.

5.1.3 Stream data

Subspace α-clustering (Kontaki et al. 2008) handles stream data which has a fixed number of objects with streaming attributes. The cluster is defined as follows:

Definition 14 (Subspace α-cluster) $C = (O, A)$ is a subspace cluster if

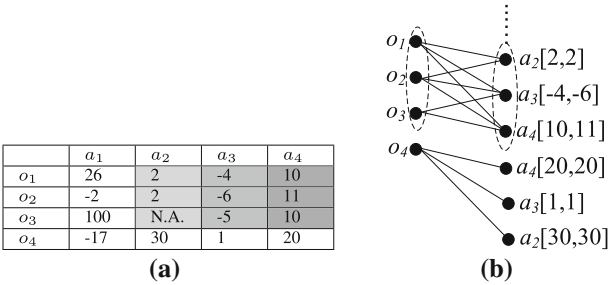
- $h(C) := \forall o, o' \in O, \forall a \in A : |x_{oa} - x_{o'a}| \leq \alpha$
- $\pi(C) := |O| \geq min_o \wedge |A| \geq min_a$

Properties of the clusters:

- *Homogeneity* The homogeneity of subspace α-cluster is similar to window based subspace cluster (Definition 7), except that subspace α-cluster requires the attributes in A to be consecutive in their timestamps.
- *Size* The size of the cluster is determined by the parameters min_o and min_a.
- *Number of parameters* Three tuning parameters are required, min_o, min_a and α. Parameter α controls the differences allowed in the values of an attribute at a timestamp.
- *Parameter sensitivity* The cluster is sensitive to the tuning parameters.

There are several high-dimensional clustering techniques for stream data. In pattern based clustering, Zhang et al proposed mining δ-CC-Clusters (Zhang et al. 2007) from stream data that contains fixed set of objects but with streaming attributes. δ-CC-Cluster is based on pCluster (Wang et al. 2002), where their homogeneity function is shown in Definition 5. Aggarwal et al. (2004) proposed mining projected clusters from stream data that contains streaming objects with fixed set of attributes. Likewise, Kriegel et al. (2011) proposed mining projected clusters from dynamic data. Dynamic data is similar to stream data that contains streaming objects with fixed set of attributes, except that the objects can be continuously inserted, updated or deleted.

In general, the cluster definitions of stream data clustering are usually inherited from its peers in static-data clustering, but the algorithms have to be overhauled to accommodate the potentially infinite stream data. The algorithms are generally required to read the data once and the clustering is usually within a fixed window of the data.



	a_1	a_2	a_3	a_4
o_1	26	2	-4	10
o_2	-2	2	-6	11
o_3	100	N.A.	-5	10
o_4	-17	30	1	20

(a) **(b)**

Fig. 6 **a** A quantitative (discrete or continuous) dataset with the shaded cells as a noise tolerant subspace cluster. **b** The bipartite graph of the quantitative dataset, where the values of the attributes are discretized into intervals. Each vertex represents an interval. The encircled vertex sets represent a quasi-biclique subgraph which corresponds to the subspace cluster of Fig. 6a

The current stream mining paradigm assumes that the data is of 2D matrix $O \times A$, with either the objects O or the attributes A being streaming. With the advancement of data collection, it is possible to have stream data in the form of 3D matrix $O \times A \times T$, where time T is being streamed.

5.1.4 Noisy data

Three diverse approaches have been proposed to handle noisy data, with the graph and Bayesian based approaches handling noisy data represented in the standard matrix format, and the probability based approach handling noisy data represented in uncertain data format.

The noise tolerance concept of the graph based approach is simpler than the others, as it simply 'relaxes' the clustering criteria, by allowing some objects that do not satisfy the clustering criteria to be included in the cluster. On the other hand, the probability and Bayesian based approaches are more sophisticated approaches that use the data distribution to infer if there is noise in the clusters.

Graph based approach As explained in Sect. 3.3, a binary dataset can be represented as a bipartite graph and biclique subgraphs mined from it correspond to subspace clusters. A biclique subgraph does not tolerate noise, as it requires an edge to exist for all pairs of vertices in the subgraph. To tolerate noise, quasi-biclique subgraphs are introduced, where some edges are allowed to be missing in the subgraphs (Li et al. 2008; Sim et al. 2006, 2009b; Mishra et al. 2005; Sim et al. 2011).

For quantitative (discrete or continuous) dataset, discretization is performed first before it is converted into a bipartite graph (Sim et al. 2006, 2009b, 2011). This lossy conversion of data has several weaknesses, as explained in Sect. 5.1.1.

Figure 6a shows an example of a quantitative dataset being discretized and represented as a graph in Fig. 6b. The vertices on the right side of the bipartite graph represent intervals of the discretized attribute values, and there is an edge between an object and an interval, if attribute value of the object falls in the interval. The encircled vertex sets in Fig. 6b is a quasi-biclique subgraph which corresponds to the noise tolerant subspace cluster shown in Fig. 6a.

There are several variations of quasi-biclique subgraphs, and their main difference is in their criteria on the missing edges, which can be characterized into two categories: (1) if there is restriction on the number of missing edges on each vertex, and (2) if the number of missing edges allowed is absolute or relative to the size of the quasi-biclique subgraph.

For the first category, restriction on the number of missing edges on each vertex in the quasi-biclique subgraph prevents skewed subgraphs to be mined. A subgraph is skewed when its distribution of missing edges is skewed, and vertices that have very low connectivities in the subgraph may be noise. (Sim et al. 2006, 2009b; Li et al. 2008) have this restriction, while (Mishra et al. 2005; Yan et al. 2005) do not have this restriction.

For the second category, (Sim et al. 2006, 2009b) allow an absolute number of missing edges in a quasi-biclique subgraph, while (Li et al. 2008; Mishra et al. 2005; Yan et al. 2005) allow a relative number of missing edges with respect to the size of the subgraph. Relative tolerance allows the number of missing edges to increase as the size of the cluster increases. Thus, relative tolerance is more natural than absolute tolerance, where the allowed number of missing edges is fixed regardless of the size of the cluster. However, efficient algorithms can be developed to mine subgraphs using absolute tolerance due to its anti-monotone property (Sim et al. 2006), which subgraphs using relative tolerance do not have.

The definition of a quasi-biclique which restricts the number of missing edges on each vertex in the subgraph, and has absolute number of missing edges in the subgraph, is presented as follows:

Definition 15 (Quasi-biclique) $C = (O, A)$ is a quasi-biclique if

- $h(C) := \begin{cases} \forall o \in O : |A| - |\{\{o, a\} | a \in A\}| \leq \epsilon \\ \forall a \in A : |O| - |\{\{o, a\} | o \in O\}| \leq \epsilon \end{cases}$
- $\pi(C) := |O| \geq min_o \wedge |A| \geq min_a$

Properties of the clusters:

- *Homogeneity* Most of the objects in a cluster have similar values in their subspace. The similarity is based on the discretization technique used on the data.
- *Size* The size of the cluster is determined by the parameters min_o and min_a.
- *Tolerate noisy data* By tolerating missing edges in the quasi-biclique subgraph, some objects that do not satisfy the clustering criteria are allowed to be in the subspace cluster. This is based on the assumption that the values of these objects fail to satisfy the clustering criteria due to noise.
- *Number of parameters* Three tuning parameters are needed, min_o, min_a for the size of the cluster and ϵ for the tolerance of the missing edges.
- *Parameter sensitivity* The cluster is sensitive to the tuning parameters. An exponential number of quasi-biclique subgraphs may be mined, if the tolerance of missing edges is extremely relaxed. Hence, the user has to set an appropriate ϵ with respect to the parameters min_o, min_a.

Quasi-biclique is extended to the time dimension in (Sim et al. 2011) and is known as cross-graph quasi-biclique. Thus, noise-tolerant 3D subspace clusters can be obtained

by mining cross-graph quasi-biclique subgraphs. $C = (O, A, T)$ is a cross-graph quasi-biclique subgraph if $\forall t \in T : (O_t, A_t)$ is a quasi-biclique and $|T| \geq min_t$.

Bayesian based approach The Bayesian based approach assumes the data is modeled by multivariate distributions, and uses them to infer if a value in the dataset belongs to any subspace cluster (Fu and Banerjee 2009). Hence, this approach is robust against noise, if the assumption holds.

The dataset is presented in the form of a matrix $\mathbb{D} = \mathbb{O} \times \mathbb{A}$, and the number of subspace clusters, k, is assumed to be known. Each row $o \in \mathbb{O}$ and column $a \in \mathbb{A}$ have k-dimensional latent bit vectors $\mathbf{z}^o$ and $\mathbf{z}^a$ respectively, which indicate their subspace cluster memberships. The subspace cluster membership for a value x_{oa} is obtained by an element-wise product of the corresponding row and column bit vectors, i.e. $\mathbf{z} = \mathbf{z}^o \odot \mathbf{z}^a$.

Definition 16 (Bayesian overlapping subspace cluster) $\forall i \in \{1, \ldots, k\} : C_i = (O_i, A_i)$ is a subspace cluster if

– $\forall o \in O_i, a \in A_i : \mathbf{z}_i = 1$, given that $\mathbf{z}_i$ is the i^{th} entry of the membership vector $\mathbf{z}$ and $\mathbf{z} = \mathbf{z}^o \odot \mathbf{z}^a$.

Bayesian overlapping subspace clusters are different from other subspace clusters, as homogeneity and support functions are not used to define the clusters.

The matrix $\mathbb{D}$ is assumed to be generated by $k + 1$ exponential families, each with parametric distribution $p(.|\theta_i), i \in \{1, \ldots, k + 1\}$. Each of the first k exponential family models a subspace cluster C_i, and the $(k + 1)^{th}$ exponential family models the noise in the matrix.

Each value $x_{oa} \in \mathbb{D}$ is assumed to be generated by some exponential families, which are picked based on the membership vector of the value, $\mathbf{z} = \mathbf{z}^o \odot \mathbf{z}^a$. For example, if $\mathbf{z_1} = \mathbf{z_3} = 1$, then x_{oa} is generated by the first and third exponential families. Thus, x_{oa} is assumed to be generated by a multiplicative mixture model as follows:

$$x_{oa} \sim \begin{cases} \frac{1}{c(\mathbf{z}^o \odot \mathbf{z}^a)} \prod_{i=1}^{k} p(x_{oa}|\theta_i, \mathbf{z}_i^o, \mathbf{z}_i^a) & \text{if } \mathbf{z}^o \odot \mathbf{z}^a \neq 0 \\ p(x_{oa}|\theta_{k+1}) & \text{otherwise} \end{cases} \tag{2}$$

where $c(.)$ is a normalization factor to guarantee that $p(.|\theta_i, \mathbf{z}_i^o, \mathbf{z}_i^a)$ is a valid distribution. If x_{oa} does not belong to any cluster membership, then it is assumed to be generated from the noise component $p(.|\theta_{k+1})$.

The latent bit vector $\mathbf{z}^o$ for each row and latent bit vector $\mathbf{z}^a$ for each column are obtained by the following generative process:

For each cluster C_i

1. For each row $o \in \mathbb{O}$
   (a) sample $\pi_i^o \sim Beta(\alpha_i^{\mathbb{O}}, \beta_i^{\mathbb{O}})$
   (b) sample $\mathbf{z}_i^o \sim Bernoulli(\pi_i^o)$
2. For each column $a \in \mathbb{A}$
   (a) sample $\pi_i^a \sim Beta(\alpha_i^{\mathbb{A}}, \beta_i^{\mathbb{A}})$
   (b) sample $\mathbf{z}_i^a \sim Bernoulli(\pi_i^a)$

Fu and Banerjee (2009) assume that the latent bit vectors are Beta-Bernoulli distributed. Thus, the problem of mining subspace clusters is transformed into a problem of estimating the parameters of the Beta-Bernoulli and exponential families. Using the matrix $\mathbb{D}$, the parameters of the Beta-Bernoulli distribution $\alpha_i^{\mathbb{O}}, \beta_i^{\mathbb{O}}, \alpha_i^{\mathbb{A}}, \beta_i^{\mathbb{A}}, \pi_i^o, \pi_i^a$ and exponential families θ_i are estimated using an EM-like algorithm.

Properties of the clusters:

- *Homogeneity and size* Both the homogeneity and size of the cluster is determined by the assumed distributions of the data.
- *Tolerate noisy data* By assuming that the data is generated by statistical distributions, we are able to infer if the values in the matrix $\mathbb{D}$ are noisy.
- *Number of parameters* There are parameters $\alpha_i^{\mathbb{O}}, \beta_i^{\mathbb{O}}, \alpha_i^{\mathbb{A}}, \beta_i^{\mathbb{A}}, \pi_i^o, \pi_i^a, \theta_i$, which describe the data distribution of each subspace cluster C_i.
- *Parameter sensitivity* Parameter sensitivity is not an issue, as the parameters of the model are estimated and not set by the user.

Probability based approach An uncertain object o_i is represented by a pdf p_{o_i}, which can be sampled by a set of vectors. For example, the uncertain object o_i can be a sensor, and the set of vectors is a set of readings, each taken at a different time interval.

Günnemann et al. (2010c) proposed a probability based approach to handle uncertain data. A subspace cluster is considered robust in uncertain data, if (1) its number of objects (support) exceeds a threshold, and (2) for each of its objects, the probability that the object is close to its medoid on the subspace is high.

Let vector $\mathbf{x} = (x_1, \ldots, x_{|\mathbb{A}|}) \in \mathbb{R}^{|\mathbb{A}|}$ be a point in the hyperspace defined by the set of attributes $\mathbb{A}$. Given a subset of attributes (subspace) A, the distance between two vectors $\mathbf{x}, \mathbf{x}'$ on subspace A is calculated in L^∞ norm, i.e. $d_\infty^A(\mathbf{x}, \mathbf{x}') = \max_{a \in A}\{|x_a - x'_a|\}$.

Let $p_{o_i}^A$ be the pdf of uncertain object o_i in subspace A, which is obtained by marginalizing over the attributes $\{|A|+1, \ldots, |\mathbb{A}|\}$.

$$p_{o_i}^A(\mathbf{x}) = p_{o_i}^A(x_1, \ldots, x_{|A|}) = \int_{x_{|A|+1}} \ldots \int_{x_{|\mathbb{A}|}} p_o(x_1, \ldots, x_{|\mathbb{A}|}) \tag{3}$$

On subspace A, the probability that the distance between uncertain object o_i and medoid m is less than ω is calculated as

$$P_{\leq\omega}(o_i, m, A) = \int_{\substack{\mathbf{x}, \mathbf{x}' \in \mathbb{R}^{|S|} \\ d_\infty^A(\mathbf{x}, \mathbf{x}') \leq \omega}} p_{o_i}^A(\mathbf{x}).p_m^A(\mathbf{x}')\, d\mathbf{x}\, d\mathbf{x}' \tag{4}$$

This probability is obtained by integrating over all possible pairs of vectors whose distances are less than ω on subspace A, and as o_i and m are assumed to be independent, the joint pdf of them is the product of both individual pdf.

Definition 17 (Subspace cluster for uncertain data) Given a medoid m, $C = (O, A)$ is a subspace cluster if

- $h(C) := \forall o \in O : P_{\leq w}(o, m, A) \geq \epsilon_{prob}$

– $\pi(C) := \sum_{\substack{o \in O \\ P_{\leq w}(o,m,A) \geq \epsilon_{prob}}} P_{\leq w}(o, m, A) \geq minSup$

Properties of the clusters:

– *Homogeneity* The homogeneity function is based on Eq. 4, which clusters objects that have high probability of being closed to the medoid m on subspace A.
– *Size* The support function of the cluster is the sum of the probabilities of the objects being closed to the medoid. Hence, a cluster with a large number of objects may not be valid if its sum of probabilities is low.
– *Account the uncertainty of the data* The uncertainty of the data is accounted when the pdf of the uncertain objects are used in the clustering, which is more accurate than using the deterministic values of the objects. However, in order to use the pdf, the distribution of the objects is assumed to be known.
– *Number of parameters* Three tuning parameters are required; the maximum distance allowed between vectors ω, the minimum probability of two uncertain objects being closed together ϵ_{prob}, and the minimum support of the cluster $minSup$, which is based on probability.
– *Parameter sensitivity* The cluster is sensitive to the tuning parameters.

5.2 Improving clustering results

5.2.1 Significant subspace clustering

There are two approaches to mine significant subspace clusters. The first approach is to mine significant subspaces, and then mine basic subspace clusters from them. Finding significant subspaces can be seen as a pre-processing step, and it is similar to the filter model of attribute selection (Dash et al. 2002), except that the filter model only finds one subspace based on the whole set of objects. Finding only one subspace contradicts the concept of subspace clustering, where objects can be homogeneous in different subspaces.

The second approach is to define significant subspace clusters and mine them directly. This is different from the first approach, where basic subspace clusters are mined, but they are mined from significant subspaces.

Entropy based subspaces Cheng et al. (1999) proposed mining basic subspace clusters from significant subspaces, to prevent an exponential number of clusters being generated.

Let A be the subspace in consideration. By treating each attribute $a \in A$ as a random variable, the entropy of A can be calculated as $H(a_1, \ldots, a_{|A|})$. Cheng et al. (1999) also measure the correlation of the attributes in A by the following equation:

$$interest(A) = \sum_{a \in A} H(a) - H(a_1, \ldots, a_{|A|}) \quad (5)$$

If the attributes are independent of each other, then the first term equates to the second term, resulting in $interest(A) = 0$. Hence correlated attributes will have high interest score.

Cheng et al. (1999) define a subspace A to be significant when the entropy of A is below ω and the interest of A is above ϵ.

A similar approach was proposed by Hsu and Chen (2004), but they only measure the entropy of each attribute and prune an attribute if its entropy is above a user-specified threshold.

Cheng et al. (1999) also proposed interesting subspaces. A subspace A is interesting when the entropy of A is below ω and the interest gain of A is above ϵ. Interest gain is defined as

$$interest\_gain(A) = interest(A) - \max_{a \in A}\{interest(A - \{a\})\} \tag{6}$$

In another words, a subspace is deemed interesting if additional attribute leads to a significant increase of $interest(A)$.

Property of the subspace:

- *Significant subspace* Cheng et al observed that a subspace with clusters typically has lower entropy than a subspace without clusters. The entropy of the subspace is the highest when the objects in the subspace are uniformly distributed. This is so, as under uniform distribution, the uncertainty of an object's location on the subspace is the highest, compared to other distributions. Whereas if the objects in the subspace are closely packed in a cluster, the entropy of the subspace is low as we are certain that an object would likely to be in the cluster.
- *Minimal subspaces* Contrary to other approaches, Cheng et al mine minimal subspaces instead of maximal subspaces, i.e. if A and A' are significant subspaces and $A \subset A'$, then only A is outputted. They argue that clusters from minimal subspace are easier to interpret and mining minimal subspaces is faster than mining maximal subspaces.
- *Number of parameters* There are two tuning parameters to set, ω and ϵ, which control the significance of the subspaces.
- *Parameter sensitivity* The cluster is sensitive to the tuning parameters.

Interesting subspaces Kailing et al. (2003) proposed a density based approach to find interesting subspaces, and basic subspace clusters are then mined from these subspaces. To measure the closeness of the objects on a subspace A, $\mathcal{N}_\epsilon^A(o)$ (described in Sect. 4.3) is used to calculate the ϵ-neighborhood of an object o on subspace A. We denote object o as a core-object of subspace A if $|\mathcal{N}_\epsilon^A(o)| \geq m$. We also denote $core[A]$ as the number of core-objects in subspace A, and $count[A]$ as the sum of all objects in the neighborhoods of all core-objects in subspace A.

A naïve way is to simply use $count[A]$ to determine the interestingness of the subspace A, but this will favor small subspaces, as large subspace has lower $count[A]$ due to $\mathcal{N}_\epsilon^A(o)$ being affected by the curse of dimensionality. Hence, Kailing et al proposed to normalize $count[A]$ by the hyperplane of the ϵ-neighborhood in subspace A, which is expressed as:

$$Quality(A) = \frac{count[A]}{|\mathbb{O}|^2.Vol_\epsilon^{|A|}} \tag{7}$$

The domain of each attribute $a \in A$ is normalized from [0, 1]. If L^∞-norm is used, then $Vol_\epsilon^{|A|}$ is a hypercube and can be computed as $Vol_\epsilon^{|A|} = (2\epsilon)^{|A|}$. If L^2-norm is used, then $Vol_\epsilon^{|A|}$ is a hypersphere and can be computed as $Vol_\epsilon^{|A|} = \frac{\sqrt{\pi}^{|A|}}{\Gamma(|A|/2+1)}.\epsilon^{|A|}$, where $\Gamma(x+1) = x \cdot \Gamma(x)$, $\Gamma(1) = 1$ and $\Gamma(\frac{1}{2}) = \sqrt{\pi}$.

All subspaces A that satisfy parameters ϵ and m will be outputted and ranked in descending order of $Quality(A)$. Its algorithm is discussed in Sect. 6.1.

Property of the subspace:

- *Significant subspace* The subspace is considered significant when the number of objects closed together is high with respect to the volume of the subspace.
- *Number of parameters* The tuning parameters required are used in the density based subspace clustering (Definition 8). There are two parameters to set; m controls the neighborhood density of an object on subspace A and ϵ controls the closeness of the objects on subspace A.
- *Parameter sensitivity* The subspace is sensitive to the tuning parameters.

High quality but minimal overlapping subspace clusters The *relevance* model (Müller et al. 2009a) and *orthogonal* model (Günnemann et al. 2009) have been proposed to mine high quality but minimal overlapping subspace clusters. The main difference between these two models is the way they reduce the overlapping clusters. In the relevance model, the subspace clusters have minimal overlapping in their sets of objects. In the orthogonal model, only subspace clusters whose sets of attributes are similar have minimal overlapping in their sets of objects.

Generally, subspace clusters are based on local information and they do not account the 'global' information of the dataset, i.e. $C = (O, A)$ is a subspace cluster because the set of objects O is homogeneous in the set of attributes A. Both relevance and orthogonal models take into account both the local and global information. In addition, both models are highly flexible as they can be used for any definitions of subspace clusters.

Relevance model Let us assume that we have fixed the type of subspace clusters to mine, and let $M = \{C_1, \ldots, C_n\} = \{(O_1, A_1), \ldots, (O_n, A_n)\} \subseteq ALL$ be a set of subspace clusters. We denote $Cov(M) = \cup_{i=1}^n O_i$ as the union of the objects in the set of clusters M. We also denote $k(C)$ as the cost function of cluster C, which measures the 'interestingness' of the cluster, and low cost $k(C)$ implies that the cluster is highly interesting. The cost function is subjected to the user's definition, which for example, can simply be the number of objects in the cluster or the density of the cluster.

The function *cluster gain* is proposed to measure the significance of a cluster $C = (O, A)$, which is defined as

$$clus\_gain(C, M) = \frac{|O \backslash Cov(M)|}{k(C)} \tag{8}$$

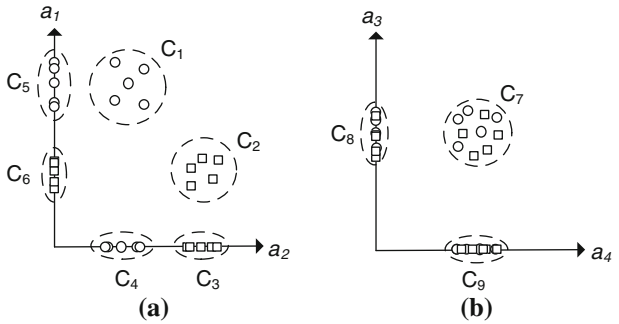


Fig. 7 Examples of objects under subspaces $\{a_1, a_2\}$ and $\{a_3, a_4\}$.C_1, C_2 are significant subspace clusters under the relevance model. C_1, C_2, C_7 are significant subspace clusters under the orthogonal model

A cluster with a high cluster gain is considered significant, if (1) it does not share its objects with other clusters (global property), and (2) its cost function is low, which means that the cluster is 'interesting', e.g. having high density (local property).

The relevance model is defined as follows:

Definition 18 (Relevance model) M^* is the set of significant subspace clusters if

- $\forall C \in M^* : clus\_gain(C, M^* \backslash \{C\}) > \Delta$.
- $\forall C' \notin M^* : clus\_gain(C', M^*) \leq \Delta$
- Overall relative cost of $M^* = \frac{\sum_{C \in M^*} k(C)}{|Cov(M^*)|}$ is minimal with respect to other sets of clusters $M \subseteq ALL$

Parameter Δ determines the significance of a cluster. The first criterion requires the clusters in M to be significant, and the second criterion requires clusters not in M to be insignificant. The last criterion requires that the best set of significant clusters is obtained. Figure 7a, b show objects in subspaces $\{a_1, a_2\}$ and $\{a_3, a_4\}$ respectively. $M^* = \{C_1, C_2\}$ is a set of significant subspace clusters under the relevance model, assuming that the cost function is dependent on the size of the subspace. Hence C_1 and C_2 have lower costs than the other clusters. Depending of the parameters setting, cluster C_7 may not be in M^* as its set of objects is highly overlapping with those of C_1 and C_2.

Orthogonal model Orthogonal subspace clustering aims to find a set of subspace clusters such that most of the objects and attributes are covered by the clusters, but there is not much overlapping of the clusters. Each subspace cluster represents a 'concept' (Günnemann et al. 2009), which is described by the attributes of the cluster. For example, the concept "taste of music" is described by attribute "number of rock concerts attended" and attribute "number of classic concerts attended". The subspace clusters are 'orthogonal' as their sets of objects and sets of attributes are almost different from each other.

Günnemann et al. (2009) first define *concept group*, which is a group of subspace clusters that have highly similar set of attributes (similar concepts) with respect to a subspace cluster $C = (O, A)$.

$$conceptGroup(C, M) = \{C_i \in M \backslash \{C\} || A_i \cap A| \geq \beta.|A|\} \tag{9}$$

β is a parameter that controls the degree of overlapping allowed in the set of attributes with cluster C. $\beta = 1$ means that a subspace cluster C_i has the same concept with C if the set of attributes A_i is equivalent to set of attributes A.

Next, *global interestingness* is used to ensure that clusters in a concept group do not have highly overlapping sets of objects.

$$I_{global}(C, M) = \frac{|O \backslash Cov(conceptGroup(C, M))|}{|O|} \tag{10}$$

A cluster is not removed if its set of objects has low overlap with the sets of objects of other clusters in the concept group, i.e. $I_{global}(C, M) \geq \alpha$, where α is the threshold controlling the strictness of the overlapping.

The set of subspace clusters M is defined as an *orthogonal clustering* if $\forall C \in M$: $I_{global}(C, M \backslash \{C\}) \geq \alpha$, i.e. the clusters in the concept group have low overlaps.

Günnemann et al further proposed a function I_{local}, which is the same as the cost function k of the relevance model, except that high I_{local} implies that the cluster is of high quality. The orthogonal model is defined as follows:

Definition 19 (Orthogonal model) M^* is the set of significant subspace clusters such that

$$M^* = \underset{M \in Ortho}{arg\,max} \{ \sum_{C \in M} I_{local}(C) \}$$
$$\text{with } Ortho = \{M \subseteq All | M \text{ is an orthogonal clustering}\}$$

In words, subspace clusters in the same concept group have minimal overlapping in their sets of objects and quality of each of subspace cluster is high.

In Fig. 7a, b, $M^* = \{C_1, C_2, C_7\}$ is a set of significant subspace clusters under the orthogonal model. Although the set of objects of C_7 is highly overlapping with those of C_1 and C_2, C_7 is of different concept group as C_1 and C_2.

Properties of the clusters of relevance and orthogonal model:

- *Homogeneity and size* The homogeneity and size of the clusters are dependent on the definition of the subspace clusters.
- *Significant subspace clusters* The subspace clusters are significant when (1) the quality of the clusters are high, and (2) there is minimal overlapping between the clusters for the relevance model, or there is minimal overlapping between clusters whose sets of attributes are similar for the orthogonal model.
- *Number of parameters* The relevance model has tuning parameter Δ which determines the significance of the cluster. The orthogonal model has tuning parameters β which controls the degree of overlapping in the attributes of the clusters and α which controls the degree of overlapping in the objects of the clusters. Depending on the definition of the subspace cluster used in the models, it is also possible to have parameters of the subspace clusters.
- *Parameter sensitivity* Both models are sensitive to their tuning parameters and it may be difficult to set them as they are non-meaningful and non-intuitive.

An extension of orthogonal subspace clustering, known as alternative subspace clustering (Günnemann et al. 2010b), is proposed. In alternative subspace clustering, a set of subspace clusters $Known$ is assumed to be known and the problem is to find orthogonal subspace clusters which are different from subspace clusters in $Known$. The similarities and differences between alternative clustering and subspace clustering are sketched in (Kriegel and Zimek 2010).

Statistical significant subspace clusters The significant subspace clusters presented so far are determined by the user, as the user has to set parameters and clusters which satisfy the parameters are considered significant. Hence, the clusters are sensitive to the parameters, and as mentioned earlier, setting the right parameters can be a difficult task for the user.

Moise and Sander proposed a fundamental shift from parameter-sensitive subspace clustering (Moise and Sander 2008). They proposed that a subspace cluster should be significant if it contains significantly more objects than expected under statistical principles. Thus, this approach is less sensitive to parameters.

Let $H = \Pi_{a \in A} D_O(a)$ be the hyper-rectangle formed by the subspace cluster $C = (O, A)$ and let $vol(H) = \Pi_{a \in A} r(D_O(a))$ be the volume of the hyper-rectangle H, and $vol(A) = \Pi_{a \in A} r(D(a))$ be the volume of the hyper-rectangle formed by the subspace A.

H is *statistically significant* if the number of objects O in hyper-rectangle H is significantly more than expected under uniform distribution. We have the null hypothesis that the number of objects O in hyper-rectangle H is Binomial distributed, i.e. $|O| \sim Binomial(|\mathbb{O}|, \frac{vol(H)}{vol(A)})$. The hyper-rectangle H is statistically significant if $|O| > \theta_\alpha$, where θ_α is the upper critical level of the hypothesis test, at significant level α.

Besides being statistically significant, Moise and Sander also require the subspace cluster not to be *induced* or *explained* by any other subspace clusters. Moise and Sander first made the following assumption:

Assumption 1 The data distribution is generated by the set of "true" subspace clusters M plus background noise.

Thus, a subspace cluster $C' = (O', A')$ is explained or induced by the set of subspace clusters M, if O' is consistent with Assumption 1. Therefore, we only need to mine the set of true subspace clusters M, as the other clusters can be explained by M.

To justify that subspace cluster C' is explained by M, we have to test if the number of objects in its hyper-rectangle H', is not significantly larger or smaller than what can be expected, under Assumption 1.

Given a subspace cluster $C = (O, A) \in M$, let $\pi_{H'}(C)$ be the hyper-rectangle of subspace cluster C' that is *explained* by the hyper-rectangle of subspace cluster C. We denote the volume of $\pi_{H'}(C)$ as $vol(\pi_{H'}(C)) = \Pi_{a \in A' \cap A} r(D_{O' \cap O}(a)).\Pi_{a \in A' \setminus A} r(D_{O'}(a))$. The number of objects in $\pi_{H'}(C)$ is assumed to follow a Binomial distribution $Binomial(n, \frac{vol(\pi_{H'}(C))}{vol(H)})$, where n is the estimated number of objects generated by the distribution of C. Using the same hypothesis test above in testing the significance of a hyper-rectangle, the upper and lower critical levels of the number of objects in $\pi_{H'}(C)$ are calculated. This test on subspace cluster C' is repeated using each cluster $C \in M$, and the upper and lower critical levels are summed up. If $|O'|$ falls within

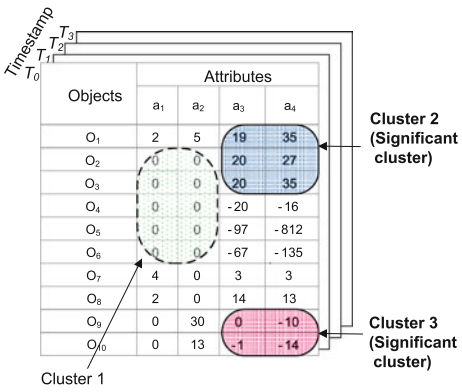


Fig. 8 Cluster 2 and 3 are correlated subspace clusters, as their values have high co-occurrences, and their co-occurrences in the cluster are not by chance (i.e., they only co-occur in their respective clusters)

the summed upper and summed lower critical levels, then we say C' is explained by the set of "true" subspace clusters M.

Definition 20 (Statistical significant subspace clusters) M is the set of statistical significant subspace clusters under uniform distribution if

- $\forall C \in M : \pi(C) := \begin{cases} 1.\ H \text{ is statistically significant} \\ 2.\ \forall a \in A : \text{objects in } O \text{ are not uniformly distributed in } D(a) \end{cases}$
- 3. M has the smallest cardinality $|M|$ in ALL, such that any subspace cluster in $ALL \backslash M$ is *explained (induced)* by at least one of the subspace clusters in M.

Criterion 2 can be easily checked by using Kolmogorov-Smirnov goodness of fit test for uniform distribution (Snedecor and Cochran 1989).

Properties of the clusters:

- *Homogeneity* Homogeneity function is not defined, but the homogeneity of the clusters are considered during the mining of the clusters. In its algorithm STATPC, each object is taken as a subspace cluster and other objects are greedily added to it based on their distance.
- *Size* The size of the cluster is defined by criteria 1 and 2 of Definition 20.
- *Significant subspace clusters* Under Assumption 1, the significant subspace clusters are the "true" subspace clusters M.
- *Number of parameters* There are two significance levels to set: the significant level on the hyper-rectangle H of the cluster and the significant level used in the Kolmogorov-Smirnov goodness of fit test in criterion 2 of Definition 20.
- *Parameter sensitivity* The sensitivity of the tuning parameters are tested and a default parameters setting is chosen (Moise and Sander 2008), which is less sensitive to the results. However, this approach has the assumption that the data follows uniform distribution, which is not always the case in real-world data.

Correlated subspace clusters Sim et al. (2009a, 2010a) proposed using mutual information to mine significant subspace clusters, known as correlated subspace clusters. In a correlated subspace cluster, the values have high co-occurrences in the dataset, and their co-occurrences in the cluster are not by chance. The latter condition is

the same as statistical significant subspace cluster's condition that the cluster is not explained by other clusters. Figure 8 shows an example of a 3D dataset, with three subspace clusters. The values in cluster 1 has high co-occurrences, but they also occur in other objects ($x_{o_9a_2} = x_{o_{10}a_1} = 0$, $x_{o_7a_2} = x_{o_8a_2} = 0$), hence cluster 1 is not significant. Cluster 2 and 3 are significant clusters, as their values have high co-occurrences, and their co-occurrences in the cluster are not by chance (i.e., they only co-occur in the clusters).

A metric known as *correlation information* is used to measure how correlated the subspace cluster is. High correlation information means that the cluster is correlated, and hence it is significant. Let us assume that a subspace cluster C contains two values x, y, and let the probability of x occurring in the dataset be denoted as $p(x)$. The correlation information is defined as $ci(C) = p(x, y)log\frac{p(x,y)}{p(x)p(y)}$. Intuitively, the first term measures the co-occurrences of the values, and the second term measures if the co-occurrences are by chance.

Correlated subspace clusters can be mined from either 2D or 3D dataset, and Sim et al explained them in the context of 3D dataset. The clusters in 2D context can be easily understood by assuming that the dataset only contains a single timestamp.

In 3D dataset, Sim et al denote sub-cuboid $\mathbb{O} \times A \times T$ as the domain of a set of attributes A at a set of timestamps T, represented as $D(A, T)$. They define a slice of the sub-cuboid as $S = \{o\} \times A \times T \in D(A, T)$. They denote $D_O(A, T) = O \times A \times T$ as the domain of the set of attributes A at a set of timestamps T, projected on the set of objects O.

Given that there is a sub-cuboid $C = (O, A, T)$ which is a correlated subspace cluster, and there is a slice $S_t = \{o\} \times A \times \{t\} \in D_O(A, \{t\})$, Sim et al convert slice S_t to a vector $\mathbf{v}_t$ by the following function:

Definition 21 *(Mapping of slice $S_t = \{o\} \times A \times \{t\}$ to column vector $\mathbf{v}_t$)* Let slice S_t be represented as a partially ordered set $\{x_{oat} | a \in A\}$ with cardinality d, and let $\mathbf{v} = (v_1, \ldots, v_d)^{\mathrm{T}}$ be a column vector of d values. S_t is mapped to $\mathbf{v}_t$ using function $\beta : S_t \rightarrow \mathbf{v}_t = x_{oat} \mapsto v_i (1 \leq i \leq d)$.

Assume that there are three vectors $\mathbf{v}_1 = (u_1, \ldots u_l)$, $\mathbf{v}_2 = (v_1, \ldots v_m)$, $\mathbf{v}_3 = (w_1, \ldots w_n)$. For brevity, a sequence of values $v_1, \ldots, v_m$ is represented as $v_{1\ldots m}$. The correlation information of these three vectors is given as follows:

$$\tilde{ci}(\mathbf{v_2}, \mathbf{v_3} | \mathbf{v_1}) = \sum_{i=1}^{m} \sum_{j=1}^{n} p(v_{1\ldots i}, w_{1\ldots j}, u_{1\ldots l}) \log \frac{p(v_{1\ldots i}, w_{1\ldots j}, u_{1\ldots l})}{p(v_{1\ldots i}, w_{1\ldots j-1}, u_{1\ldots l}) p(v_{1\ldots i-1}, w_{1\ldots j}, u_{1\ldots l})} \tag{11}$$

where $p(.)$ is the probability of the values occurring in the dataset, and kernel density estimation (Silverman 1986) is used in calculating the probability of the continuous values.

Definition 22 (Correlated subspace cluster) A sub-cuboid $C = (O, A, T)$ is a correlated subspace cluster if

$$\pi(C) := \tilde{ci}(C) = \sum_{i \in T} \sum_{\mathbf{v_1} \in D_O(A,\{1\}) \ldots \mathbf{v_i} \in D_O(A,\{i\})} \tilde{ci}(\mathbf{v_i}, \mathbf{v_{i-1}} | \mathbf{v_{1,\ldots,i-2}}) \quad (12)$$

is high

Intuitively, a sub-cuboid C is a correlated subspace cluster if (1) for each time frame $O \times A \times \{t\}$ of C, its values are correlated and (2) for each pair of contiguous time frames $O \times A \times \{t\}$ and $O \times A \times \{t+1\}$ of C, they are correlated, given prior time frames. Determining how high $\tilde{ci}(C)$ is considered to be significant is explained in the correlated subspace clustering algorithm MIC in Sect. 6.3.

Properties of the clusters:

- *Homogeneity* The homogeneous function is not used in correlated subspace cluster, but the homogeneity of the values in the cluster is considered when the kernel density estimation is used in calculating the probability of the values, as values that are closed together have high probabilities.
- *Size* There is no size requirement.
- *Significant subspace cluster* The significance of the cluster is based on how correlated its values are, which is measured using correlation information. Values are correlated when they have high-occurrences, and their co-occurrences are not by chance.
- *Number of parameters* No parameters are required to define the cluster, but the algorithm MIC requires setting of a tuning parameter, which determines the number of seeds (pairs of values) used in building the clusters.
- *Sensitivity to parameters* Not applicable.

5.2.2 Semi-supervised subspace clustering

Constraint based subspace clustering In constraint based subspace clustering, the definition of the subspace clusters is dependent of the user. For example, density based or window based subspace clusters can be used. Constraints, which we can consider as additional criteria, are then 'add-on' to the clusters.

Similar to traditional constraint based clustering (Wagstaff et al. 2001), object-level constraints are incorporated in constraint based subspace clustering (Fromont et al. 2009). There are two types of object-level constraints: must-link and cannot-link. Must-link indicates that a pair of objects must be in the same cluster, while cannot-link indicates that a pair of objects cannot be in the same cluster.

Properties of the clusters:

- *Homogeneity, Size, Sensitivity to parameters, Number of parameters* These properties are dependent on the definition of the subspace cluster. In (Fromont et al. 2009), grid based and window based subspace clusters are used.
- *Semi-supervised* The cluster satisfies the must-link and cannot-link constraints.

Actionable subspace clustering The concept of actionable is derived from actionable patterns (Kleinberg et al. 1998), which are patterns that have the ability to suggest profitable action for the decision-makers. Sim et al. (2010b) proposed actionable subspace clusters, which are particularly useful in financial data mining. For example,

investors can generate profit by buying stocks from an actionable subspace cluster defined by a set of stocks (objects) and a set of financial ratios (attributes).

A continuous attribute known as *utility* is proposed to measure the actionability of the cluster; the higher the utility of the objects, the higher is the actionability of the cluster. Utility is similar to the object-level constraints, except that utility is continuous while object-level constraints are binary indicators.

Actionable subspace clusters are mined from a 3D dataset $\mathbb{D} = \mathbb{O} \times \mathbb{A} \times \mathbb{T}$, and are defined as follows:

Definition 23 (Actionable subspace cluster) $C = (O, A)$ is a subspace cluster if $h(C) :=$

- $\forall t \in \mathbb{T}$: the objects in O are similar on attributes in A
- the objects in O have high and correlated utility

Thresholds are not used to explicitly define the goodness of the objects' similarity and correlation. Instead, the goodness of the objects are expressed in weights, and an objective function is used to calculate the weights with respect to a centroid. By maximizing this objective function, objects which have high weights are clustered with the centroid. Its algorithm MASC (described in Sect. 6.4) adaptively determines a threshold on the weights, and objects whose weights are above the threshold are used in the clustering.

Given a centroid c, let p_{oa} be the weight indicating if the object o should be part of a cluster containing c, on attribute a. Each attribute a has a set of weights $P = \{p_{o_1 a}, \ldots, p_{o_{|\mathbb{O}|} a}\}$ and each set of weights P is calculated by optimizing the objective function

$$f(P) = f^{util}(P) \cdot f^{corr}(P) \tag{13}$$

$f^{util}(P)$ measures the utility of each object and its similarity to the centroid, i.e.

$$f^{util}(P) = \sum_{o \in \mathbb{O}} p_{oa}\, s(c, o)\, util(o) \rho(c, o) \tag{14}$$

$s(c, o)$ is the similarity metric between c and o in Euclidean space, $util(o)$ is the utility of o and $\rho(c, o)$ measures the linear correlation of the utility between c and o. An object will have a high weight, if (1) it is highly similar to centroid c on attribute a, (2) its utility is high and correlated to the utility of c.

$f^{corr}(P)$ measures the linear correlation between objects, and their similarity to the centroid, i.e.

$$f^{corr}(P) = \sum_{o,o' \in \mathbb{O} \times \mathbb{O} | o \neq o'} p_{oa}\; p_{o'a}\; s(c, o)\; s(c, o')\; \rho(o, o'), \tag{15}$$

This function ensures that higher weightages are given to objects that have correlated utilities and that are similar to centroid c.

Properties of the clusters:

- *Homogeneity* The homogeneity is dependent on the weights of the objects, and is measured using Euclidean distance. There is a requirement of the clusters to be persistent in every timestamps, which may be too stringent. Hence actionable subspace clusters have the same weakness of coherent gene clusters (Definition 11).
- *Size* Similar to homogeneity, the size of the cluster is dependent on the weights of the objects.
- *Semi-supervised* The objects in the cluster are required to have high and correlated utilities.
- *Number of parameters* The user has to select the centroids, based on the domain knowledge of the user. This selection of centroids can be considered as a parameter.
- *Sensitivity to parameters* No parameters are required to explicitly define the goodness of the cluster.

There are projected clusterings which also use optimization based approach (Chan et al. 2004; Domeniconi et al. 2004; Jing et al. 2007). Similar to actionable subspace clustering, their objective functions are formulated such that the distances between the objects and the centroid of a cluster are minimized. However, they are k-means algorithms which optimize their objective function in a global sense, i.e. they aim to obtain an optimal partition of the objects. Actionable subspace clustering optimizes its objective function in a local sense, i.e. a set of objects are optimally clustered with respect to a centroid.

In these projected clusterings, the objects are clustered based on the whole set of attributes, and not on subspaces. Hence, we do not know in which subspaces the objects are homogeneous in. These projected clusterings may also be sensitive to outliers, as each object is required to be in a cluster, and this problem is aggravated by the need to select an appropriate number of clusters.

Twofold clustering In some applications, data in the form of 2D dataset $\mathbb{D} = \mathbb{O} \times \mathbb{A}$, and a graph G (with the objects as the vertices, i.e. $V(G) = \mathbb{O}$), are available. Günnemann et al. (2010a) proposed simultaneous clustering on both matrix and graph, to produce more meaningful and accurate clusters, denoted as *twofold clusters*. More specifically, their aim is to mine sets of objects which are homogeneous in their subspaces of the matrix, and at the same time, are densely connected in the graph. For example in target and viral marketing, it is useful to find groups of people which have similar attributes, and know (connected to) each other. Figure 9 shows an example, where each node represents a person, and two persons are connected if they know each other. Person 2-5 are densely connected, and at the same time, their attributes (age and interest) are similar. Hence, the set of objects (person) {2, 3, 4, 5} and the set of attributes $\{age, interest\}$ form a subspace cluster.

This synthesis of the two paradigms can be seen as a form of semi-supervised subspace clustering, as information from the graph data is used to guide and improve the subspace clustering process.

Let $deg^O(o)$ be the degree of vertex(object) o within the set of vertices(objects) O, i.e. $deg^O(o) = |\{o' \in O | (o, o') \in E(G)\}|$. We measure the density of a subgraph formed by a set of vertices (objects) O by $\gamma(O) = \frac{min_{o \in O}\{deg^O(o)\}}{|O|-1}$. The higher the density, the more connected are the vertices of the subgraph.

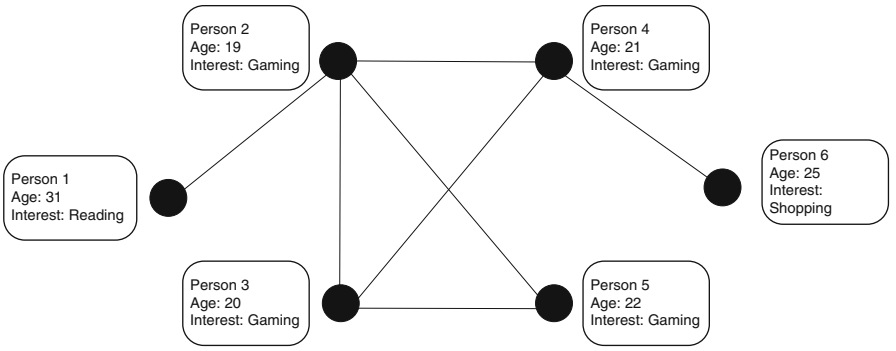


Fig. 9 A synergy of graph and 2D dataset $\mathbb{D} = \mathbb{O} \times \mathbb{A}$. Each node represents a person, and the attributes of the person are shown. The set of objects (person) $\{2, 3, 4, 5\}$ and the set of attributes $\{age, interest\}$ form a subspace cluster

Definition 24 (Twofold cluster) $C = (O, A)$ is a twofold cluster if

- $h(C) := \begin{cases} 1. \begin{cases} \forall a \in A : \forall o, o' \in O : |x_{oa} - x_{o'a}| \leq w \\ \forall a \in \mathbb{A}/A : \forall o, o' \in O : |x_{oa} - x_{o'a}| > w \end{cases} \\ 2. \ \gamma(O) \geq \gamma_{min} \end{cases}$
- $\pi(C) := |O| \geq min_o \wedge |A| \geq min_a$

In the homogeneity function $h(C)$, the first criterion focuses on the subspace cluster C, while the second criterion focuses on its corresponding dense subgraph.

Like the other subspace clusters, it is possible that an exponential number of twofold clusters can be generated, depending on the parameters setting. Hence, Günnemann et al. (2010a) proposed mining a set of optimal twofold clusters, and remove twofold clusters that are redundant. A twofold cluster C' is *redundant* if it is highly similar to another twofold cluster C, and its quality is lower than the quality of C. The quality of a twofold cluster $C = (O, A)$ is defined as $Q(C) = \gamma(O)^a.|O|^b.|A|^c$, where parameters a, b, c control the weightages of the cluster's characteristics in contribution to the quality of the cluster.

Formally, a twofold cluster $C' = (O', A')$ is redundant if there is another twofold cluster $C = (O, A)$, such that (1) $Q(C') < Q(C)$, (2) $\frac{|O' \cap O|}{|O'|} \geq r_o$, and (3) $\frac{|A' \cap A|}{|A'|} \geq r_a$. We denote this redundancy relationship as $C' \prec_{red} C$, and parameters r_o, r_a determine the thresholds on the clusters' overlapping. This redundancy relationship is used in obtaining the optimal twofold clustering, which is defined as follows:

Definition 25 (Optimal twofold clustering)
M is the set of optimal twofold clusters such that

- $\neg \exists C_i, C_j \in M : C_i \prec_{red} C_j$
- $\forall C_i \in ALL \backslash M : \exists C_j \in M : C_i \prec_{red} C_j$

Properties of the clusters:

- *Homogeneity* The homogeneity is dependent on the two main criteria of the homogeneity function shown in Definition 24. The first criterion requires the cluster to

be homogeneous in its subspace[2] and not homogeneous in other attributes, with parameter w controlling the distance allowed between the objects.
The second criterion requires the objects in the cluster to be highly connected to each other, with parameter γ_{min} determining the minimum connectivity of the objects in the cluster.

- *Size* The size of the cluster is determined by the parameters min_o and min_a.
- *Significant subspace cluster* The optimal twofold clustering is similar to the concept of the relevance (Definition 18) and orthogonal (Definition 19) models, where the optimal twofold clusters are high in quality and have low overlapping among each other.
- *Semi-supervised* The objects in the cluster are required to be highly connected.
- *Number of parameters* There are a total of nine tuning parameters to set, $w, \gamma_{min}, min_o, min_a, a, b, c, r_o, r_a$.
- *Sensitivity to parameters* The cluster is sensitive to the tuning parameters.

5.3 Summary

Table 3 presents the desired properties that the subspace clustering approaches have. Compared to the basic subspace clustering approaches, the enhanced subspace clustering approaches have more of the desired properties.

6 Subspace clustering: algorithms

We categorize the subspace clustering algorithms into four main families, namely lattice based algorithm, statistical model, approximation algorithm and hybrid algorithm.

6.1 Lattice based algorithm

Traversal on the lattice (also known as the Hasse diagram) is the most common method used in subspace clustering, which is also a commonly used method in frequent itemset mining (Agrawal and Srikant 1994) and graph mining (Tomita et al. 2004).

Let us assume that we have a set of candidates (or building blocks) of subspace clusters. The candidates can be the units of grid based subspace cluster (Definition 6), the windows of the window based subspace cluster (Definition 7), attribute values, or attributes, etc.

The powerset of the candidates is modeled as a lattice, with each node of the lattice representing a set of candidates. Potential subspace clusters are mined from each node. Figure 10a shows an example of a lattice with candidates $\{a, b, c, d\}$. In some literatures (Kriegel et al. 2009), the lattice is inverted and traversal from the root is known as bottom up traversal.

[2] This is similar to the homogeneity function of window based subspace cluster (Definition 7).

Table 3 A summary of the desired properties of subspace clusters

Subspace cluster	Significant size	Maximal	Handles 3D	Subspace in the third dimension	Handles categorical data	Tolerate noisy data	Handles uncertain data	Significant clusters	Semi-supervised clusters	Number of parameters	Parameter-insensitive
Grid based subspace cluster	√	√								2	
Density based subspace cluster	√	√								2	
Window based subspace cluster	√	√								3	
Binary 3D subspace cluster	√	√	√	√						3	
Dense 3D subspace cluster		√	√	√		√				1	
Coherent gene cluster	√	√	√							3	
Tricluster	√	√	√	√						7	
Quasi-biclique	√	√	√	√		√				3	
Categorical subspace cluster	√	√			√					1	
Bayesian overlapping subspace cluster		√				√				7	√
Subspace cluster for uncertain data	√	√					√			3	
Subspace α-cluster	√	√								3	
Entropy based subspace								√		2	
Interesting subspace		√						√		2	
Relevance model	–	–						√		1	
Orthogonal model	–	–						√		2	

Table 3 continued

Subspace cluster	Significant size	Maximal	Handles 3D	Subspace in the third dimension	Handles categorical data	Tolerate noisy data	Handles uncertain data	Significant clusters	Semi-supervised clusters	Number of parameters	Parameter-insensitive
Statistical significant subspace cluster								√		2	√
Correlated subspace cluster		√	√	√				√		0	√
Constraint based subspace cluster	√	√							√	–	
Actionable subspace cluster		√	√						√	1	√
Twofold cluster	√	√						√	√	9	

A '√' indicates the cluster has the desired property. A '–' indicates the property is dependent on the type of subspace cluster used and not dependent on the model. Homogeneity is one of the desired cluster, but is not shown since all subspace clusters have this property

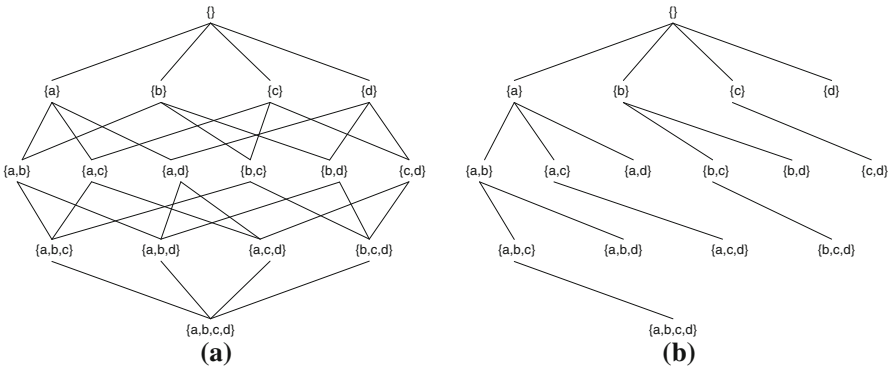


Fig. 10 **a** The lattice (also known as the Hasse diagram) of candidates $\{a, b, c, d\}$. **b** The set enumeration tree of candidates $\{a, b, c, d\}$

Let *cand* be the set of candidates of a node, and let us assume that a clustering algorithm has a function $f(cand)$, which generates one or more subspace clusters based on *cand*, i.e. $f(cand) = \{C_1, \ldots, C_m\}$. An efficient algorithm will ensure that subspace clusters generated from a node are unique from the others, to avoid generation of duplicated clusters.

The worst-case time complexity of the traversal of the search space is $O(2^n)$, assuming that we have n candidates. Thus, efficient traversal and pruning of the search space are the two main concerns of clustering algorithms using this method.

There are two ways to traverse the lattice, breadth-first and depth-first. In breath-first traversal, all nodes at a level of the lattice are traversed before moving down to the children nodes in the next level. For example in Fig. 10a, node $\{a, b, c\}$ is traversed after its parent nodes $\{a, b\}$, $\{a, c\}$, $\{b, c\}$ are traversed. For depth-first traversal, the lattice can be arranged as a set enumeration tree (Rymon 1992), where each child is traversed only by one of its parents, to prevent duplicate traversals of nodes. Figure 10b shows an example of a set enumeration tree. In depth-first traversal, the nodes are recursively traversed downwards.

The choice of traversal depends on the properties of the subspace clusters and memory overhead. In terms of the properties of the subspace clusters, it is possible that exploitation of them may lead to more efficient traversal in either breadth-first or depth-first.

In terms of memory overhead, depth-first traversal will be the better choice as it uses less memory. Depth-first traversal only needs to consider a recursive path of nodes during traversal, while breath-first traversal needs to consider the nodes per level, which the number of nodes can be in exponential. This usually happens in the middle levels of the lattice, as the lattice is typically diamond-shaped.

6.1.1 Subspace cluster with anti-monotone property

To allow efficient pruning of the search space, some subspace clusters are defined in such a way that their properties can be exploited to prune the search space. The most common pruning property is the anti-monotonicity of the subspace cluster.

Table 4 A summary of subspace clusters that are mined using the lattice traversal method

Cluster/subspace	Algorithm	Traversal	Monotonicity
Grid based subspace cluster (Definition 6)	CLIQUE (Agrawal et al. 1998)	Breadth-first	√
Density based subspace cluster (Definition 8)	SUBCLU (Kailing et al. 2004)	Breadth-first	√
Window based subspace cluster (Definition 7)	MaxnCluster (Liu et al. 2009)	Depth-first	√
Binary 3D subspace cluster (Definition 9)	Data Peeler (Cerf et al. 2008, 2009)	Depth-first	√
	CubeMiner (Ji et al. 2006)	Depth-first	√
	TRIAS (Jaschke et al. 2006)	Breadth-first or depth-first	√
Dense 3D subspace cluster (Definition 10)	DCE (Georgii et al. 2010)	Reverse	
Coherent gene cluster (Definition 11)	Gene-sample search and Sample-gene search (Jiang et al. 2004a)	Depth-first	√
Triclusters (Definition 12)	TRICUSTER (Zhao and Zaki 2005)	Depth-first	√
Categorical subspace clusters (Definition 13)	CLICKS (Zaki et al. 2005)	Depth-first	
Subspace α-cluster (Definition 14)	CI, CM-UPALL, CM-UPONE (Kontaki et al. 2008)	Breadth-first	√
Quasi-biclique (Definition 15)	MQBminer (Sim et al. 2009b)	Depth-first	√
CGQB (Sect. 2.3.1)	CGQBminer (Sim et al. 2011)	Depth-first	√
Entropy based subspace (Sect. 5.2.1)	ENCLUS (Cheng et al. 1999)	Breadth-first	√
Interesting subspace (Sect. 5.2.1)	RIS (Kailing et al. 2003)	Breadth-first	√
Constraint based subspace cluster (Sect. 5.2.2)	SC-MINER (Fromont et al. 2009)	Depth-first	√
Twofold cluster (Definition 24)	GAMER (Günnemann et al. 2010a)	Depth-first	

Definition 26 (Anti-monotonicity) If submatrix $O \times A$ forms a subspace cluster, then any submatrix $O' \times A'$ that is a subset of $O \times A$, i.e. $O' \subseteq O$, $A' \subseteq A$, also forms a subspace cluster.

Hence, if a submatrix does not form a subspace cluster, then its superset also does not form a subspace cluster. The anti-monotonicity of the subspace cluster is a simple yet efficient way of pruning the search space; if a node of the lattice does not have a subspace cluster, then there is no need to traverse to its children because its children too will not have subspace clusters.

In subspace clustering algorithms, it is quite common to use lattice based algorithm with anti-monotonicity as the pruning measure. We present the algorithms and their characteristics in Table 4. In the following, we discuss variations of the standard lattice based algorithm.

Coherent gene clustering (Definition 11) Two lattice based algorithms are proposed (Jiang et al. 2004a) to mine coherent gene clusters. The Gene-Sample search algorithm creates a set enumeration tree of genes, with each node representing a set of genes. Depth-first traversal is performed on the tree, and when a node is traversed, its set of genes is used to obtained its corresponding set of samples, which together form a coherent gene cluster. Similarly, the Sample-Gene search algorithm creates a set enumeration tree of samples and depth-first traversal is performed to mine the clusters.

Similarly, the Sample-Gene search algorithm creates a set enumeration tree of samples and depth-first traversal is performed to mine the clusters. The choice of the algorithm is dependent of the dataset. In microarray dataset, the number of genes is typically larger than the number of samples. Thus the Sample-Gene search is faster than the Gene-Sample search, as its set enumeration tree is smaller.

Subspace α-clustering (Definition 14) To allow efficient clustering in stream data, subspace α-clusters (Kontaki et al. 2008) are mined in a sliding window of size w on the streaming attributes, instead of the whole stream data. Hence, the clustering is localized to a matrix $\mathbb{O} \times \{a_i, \ldots, a_{i+w}\}$. When a right-most attribute a_{i+1+w} is streamed in, the matrix is shifted and becomes $\mathbb{O} \times \{a_{i+1}, \ldots, a_{i+1+w}\}$.

Three lattice based algorithms are developed to mine subspace α-clusters, with the nodes of the lattice representing a set of objects and a set of attributes that can form the clusters. The first algorithm CI, is used to mine the initial set of subspace α-clusters based on the first sliding window of the streaming data. It traverses the lattice in breadth-first search to mine the clusters, and pruning of the search space is based on the anti-monotonicity of the subspace α-clusters.

The second algorithm CM-UPALL, is used when the sliding window is moved one position to the right on the attributes, i.e., the data is changed from matrix $\mathbb{O} \times \{a_i, \ldots, a_{i+w}\}$ to matrix $\mathbb{O} \times \{a_{i+1}, \ldots, a_{i+1+w}\}$. CM-UPALL first checks the existing clusters with respect to the new matrix. Clusters that fail the cluster criteria with respect to the new matrix are removed. And some clusters are expanded due to the new matrix. Next, CM-UPALL mines new clusters from the new matrix, based on the same algorithm as CI. The third algorithm CM-UPONE, is used when the sliding window is moved one position to the right on the attributes, but only the data of an object o is involved. CM-UPONE is similar to CM-UPALL, except that it only updates clusters that involved object o.

Mining interesting subspaces (Sect. 5.2.1) To mine interesting subspaces, Kailing et al. (2003) proposed taking each object as a centroid. For each centroid, a lattice is constructed and used to find interesting subspaces with respect to the centroid. The nodes of the lattice are sets of attributes. The traversal is in breadth-first and there is anti-monotonicity on the interesting subspaces for efficient pruning of the lattices.

Constraint based subspace clustering (Sect. 5.2.2) Fromont et al. (2009) proposed an algorithm that allows incorporation of object-level constraints into grid based and window based subspace clusterings. The algorithm SC-MINER uses the set enumeration tree method to mine the clusters, and it pushes the cannot-link and must-link constraints into the mining for efficient traversal of the tree. Let D be a set of bins,

which are the grids or windows of the dataset. The choice of using grids or windows as bins depends on which type of subspace clusters are to be mined.

SC-MINER traverses the tree in depth-first search, and recursively enumerates all constraint based subspace clusters (O, A). Each node of the tree contains a triplet $< (O, D), (O', D'), (O_N, D_N) >$, where (O, D) are the members of the subspace cluster currently enumerated, (O', D') are objects and bins yet to be enumerated, and (O_N, D_N) are objects and bins that have been enumerated as elements that do not belong to any subspace clusters currently being mined.

During the traversal from a node, an element (object or bin) e is enumerated and put in (O, D), and all elements of (O', D') that are not related to e are removed, thus the search space is reduced considerably. For example, objects that have the cannot-link constraint with e are removed from (O', D'), objects o' that have the must-link constraint with e are put into (O, D), or objects that have the cannot-link constraint with o' are removed from (O', D').

Graph based clustering In some algorithms, the data is first pre-processed and information required to mine the clusters is represented in the form of graphs. Specialized subgraphs such as cliques or bicliques are then mined from the graphs, and these specialized subgraphs either correspond to the clusters, or post-processing is done on these specialized subgraphs to get the clusters.

In these algorithms, the mining of the subgraphs is done using the lattice, with the nodes of the lattice representing sets of vertices of the graph.

Mining quasi-bicliques (Definition 15) Quasi-bicliques are mined from the set enumeration tree of the vertices of the graph (dataset) in depth-first traversal (Sim et al. 2009b). The anti-monotone property of quasi-bicliques is used to prune the tree.

Mining triclusters (Definition 12) Mining of triclusters consists of two stages (Zhao and Zaki 2005). In the first stage, for each timestamp t of the dataset, a directed weighted multigraph G_t is constructed, with the vertices representing the attributes. Note that a multigraph allows multiple edges between a pair of vertices. A directed edge from one vertex (attribute) a to another vertex (attribute) a' is associated with a set of objects that is related on the attributes a and a'.

Maximal clique subgraphs are mined from each multigraph G_t, with each maximal clique subgraph representing a set of objects (obtained from the edges of the clique) related to a set of attributes (obtained from the set of vertices of the clique) in timestamp t. Thus, maximal clique subgraphs can be considered as biclusters which will be used to form triclusters. The maximal clique subgraphs are systematically mined by traversing a set enumeration tree in depth-first order, with each node of the tree corresponding to a set of vertices (attributes) of the multigraph.

In the second stage, a final weighted multigraph is constructed, with the vertices representing the timestamps. An edge between vertex (timestamp) t and vertex (timestamp) t' is associated with a pair of highly overlapping biclusters from timestamps t and t'. Similar to the previous stage, maximal clique subgraphs are mined from this multigraph, and each maximal clique subgraph corresponds to a tricluster.

The two-stage mining of TRICLUSTER may have efficiency issues, as during the first stage mining, the time information is not used to prune the biclusters. Thus, it is possible that a large number of biclusters are mined on each multigraph in the first stage, but only a small number of them are part of triclusters.

6.1.2 Subspace cluster without anti-monotone property

There are certain subspace clusters that do not have the anti-monotonicity property in their definition, such as the dense 3D subspace cluster (Definition 10) and quasi-biclique with relative noise tolerance (Sect. 5.1.4).

Categorical subspace clustering (Definition 13) CLICKS (Zaki et al. 2005) is a graph based algorithm that uses the lattice to mine categorical subspace clusters. A k-partite graph is first created, where the vertices represent attribute values of the data. Two vertices representing attribute values x_a, $x_{a'}$ have an edge if they are 'dense' together, i.e. their occurrence together is more than expected (cf. Definition 13 second criterion).

The k-partite graph is converted to a set enumeration tree, with the nodes of the tree representing the sets of vertices (attribute values). Depth-first traversal is then performed on the tree to mine k-partite clique subgraphs, which are the categorical subspace clusters. In a k-partite clique subgraph, there are k vertices (attribute values), and they are connected to each other. In addition, each of these k vertices belongs to different attributes.

Due to the non-monotonic definition of categorical subspace cluster, it is possible that a k-partite clique subgraph is not a subspace cluster, but its subgraphs are. Hence, there is post-processing to check if a k-partite clique subgraph is a valid categorical subspace cluster. If the k-partite clique subgraph is an invalid cluster, its subgraphs are checked for subspace clusters.

Dense 3D subspace clustering (Definition 10) For dense 3D subspace clustering (Georgii et al. 2010), anti-monotonicity is induced by using the reverse search paradigm (Avis and Fukuda 1996) in its mining process.

The reverse search paradigm arranges the lattice into a *reverse search tree*, with each node representing a dense 3D subspace cluster $C = (O, A, T)$, and each node has only one parent, to avoid duplicate traversals of nodes. In the tree, the density of the cluster in a node is at least as large as the maximum density among its children. Hence, if the density of the cluster of the node is less than a threshold θ, then this node can be pruned as the density of the clusters of its children nodes will be less than θ.

We shall explain how the parent-child relation is established between nodes, which is needed to ensure anti-monotonicity of the reverse search tree. Let us denote degree as $deg_C(o) = \sum_{x_{oat} \in \{o\} \times A \times T} x_{oat}$, $deg_C(a) = \sum_{x_{oat} \in O \times \{a\} \times T} x_{oat}$ and $deg_C(t) = \sum_{x_{oat} \in O \times A \times \{t\}} x_{oat}$. Given that there is a cluster $C = (O, A, T)$, let us assume that we obtain cluster C' by removing the element $u \in \{O, A, T\}$ with the minimal degree $deg_C(u)$ from cluster C. There is a parent-child relation between clusters C' and C, where the node with cluster C' is the parent of the node with cluster C.

Twofold clustering (Definition 24) Günnemann et al. (2010a) proposed algorithm GAMER, which performs a depth-first search on a set enumeration tree of objects to mine twofold clusters. Each node of the tree represents a set of objects that potentially can form a twofold cluster. Pruning of the tree is based on the properties of the cluster definition and the optimal twofold clustering (cf. Definition 25).

However, not all invalid clusters can be pruned, as the density of the subgraphs and the redundancy relationship do not have monotonicity. To circumvent this problem,

all generated clusters are first stored in a queue. After the set enumeration tree is traversed, this queue will be processed to obtain the final clusters, based on the criteria of the optimal twofold clustering. This queue also stores subtrees that are not traversed yet, due to them having the possibility of containing redundant clusters. During the processing of queue, each subtree is checked and if it is found to be redundant, it is discarded. Otherwise, the subtree is traversed to mine the clusters.

Properties of the algorithms:

- *Complete result* The completeness is guaranteed as the traversal on the lattice to mine the clusters is deterministic and systematic.
- *Stable result* Similarly, the result is stable due to the deterministic and systematic traversal.
- *Efficiency* The algorithms generally exploit the properties of the clusters to prune the search space for efficient traversal. Another factor to determine the efficiency of the algorithms is the number of clusters enumerated, which is dependent on the setting of the parameters of the clusters (e.g. minimum size thresholds).
- *Semi-supervised* In constraint based subspace clustering, the object-level constraints are pushed into the set enumeration tree. This results in more efficient mining as the search space is aggressively pruned.
- *Up-to-date clustering* In stream data clustering, the clusters are constantly updated, and are not mined from scratch. The mining is also localized to the latest data stream of window size w for efficiency's sake. Hence, the user can control the efficiency of the algorithm via w.
- *Number of parameters* The algorithms using this method generally have a number of tuning parameters to set, and most of the tuning parameters are related to the clusters, and not the algorithms. There is an exception for stream data clustering, where its parameter w is related to the algorithm.
- *Parameter sensitivity* The results are sensitive to the tuning parameters of the clusters and the algorithms. As mentioned in Sect. 2.3.2, there are several weaknesses in methods that are sensitive to tuning parameters.

6.2 Statistical model method

In this method, the data is assumed to follow a statistical model, and clustering becomes a problem of estimating the parameters of the statistical model.

A common algorithm to estimate the parameters is the Expectation-Maximization (EM) algorithm. The EM algorithm is an iteration of estimation and maximization steps to compute the maximum likelihood estimate of the parameters, with respect to the data (Duda et al. 2001). In the estimation step, the expectation of the log-likelihood is calculated using the current estimated parameters. In the maximization step, the parameters maximizing the expected log-likelihood are estimated, and these parameters are then used in the next estimation step.

Bayesian overlapping subspace clustering (Sect. 5.1.4) In Bayesian overlapping subspace clustering (Fu and Banerjee 2009), the parameters of the Beta-Bernoulli distribution $\alpha_i^{\mathbb{O}}, \beta_i^{\mathbb{O}}, \alpha_i^{\mathbb{A}}, \beta_i^{\mathbb{A}}, \pi_i^o, \pi_i^a$ and exponential families θ_i (cf. Definition 16) are estimated using an EM-like algorithm. Let $\boldsymbol{\alpha}^{\mathbb{O}}, \boldsymbol{\beta}^{\mathbb{O}}, \boldsymbol{\alpha}^{\mathbb{A}}, \boldsymbol{\beta}^{\mathbb{A}}, \boldsymbol{\theta}, \boldsymbol{\pi}^o, \boldsymbol{\pi}^a$ be the

respective sets of parameters to be estimated, with $p(\mathbb{D}, Z^o, Z^a, \boldsymbol{\pi}^o, \boldsymbol{\pi}^a|\boldsymbol{\alpha}^{\mathbb{O}}, \boldsymbol{\beta}^{\mathbb{O}}, \boldsymbol{\alpha}^{\mathbb{A}}, \boldsymbol{\beta}^{\mathbb{A}}, \boldsymbol{\theta})$ as the likelihood. $Z^o = [\mathbf{z}^o]$ and $Z^a = [\mathbf{z}^a]$ are the $|\mathbb{O}| \times k$ and $|\mathbb{A}| \times k$ binary matrices of the k-dimensional latent bit vectors for objects and attributes respectively.

In the expectation step, the goal is to estimate the expectation of the log-likelihood $E[\log p(\mathbb{D}, Z^o, Z^a|\boldsymbol{\alpha}^{\mathbb{O}}, \boldsymbol{\beta}^{\mathbb{O}}, \boldsymbol{\alpha}^{\mathbb{A}}, \boldsymbol{\beta}^{\mathbb{A}}, \boldsymbol{\theta})]$. Gibbs sampling is used to approximate the expectation. Note that the sets of parameters $\boldsymbol{\pi}^o, \boldsymbol{\pi}^a$ do not need to be estimated as they are generated by the Beta distributions, which are conjugate priors to Bernoulli distributions which generate Z^o and Z^a. In the maximization step, the parameters $\boldsymbol{\alpha}^{\mathbb{O}*}$, $\boldsymbol{\beta}^{\mathbb{O}*}$, $\boldsymbol{\alpha}^{\mathbb{A}*}$, $\boldsymbol{\beta}^{\mathbb{A}*}$ and $\boldsymbol{\theta}^*$, which maximizes the expectation are estimated.

Properties of the algorithm:

- *Complete result* The completeness is guaranteed as k subspace clusters are required to be mined.
- *Stable result* As the method uses Gibbs sampling, which makes use of random numbers, it is possible that the result is different for each run. If the log-likelihood is not concave, then the method may have to run multiple times with different initializations of parameters to get the optimal result.
- *Efficiency* The number of iterations of the method to convergence is dependent on the size of the dataset.
- *Number of parameters* The algorithm has only one tuning parameter k to set, which specifies the number of clusters.
- *Parameter sensitivity* Although the cluster is parameter insensitive (cf. Definition 16), the result is highly sensitive to the algorithm's tuning parameter k, which influences the outcome of the clustering result. This problem of determining the number of clusters is known as the "fundamental problem of cluster validity", and a number of solutions can be found in (Xu and Wunsch 2005).

6.3 Approximation algorithm

Approximation algorithm is used when it is computationally infeasible to use the other methods. Approximation algorithms are normally developed in an ad hoc basis to suit the clustering problem. During the clustering process, the decision on how to proceed to the next step is greedy based, i.e. at the current step, the algorithm will decide the next step based on the current information it has. Hence, the completeness and quality of the results are sacrificed for the sake of computational feasibility. As the approximation algorithms normally lack theoretical foundations, the authors will usually conduct experiments to empirically show the effectiveness of the algorithms.

Subspace clustering for uncertain data (Definition 17) Günnemann et al. (2010b) proposed a greedy and iterative method to mine subspace clusters from uncertain data. Each iteration generates a subspace cluster, and the iteration repeats until no clusters are mined.

In an iteration, a number of medoids are randomly selected from the dataset, and for each medoid, subspace clusters with respect to the criteria of Definition 17 are mined. The clusters are mined using the lattice based algorithm (cf. Sect. 6.1), with the nodes of the lattice representing sets of attributes. At the end of the iteration, the subspace cluster with the best quality is selected among the clusters and outputted.

In the next iteration, objects which are in the selected subspace cluster and have high probability $P_{\leq w}(m, o, A)$ are excluded from the dataset to prevent the same subspace cluster to be mined again.

Properties of the algorithm:

- *Complete result* The completeness is not guaranteed as the traversal of the search space of the data is random and greedy based.
- *Stable result* The result is unstable as the medoids are randomly chosen. Monte-Carlo sampling is also used to calculate the probabilities.
- *Efficiency* The algorithm is efficient in relative to algorithms that traverse the complete search space.
- *Number of parameters* The algorithm does not have parameters, but it is affected by the parameters of the cluster (cf. Definition 17).
- *Parameter sensitivity* The algorithm is sensitive to the parameters of the cluster.

Relevance model (Definition 18) Mining significant subspace clusters under the relevance model is NP-hard and it is computationally expensive to first mine the complete set of subspace clusters ALL, and then mine the set of significant subspace clusters M^* from ALL. Hence, Müller et al. (2009a) proposed an approximation algorithm RESCU to mine the significant subspace clusters.

Algorithm RESCU uses a greedy based technique (Müller et al. 2009b) to 'jump' to subspace clusters that have high cluster gain. These subspace clusters with high cluster gain are mined on-demand and a list of them, ranked in the descending order of the cluster gain, is maintained. Let the set of significant subspace clusters be denoted as M^*, and RESCU uses an iteration process to mine this set. In each iteration i, a significant subspace cluster is picked from the list and added to M_i. The iteration process repeats until no more significant subspace clusters can be added, and the final M_i is approximated as M^*.

Orthogonal model (Definition 19) Similar to the relevance model, mining significant subspace clusters under the orthogonal model is NP-hard (Günnemann et al. 2009), and thus, an approximation algorithm OSCLU is proposed to mine them. Algorithm OSCLU consists of iterations of mining good quality subspaces, and mining significant subspace clusters from these subspaces.

OSCLU first traverses a lattice of attributes from bottom-up, breadth-first manner, where each node of the lattice represents a subspace (set of attributes). At a level of the lattice, OSCLU traverses all its subspaces, and each of them are evaluated and ranked according to (1) the number of overlaps it has with the subspaces of significant subspace clusters mined previously, and (2) the quality of the subspace, which can be measured using any of the methods described in Sect. 5.2.1. A subspace is ranked high if it has little overlapping and is of high quality.

Subspace clusters are then mined from the high ranking subspaces. When a cluster is mined, it will be checked using the criteria in Definition 19. If previously mined significant subspace clusters have lower I_{local} and have high overlaps with the newly mined cluster, then these significant subspace clusters are removed from the results.

Properties of the algorithms:

- *Complete result* The completeness is not guaranteed as there is no exhaustive traversal on the search space of the data.

- *Stable result* The result of algorithm OSCLU is unstable as some of its steps is random based, while algorithm RESCU's result is stable.
- *Efficiency* The algorithms are efficient in relative to algorithms that traverse the complete search space.
- *Number of parameters* The algorithms do not have parameters, but they are affected by the parameters of the clustering models.
- *Parameter sensitivity* The algorithms are sensitive to the parameters of their clustering models.

Statistical significant subspace clustering (Definition 20) The statistically significant subspace clusters are mined using a centroid based greedy algorithm, known as STATPC (Moise and Sander 2008). Each object o is taken as a centroid. For an attribute a, if the distribution of the value of o and its neighboring values deviates from the values expected under the assumption of uniform distribution (to a statistically significant degree), we denote attribute a as a signaled attribute of o.

The centroid o on its set of signaled attributes is taken as a subspace cluster, and objects that are closest to the subspace cluster on the set of signaled attributes are iteratively added to the cluster. Let R^{local} be a sequence of subspace clusters, with each cluster obtained from an iteration on centroid o. A locally optimal subspace cluster is then selected from R^{local}, such that it can *explain* or *induce* (cf. Assumption 1) all subspace clusters in R^{local}.

Each centroid o will have a locally optimal subspace cluster, and these locally optimal subspace clusters are randomly added into a set $R^{reduced}$, until all objects are involved in one of the locally optimal subspace clusters in $R^{reduced}$.

In the last phase, subspace clusters from $R^{reduced}$ is greedily and iteratively removed and added to a set M, until M is able to explain $R^{reduced}$. The final set of M is taken as the set of statistical significant subspace clusters.

Properties of the algorithm:

- *Complete result* The completeness is not guaranteed as there is no exhaustive traversal on the search space of the data.
- *Stable result* Its result is unstable since subspace clusters are randomly added to the set R^{local}
- *Efficiency* The algorithms are efficient in relative to algorithms that traverse the complete search space.
- *Number of parameters* There is a significance level to set, which is used in mining the signaled attributes. There is also a tuning parameter δ to set, which is the distance constraint used to determine the neighboring values of centroid o when finding the signaled attributes.
- *Parameter sensitivity* The sensitivity of the significance level for the signaled attributes is tested and a default significance level is chosen (Moise and Sander 2008). However, δ is chosen heuristically, and no experiments are conducted to test its sensitivity.

Correlated subspace clustering (Definition 22) In correlated subspace clustering (Sim et al. 2010a), pairs of values whose correlation information are significantly high (in statistical sense) are mined from the dataset, and these pairs of values are considered as seeds (building blocks) to build the correlated subspace clusters. The significance

of the correlation information of a pair of value is based on its p-value; a pair of values is considered significant, if the p-value of its correlation information is lower than a default significance level α, under the assumption that the correlation information is gamma distributed.

Each seed is taken as a centroid, and for each centroid, other seeds are greedily and iteratively added to it to create a subspace cluster. A seed is added to the subspace cluster if it leads to an increase of correlation information of the subspace cluster. This iterative addition continues until there is no more increase of correlation information of the subspace cluster.

Properties of the algorithm:

- *Complete result* The completeness is not guaranteed as the algorithm is greedy based, and only a portion of the values are used in the mining of the clusters.
- *Stable result* During an extension of a subspace cluster, if there are more than one seeds which give the highest correlation information increase to the cluster, the algorithm will randomly select one of the seeds for the extension. Hence, the result can be unstable.
- *Efficiency* The algorithm is efficient in relative to algorithms that traverse the complete search space. The algorithm may not be scalable to large datasets as it is computationally intensive; kernel density estimation is used to calculate the probability and the correlation information formula is a series of summation calculations.
- *Number of parameters* Unlike other subspace clustering methods, thresholds are not needed to determine if the correlation information of a subspace cluster is high. The algorithm has a parameter to set, which is the significance level α.
- *Parameter sensitivity* The default significance level is shown to be parameter-insensitive in the experiments in (Sim et al. 2010a).

6.4 Hybrid algorithm

The hybrid algorithm combines different techniques to mine subspace clusters, in such a way that the strengths of each technique is utilized to maximize the efficiency of the algorithm.

Actionable subspace clustering (Definition 23) Algorithm MASC is a centroid based algorithm that uses a hybrid of optimization and frequent pattern mining methods to mine actionable subspace clusters (Sim et al. 2010b).

Let $P = \{p_{o_1 a}, \ldots, p_{o_{|\mathbb{O}|} a}\}$ be the set of weights indicating if the objects should be part of a cluster containing centroid c, on attribute a (cf. Eq. 13). The optimal set of weights P is calculated by using the augmented Lagrangian multiplier method to optimize the objective function in Eq. 13.

After obtaining the optimal set of weights P, a heuristic method is used to determine a threshold to binarize the weights, such that weights that are above the threshold are set to '1' and the others as '0'. This procedure of optimization and binarization is repeated for each attribute $a \in \mathbb{A}$ to obtain a binary matrix $\mathbb{O} \times \mathbb{A}$. Maximal biclique subgraphs are then mined from this binary matrix, where each subgraph corresponds to an actionable subspace cluster.

Properties of the algorithm:

- *Complete result* The algorithm does not mine all actionable subspace clusters of the dataset. It only guarantees the result is complete with respect to the centroid.
- *Stable result* In (Sim et al. 2010b), Algorithm BCLM is used for the optimization, which produces stable results if the optimization problem is well conditioned, and condition of the problem is dependent on the data. The user can check if the problem is well conditioned, by making small perturbations to the data and see if the obtained results are similar.
- *Efficiency* The efficiency of the algorithm MASC is dependent on the optimization algorithm and graph mining algorithm used. The number of centroids and the number of attributes also determine the efficiency of MASC, as both numbers determine the number of iterations in MASC.
- *Number of parameters* The algorithm BCLM requires setting of four tuning parameters. Besides having to set parameters, the centroids have to be selected by the user too. Unless the user has the domain knowledge to select good quality centroids, selecting the right centroids can be difficult. For user without domain knowledge, the user is given the option of selecting centroids that have high average utility.
- *Parameter sensitivity* The default settings of algorithm BCLM are shown to be insensitive (Sim et al. 2010b; Nocedal and Wright 2006) in well conditioned optimization problems. Using an objective function to measure the goodness of the clusters is more robust to noise, since small changes in the dataset should not drastically reduce the goodness of the clusters.

6.5 Summary

Table 5 presents the desired properties of the subspace clustering algorithms. For the number of algorithm parameters, we do not consider parameters that can be both cluster and algorithm parameters.

7 Open problems

We discuss important open problems that have the potential to be future research areas of enhanced subspace clustering.

7.1 Towards tuning parameter-light and tuning parameter-insensitive mining

As mentioned in Sect. 2.3.1, setting tuning parameters is usually a 'guessing game', as most of the parameters are non-meaningful and non-intuitive to the user. Thus, these tuning parameters are usually set upon the biased assumptions of the user, resulting in highly skewed clusters.

A possible solution is to set the tuning parameters such that the run time of the algorithm is fast or the size of the result is small. However, setting the parameters to

Table 5 A summary of the desired properties of the subspace clustering algorithm

Algorithm	Complete result	Stable result	Up-to-date clustering	Number of parameters	Parameter-insensitive
Grid based subspace cluster	√	√		0	
Density based subspace cluster	√	√		0	
Window based subspace cluster	√	√		0	
Binary 3D subspace cluster	√	√		0	
Dense 3D subspace cluster	√	√		0	
Coherent gene cluster	√	√		0	
Tricluster	√	√		0	
Quasi-biclique	√	√		0	
Categorical subspace cluster	√	√		0	
Bayesian overlapping subspace cluster	√	√		1	
Subspace cluster for uncertain data				0	
Subspace α-cluster	√	√	√	1	
Entropy based subspace	√	√		0	
Interesting subspace	√	√		0	
Relevance model		√		0	
Orthogonal model				0	
Statistical significant subspace cluster				2	
Correlated subspace cluster				1	√
Constraint based subspace cluster	√	√		0	
Actionable subspace cluster		√		4	√
Twofold cluster	√	√		0	

suit the algorithm only serves to fulfill the mechanism of the algorithm, and not really extracting useful information from the data.

There is a raising interest in parameter-light and parameter-insensitive mining from the subspace clustering community recently, such as those described in Sect. 5.2, where they mitigate the problem of parameter-sensitivity by making statistical assumptions on the data distribution or using heuristics in their clustering.

In the other research domains, there are works (Faloutsos and Megalooikonomou 2007; Keogh et al. 2004) which advocate the use of compression theory such as Minimum Description Length (MDL) and Kolmogorov complexity to achieve parameter-free data mining. Perhaps, the same can be achieved in subspace clustering.

7.2 Statistical assumption-free mining

To overcome the problem of parameter-sensitive mining, some approaches such as the Bayesian overlapping clustering (Definition 16), statistical significant subspace clustering (Definition 20) and subspace clustering for uncertain data (Definition 17) assume that the data and the clusters are generated by statistical models. The clustering problem will translate to a problem of fitting the data and the clusters to the assumed statistical model, and estimating the parameters of the model. Hence, the clusters are described by the statistical model with the estimated parameters, and are not clusters that satisfy the parameters set by the user.

This approach is theoretically solid, but in order to be effective, the assumption of the statistical model on the data and clusters must be correct, and this assumption may become another 'guessing game', as we may not know the true distribution of the data. The perils of assuming statistical models in solving problems have being raised by Breiman (2001). In a nutshell, Breiman proposed that real-world data are generated by complex forces of nature and statistical models may be poor emulation of nature. Even if a model is fitted to a data, the discovered clusters are based on the model's mechanism and not on the nature's mechanism.

Färber et al. (2010) also discussed about the imperfections of the usage of models (including statistical models) in clustering. Let us assume that the true model of the data is known, and clusters that satisfy the model are found. However, the original purpose of clustering is to gain unknown and interesting knowledge from the data, and clusters that satisfy the model do not conform to this purpose. Moreover, there are no established ways to evaluate the usefulness of clusters.

7.3 Semi-supervised mining

Semi-supervised mining is about using additional information such as constraints or utility to improve the algorithm's efficiency or improving the clustering results. With the proliferation of data in recent times, more can be done in this area. An interesting technique that can be explored is transfer learning (Pan and Yang 2010), which has been successfully applied in co-clustering (known as self-taught clustering) (Dai et al. 2008). Given an auxiliary dataset, a target dataset and a set of common attributes, self-taught clustering uses the auxiliary dataset to improve the clustering results on the target dataset, by finding clusters in both datasets which share the common subspaces. Likewise in subspace clustering, we can transfer knowledge from auxiliary data to aid in the subspace clustering of target data.

7.4 Unified framework for subspace clustering

Many variations of subspace clusters have been proposed to deal with various types of data and to solve different problems. The current paradigm to solve subspace clustering problem is to first define the clusters of interest and their properties, and then develop an algorithm that can exploit the properties of the clusters (e.g. anti-monotonicity) to efficiently mine them. Hence, the cluster definition and its algorithm

are closely intertwined, and Kriegel et al. (2009) even identified some extreme approaches where the clusters are defined to match the algorithms.

Based on the current paradigm, extending existing definitions of 2D subspace cluster to 3D, or creating new definitions of subspace cluster, will result in new algorithms developed specifically for each cluster definition. It is not possible to simply take a new cluster definition and plug it into any of the existing algorithms to mine the desired clusters. However, this laborious process can be avoided, if the following problem is solved: "Is there a paradigm such that existing and new cluster definitions can be solved by a unifying algorithm?".

This question is pondered upon by Faloutsos and Megalooikonomou (2007). They argued that data mining, including clustering, are related to compression and Kolmogorov complexity. Calculating the optimal compression, which involves estimating the Kolmogorov complexity, translates to finding the optimal clusters. However, Kolmogorov complexity is undecidable, which means that the unceasing development of new algorithms for new data mining tasks is a necessity. Faloutsos and Megalooikonomou concluded that data mining is an art, where the goal is to find better models or patterns that fit the datasets. Therefore, it remains to be seen whether a unifying framework for subspace clustering is possible.

7.5 Post-processing of subspace clusters

As mentioned earlier, most of the research on subspace clustering are focused on defining the subspace clusters and how to efficiently mine them. The clusters are information extracted from the data, but not knowledge that is useful to the users. To convert information to knowledge, post-processing of the clusters is needed. Examples of post-processing techniques include limiting the number of clusters, organizing the clusters to explore the relations between them, or deriving models that represent the results. These proposed techniques are still in their infancy and much more possibilities can still be explored.

On limiting the number of clusters, it is done to prevent the user from being overwhelmed by the result and to allow easy analysis of the result. Mining significant subspace clusters is one of the approaches proposed to solve this. Several definitions of significance are defined by the different clustering approaches, but there is no universally accepted definition. Moreover, it is still possible that a large number of clusters can still be generated as long as they satisfy the significant criterion of the clustering approaches.

On organizing the clusters, Müller et al. (2008) developed a system known as Morpheus, which provides visualization and interactive exploration of subspace clusters. This can be a way to manually extract knowledge from the clusters, provided that the number of clusters is small. Achtert et al proposed arranging the projected clusters (Achtert et al. 2006a) and density based subspace clusters (Achtert et al. 2007) into a hierarchical structure to explore the relations between clusters. The clusters are arranged in levels, with each level representing the attributes' size of the cluster. An edge exists between two clusters if the objects of one of the cluster are contained in the other cluster. Organizing the clusters in hierarchical structure is a simple way to

explore the clusters' relations, but the hierarchical structure may become too messy when there are too many clusters.

On deriving models to represent the results, Achtert et al. (2006b) proposed deriving quantitative models for correlation clusters, to explain the linear dependencies within a correlation cluster. This concept can be bought to subspace clustering, where model of the subspace clustering results are derived to provide a summary of the results. The user will then have a general understanding of the results before delving into the details of each cluster.

8 Conclusion

Research in subspace clustering has progressed much since the pioneer paper by Agrawal et al. (1998). The original subspace clustering focuses on mining clusters from high-dimensional dataset, where objects in a cluster are closed together on a subspace of the dataset. In recent years, due to the proliferation of data and advancement of data collection, and the necessity to solve more complex and demanding tasks, the research trend has shifted from basic subspace clustering to enhanced subspace clustering. Enhanced subspace clustering focuses on two aspects: (1) handling complex data such as 3D data, categorical data, stream data or noisy data, and (2) improving the clustering results. In this survey, we presented the clustering problems, the cluster definitions and algorithms of enhanced subspace clustering. We also described the basic subspace clustering, the related high-dimensional clustering techniques, and explained how they are related. Research in subspace clustering has come a long way, but it is still a young and exciting area to work on, notably with the discussed open problems.

References

Achtert E, Böhm C, Kriegel HP, Kröger P, Müller-Gorman I, Zimek A (2006a) Finding hierarchies of subspace clusters. In: Proceedings of the 10th European conference on principles and practice of knowledge discovery in databases (PKDD), pp 446–453

Achtert E, Böhm C, Kriegel HP, Kröger P, Zimek A (2006b) Deriving quantitative models for correlation clusters. In: Proceedings of the 12th ACM international conference on knowledge discovery and data mining (KDD), pp 4–13

Achtert E, Böhm C, Kriegel HP, Kröger P, Müller-Gorman I, Zimek A (2007) Detection and visualization of subspace cluster hierarchies. In: Proceedings of the 12th international conference on database systems for advanced applications (DASFAA), pp 152–163

Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases. In: Proceedings of 20th international conference on very large data bases (VLDB), pp 487–499

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

Aggarwal CC, Wolf JL, Yu PS, Procopiuc C, Park JS (1999) Fast algorithms for projected clustering. In: Proceedings of the ACM international conference on management of data (SIGMOD), pp 61–72

Aggarwal CC, Hinneburg A, Keim D (2001) On the surprising behavior of distance metrics in high dimensional space. In: Proceedings of the 8th international conference on database theory (ICDT), pp 420–434

Aggarwal CC, Han J, Wang J, Yu PS (2004) A framework for projected clustering of high dimensional data streams. In: Proceedings of 30th international conference on very large data bases (VLDB), pp 852–863

Assent I, Krieger R, Müller E, Seidl T (2007) DUSC: dimensionality unbiased subspace clustering. In: Proceedings of the 7th IEEE international conference on data mining (ICDM), pp 409–414

Assent I, Krieger R, Müller E, Seidl T (2008a) EDSC: efficient density-based subspace clustering. In: Proceedings of the 17th ACM conference on information and knowledge management (CIKM), pp 1093–1102

Assent I, Krieger R, Müller E, Seidl T (2008b) INSCY: indexing subspace clusters with in-process-removal of redundancy. In: Proceedings of the 8th IEEE international conference on data mining (ICDM), pp 719–724

Avis D, Fukuda K (1996) Reverse search for enumeration. Discr Appl Math 65(1-3):21–46

Bennett KP, Fayyad U, Geiger D (1999) Density-based indexing for approximate nearest-neighbor queries. In: Proceedings of the 5th ACM international conference on knowledge discovery and data mining KDD, pp 233–243

Berkhin P (2006) A survey of clustering data mining techniques. In: Kogan J, Nicholas C, Teboulle M (eds) Grouping multidimensional data, chap 2. Springer, New York pp 25–71

Beyer K, Goldstein J, Ramakrishnan R, Shaft U (1999) When is "nearest neighbor" meaningful?. In: Proceedings of the 7th international conference on database theory (ICDT), pp 217–235

Böhm C, Kailing K, Kröger P, Zimek A (2004) Computing clusters of correlation connected objects. In: Proceedings of the ACM international conference on management of data (SIGMOD), pp 455–466

Breiman L (2001) Statistical modeling: the two cultures. Stat Sci 16(3):199–231

Cerf L, Besson J, Robardet C, Boulicaut JF (2008) Data peeler: contraint-based closed pattern mining in n-ary relations. In: Proceedings of the 8th SIAM international conference on data mining (SDM), pp 37–48

Cerf L, Besson J, Robardet C, Boulicaut JF (2009) Closed patterns meet *n*-ary relations. Trans Knowl Discov Data 3(1):1–36

Chan EY, Ching WK, Ng MK, Huang JZ (2004) An optimization algorithm for clustering using weighted dissimilarity measures. Pattern Recog 37(5):943–952

Cheng CH, Fu AW, Zhang Y (1999) Entropy-based subspace clustering for mining numerical data. In: Proceedings of the 5th ACM international conference on knowledge discovery and data mining (KDD), pp 84–93

Cheng Y, Church GM (2000) Biclustering of expression data. In: Proceedings of the 18th international conference on intelligent systems for molecular biology (ISMB), pp 93–103

Chiaravalloti AD, Greco G, Guzzo A, Pontieri L (2006) An information-theoretic framework for process structure and data mining. In: Proceedings of the 8th international conference on data warehousing and knowledge discovery (DaWaK), pp 248–259

Dai W, Yang Q, Xue GR, Yu Y (2008) Self-taught clustering. In: Proceedings of the 25th international conference on machine learning (ICML), pp 200–207

Dash M, Choi K, Scheuermann P, Liu H (2002) Feature selection for clustering - a filter solution. In: Proceedings of the 2nd IEEE international conference on data mining (ICDM), pp 115–122

Dhillon IS (2001) Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the 7th ACM international conference on knowledge discovery and data mining (KDD), pp 269–274

Dhillon IS, Mallela S, Modha DS (2003) Information-theoretic co-clustering. In: Proceedings of the 9th ACM international conference on knowledge discovery and data mining (KDD), pp 89–98

Ding CHQ, He X, Zha H, Simon HD (2002) Adaptive dimension reduction for clustering high dimensional data. In: Proceedings of the 2nd IEEE international conference on data mining (ICDM), pp 147–154

Domeniconi C, Papadopoulos D, Gunopulos D, Ma S (2004) Subspace clustering of high dimensional data. In: Proceedings of the 4th SIAM international conference on data mining (SDM), pp 517–521

Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, New York

Faloutsos C, Megalooikonomou V (2007) On data mining, compression, and kolmogorov complexity. Data Mining Knowl Discov 15(1):3–20

Färber I, Günnemann S, Kriegel HP, Kröger P, Müller E, Schubert E, Seidl T, Zimek A (2010) On using class-labels in evaluation of clusterings. In: Proceedings of the 1st international workshop on discovering, summarizing and using multiple clusterings (MultiClust) held in conjunction with KDD 2010

Francois D, Wertz V, Verleysen M (2007) The concentration of fractional distances. IEEE Trans Knowl Data Eng 19(7):873–886

Fromont É, Prado A, Robardet C (2009) Constraint-based subspace clustering. In: Proceedings of the 9th SIAM international conference on data mining (SDM), pp 26–37

Fu Q, Banerjee A (2009) Bayesian overlapping subspace clustering. In: Proceedings of the 9th IEEE international conference on data mining (ICDM), pp 776–781

Gao B, Liu TY, Ma WY (2006) Star-structured high-order heterogeneous data co-clustering based on consistent information theory. In: Proceedings of the 6th IEEE international conference on data mining (ICDM), pp 880–884

Georgii E, Tsuda K, Schölkopf B (2010) Multi-way set enumeration in weight tensors. Mach Learn 82(2):123–155

Guha S, Rastogi R, Shim K (1999) ROCK: a robust clustering algorithm for categorical attributes. In: Proceedings of the 15th international conference on data engineering (ICDE), pp 512–521

Günnemann S, Müller E, Färber I, Seidl T (2009) Detection of orthogonal concepts in subspaces of high dimensional data. In: Proceedings of the 18th ACM conference on information and knowledge management (CIKM), pp 1317–1326

Günnemann S, Färber I, Boden B, Seidl T (2010a) Subspace clustering meets dense subgraph mining: a synthesis of two paradigms. In: Proceedings of the 10th IEEE international conference on data mining (ICDM), pp 845–850

Günnemann S, Färber I, Müller E, Seidl T (2010b) ASCLU: alternative subspace clustering. In: Proceedings of the 1st international workshop on discovering, summarizing and using multiple clusterings (MultiClust) held in conjunction with KDD 2010

Günnemann S, Kremer H, Seidl T (2010c) Subspace clustering for uncertain data. In: Proceedings of the 10th SIAM international conference on data mining (SDM), pp 385–396

Hinneburg A, Aggarwal CC, Keim DA (2000) What is the nearest neighbor in high dimensional spaces?. In: Proceedings of the 26th international conference on very large data bases (VLDB), pp 506–515

Houle ME, Kriegel HP, Kröger P, Schubert E, Zimek A (2010) Can shared-neighbor distances defeat the curse of dimensionality?. In: Proceedings of the 22nd international conference on scientific and statistical database management (SSDBM)

Hsu CM, Chen MS (2004) Subspace clustering of high dimensional spatial data with noises. In: Proceedings of the 8th Pacific-Asia conference advances in knowledge discovery and data mining (PAKDD), pp 31–40

Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323

Jaschke R, Hotho A, Schmitz C, Ganter B, Stumme G (2006) TRIAS–an algorithm for mining iceberg tri-lattices. In: Proceedings of the 6th IEEE international conference on data mining (ICDM), pp 907–911

Ji L, Tan KL, Tung AKH (2006) Mining frequent closed cubes in 3D datasets. In: Proceedings of the 32nd international conference on very large data bases (VLDB), pp 811–822

Jiang D, Pei J, Ramanathan M, Tang C, Zhang A (2004a) Mining coherent gene clusters from gene-sample-time microarray data. In: Proceedings of the 10th ACM international conference on knowledge discovery and data mining (KDD), pp 430–439

Jiang D, Tang C, Zhang A (2004b) Cluster analysis for gene expression data: a survey. IEEE Trans Knowl Data Eng 16(11):1370–1386

Jing L, Ng MK, Huang JZ (2007) An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. IEEE Trans Knowl Data Eng 19(8):1026–1041

Kailing K, Kriegel HP, Kröger P, Wanka S (2003) Ranking interesting subspaces for clustering high dimensional data. In: Proceedings of the 7th European conference on principles and practice of knowledge discovery in databases (PKDD), pp 241–252

Kailing K, Kröger P, Kriegel HP (2004) Density-connected subspace clustering for high-dimensional data. In: Proceedings of the 4th SIAM international conference on data mining (SDM), pp 246–257

Ke Y, Cheng J, Ng W (2006) Mining quantitative correlated patterns using an information-theoretic approach. In: Proceedings of the 12th ACM international conference on knowledge discovery and data mining (KDD), pp 227–236

Keogh EJ, Lonardi S, Ratanamahatana CA (2004) Towards parameter-free data mining. In: Proceedings of the 10th ACM international conference on knowledge discovery and data mining (KDD), pp 206–215

Kleinberg J, Papadimitriou C, Raghavan P (1998) A microeconomic view of data mining. Data Mining Knowl Discov 2(4):311–324

Kohavi R, John G (1997) Wrappers for feature subset selection. Artif Intell 97(1-2):273–324

Kontaki M, Papadopoulos AN, Manolopoulos Y (2008) Continuous subspace clustering in streaming time series. Inf Syst 33(2):240–260

Kriegel HP, Zimek A (2010) Subspace clustering, ensemble clustering, alternative clustering, multiview clustering: what can we learn from each other? In: Proceedings of the 1st international workshop on discovering, summarizing and using multiple clusterings (MultiClust) held in conjunction with KDD 2010

Kriegel HP, Kröger P, Renz M, Wurst S (2005) A generic framework for efficient subspace clustering of high-dimensional data. In: Proceedings of the 5th IEEE international conference on data mining (ICDM), pp 250–257

Kriegel HP, Borgwardt KM, Kröger P, Pryakhin A, Schubert M, Zimek A (2007) Future trends in data mining. Data Mining Knowl Discov 15(1):87–97

Kriegel HP, Kröger P, Zimek A (2009) Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Trans Knowl Discov Data 3(1):1–58

Kriegel HP, Kröger P, Ntoutsi I, Zimek A (2011) Density based subspace clustering over dynamic data. In: Proceedings of the 23rd international conference on scientific and statistical database management (SSDBM), pp 387–404

Li T, Ma S, Ogihara M (2004) Document clustering via adaptive subspace iteration. In: Proceedings of the 27th ACM international conference on research and development in information retrieval (SIGIR), ACM, pp 218–225

Li J, Li H, Soh D, Wong L (2005) A correspondence between maximal complete bipartite subgraphs and closed patterns. In: Proceedings of the 9th European conference on principles and practice of knowledge discovery in databases (PKDD), pp 146–156

Li J, Sim K, Liu G, Wong L (2008) Maximal quasi-bicliques with balanced noise tolerance: concepts and co-clustering applications. In: Proceedings of the 8th SIAM international conference on data mining (SDM), pp 72–83

Liu G, Sim K, Li J (2006) Efficient mining of large maximal bicliques. In: Proceedings of the 8th international conference on data warehousing and knowledge discovery (DaWak), pp 437–448

Liu G, Sim K, Li J, Wong L (2009) Efficient mining of distance-based subspace clusters. Stat Anal Data Mining 2(5-6):427–444

Madeira SC, Oliveira AL (2004) Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans Comput Biol Bioinform 1(1):24–45

Mishra N, Ron D, Swaminathan R (2005) A new conceptual clustering framework. Mach Learn 56(1-3): 115–151

Moise G, Sander J (2008) Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering. In: Proceedings of the 14th ACM international conference on knowledge discovery and data mining (KDD), pp 533–541

Moise G, Zimek A, Kröger P, Kriegel HP, Sander J (2009) Subspace and projected clustering: experimental evaluation and analysis. Knowl Inf Syst 21(3):299–326

Müller E, Assent I, Krieger R, Jansen T, Seidl T (2008) Morpheus: interactive exploration of subspace clustering. In: Proceedings of the 14th ACM international conference on knowledge discovery and data mining (KDD), pp 1089–1092

Müller E, Assent I, Günnemann S, Krieger R, Seidl T (2009a) Relevant subspace clustering: mining the most interesting non-redundant concepts in high dimensional data. In: Proceedings of the 9th IEEE international conference on data mining (ICDM), pp 377–386

Müller E, Assent I, Krieger R, Günnemann S, Seidl T (2009b) DensEst: density estimation for data mining in high dimensional spaces. In: Proceedings of the 9th SIAM international conference on data mining (SDM), pp 173–184

Müller E, Assent I, Seidl T (2009c) HSM: heterogeneous subspace mining in high dimensional. In: Proceedings of the 21st international conference on scientific and statistical database management (SSDBM), pp 497–516

Müller E, Günnemann S, Assent I, Seidl T (2009d) Evaluating clustering in subspace projections of high dimensional data. Proc VLDB Endow 2(1):1270–1281

Nagesh H, Goil S, Choudhary A (2001) Adaptive grids for clustering massive data sets. In: Proceedings of the 1st SIAM international conference on data mining (SDM)

Nocedal J, Wright SJ (2006) Numerical optimization. Springer, New York 497–528

Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359

Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. ACM SIGKDD Explor Newsl 6(1):90–105

Pasquier N, Bastide Y, Taouil R, Lakhal L (1999) Discovering frequent closed itemsets for association rules. In: Proceedings of the 7th international conference on database theory (ICDT), pp 398–416

Patrikainen A, Meila M (2006) Comparing subspace clusterings. IEEE Trans Knowl Data Eng 18(7):902–916

Pensa R, Boulicaut J (2008) Constrained co-clustering of gene expression data. In: Proceedings of the 8th SIAM international conference on data mining (SDM), pp 25–36

Rege M, Dong M, Fotouhi F (2006) Co-clustering documents and words using bipartite isoperimetric graph partitioning. In: Proceedings of the 6th IEEE international conference on data mining (ICDM), pp 532–541

Rymon R (1992) Search through systematic set enumeration. In: Proceedings of the 8th international conference on principles and knowledge representation and reasoning (KR), pp 539–550

Sequeira K, Zaki MJ (2004) SCHISM: a new approach for interesting subspace mining. In: Proceedings of the 4th IEEE international conference on data mining (ICDM), pp 186–193

Silverman BW (1986) Density estimation for statistics and data analysis (Chapman and Hall/CRC monographs on statistics and applied probability), 1st edn. Chapman and Hall/CRC, London

Sim K, Li J, Gopalkrishnan V, Liu G (2006) Mining maximal quasi-bicliques to co-cluster stocks and financial ratios for value investment. In: Proceedings of the 6th IEEE international conference on data mining (ICDM), pp 1059–1063

Sim K, Gopalkrishnan V, Chua HN, Ng SK (2009a) MACs: multi-attribute co-clusters with high correlation information. In: Proceedings of the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD), pp 398–413

Sim K, Li J, Gopalkrishnan V, Liu G (2009b) Mining maximal quasi-bicliques: novel algorithm and applications in the stock market and protein networks. Stat Anal Data Mining 2(4):255–273

Sim K, Aung A, Vivekanand G (2010a) Discovering correlated subspace clusters in 3D continuous-valued data. In: Proceedings of the 10th IEEE international conference on data mining (ICDM), pp 471–480

Sim K, Poernomo AK, Gopalkrishnan V (2010b) Mining actionable subspace clusters in sequential data. In: Proceedings of the 10th SIAM international conference on data mining (SDM), pp 442–453

Sim K, Liu G, Gopalkrishna V, Li J (2011) A case study on financial ratios via cross-graph quasi-bicliques. Inf Sci 181(1):201–216

Snedecor GW, Cochran WG (1989) Statistical methods, 8th edn. Iowa State University Press, Ames

Srikant R, Agrawal R (1996) Mining quantitative association rules in large relational tables. In: Proceedings of the ACM international conference on management of data (SIGMOD), pp 1–12

Sun J, Faloutsos C, Papadimitriou S, Yu PS (2007) Graphscope: parameter-free mining of large time-evolving graphs. In: Proceedings of the 13th ACM international conference on knowledge discovery and data mining (KDD), pp 687–696

Tanay A, Sharan R, Shamir R (2004) Biclustering algorithms: a survey. Handbook of computational molecular biology. Chapman & Hall/CRC, London

Tomita E, Tanaka A, Takahashi H (2004) The worst-case time complexity for generating all maximal cliques. In: Proceedings of the 10th international computing and combinatorics conference (COCOON), pp 161–170

Uno T, Kiyomi M, Arimura H (2004) LCM ver. 2: efficient mining algorithms for frequent/closed/maximal itemsets. In: Proceedings of the 2nd international workshop on frequent itemset mining implementations (FIMI) held in conjuction with ICDM 2004

Vreeken J, Zimek A (2011) When pattern met subspace cluster—a relationship story. In: Proceedings of the 2nd international workshop on discovering, summarizing and using multiple clusterings (MultiClust) held in conjunction with ECML PKDD 2011, pp 7–18

Wagstaff K, Cardie C, Rogers S, Schrödl S (2001) Constrained k-means clustering with background knowledge. In: Proceedings of the 18th international conference on machine learning (ICML), pp 577–584

Wang H, Wang W, Yang J, Yu PS (2002) Clustering by pattern similarity in large data sets. In: Proceedings of the ACM international conference on management of data (SIGMOD), pp 394–405

Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678

Xu X, Lu Y, Tung AKH, Wang W (2006) Mining shifting-and-scaling co-regulation patterns on gene expression profiles. In: Proceedings of the 22nd international conference on data engineering (ICDE), p 89

Xu X, Lu Y, Tan KL, Tung AKH (2009) Finding time-lagged 3D clusters. In: Proceedings of the 25th international conference on data engineering (ICDE), pp 445–456

Yan C, Burleigh JG, Eulenstein O (2005) Identifying optimal incomplete phylogenetic data sets from sequence databases. Mol Phylogenet Evol 35:528–535

Yang J, Wang W, Wang H, Yu P (2002) δ-clusters: capturing subspace correlation in a large data set. In: Proceedings of the 19th international conference on data engineering (ICDE), pp 517–528

Zaki MJ, Peters M, Assent I, Seidl T (2005) CLICKS: an effective algorithm for mining subspace clusters in categorical datasets. In: Proceedings of the 11th ACM international conference on knowledge discovery and data mining (KDD), pp 736–742

Zhang X, Wang W (2007) An efficient algorithm for mining coherent patterns from heterogeneous microarrays. In: Proceedings of the 19th international conference on scientific and statistical database management (SSDBM), p 32

Zhang Q, Liu J, Wang W (2007) Incremental subspace clustering over multiple data streams. In: Proceedings of the 7th IEEE international conference on data mining (ICDM), pp 727–732

Zhao L, Zaki MJ (2005) TRICLUSTER: an effective algorithm for mining coherent clusters in 3D microarray data. In: Proceedings of the 25th ACM international conference on management of data (SIGMOD), pp 694–705