Chapter 16
Frequent Pattern Mining Algorithms for Data Clustering

Arthur Zimek, Ira Assent and Jilles Vreeken

Abstract Discovering clusters in subspaces, or subspace clustering and related clustering paradigms, is a research field where we find many frequent pattern mining related influences. In fact, as the first algorithms for subspace clustering were based on frequent pattern mining algorithms, it is fair to say that frequent pattern mining was at the cradle of subspace clustering—yet, it quickly developed into an independent research field.

In this chapter, we discuss how frequent pattern mining algorithms have been extended and generalized towards the discovery of local clusters in high-dimensional data. In particular, we discuss several example algorithms for subspace clustering or projected clustering as well as point out recent research questions and open topics in this area relevant to researchers in either clustering or pattern mining.

Keywords Subspace clustering · Monotonicity · Redundancy

1 Introduction

Data clustering is the task of discovering groups of objects in a data set that exhibit high similarity. Clustering is an unsupervised task, in that we do not have access to any additional information besides some geometry of the data, usually represented by some distance function. Useful groups should consist of objects that are more similar to each other than to objects assigned to other groups. The goal of the clustering results is that it provides information for the user regarding different categories of objects that the data set contains.

A. Zimek (✉)
Ludwig-Maximilians-Universität München, Munich, Germany
e-mail: zimek@dbs.ifi.lmu.de

I. Assent
Department of Computer Science, Aarhus University, Aarhus, Denmark
e-mail: ira@cs.au.dk

J. Vreeken
Max-Planck Institute for Informatics and Saarland University, Saarbrücken, Germany
e-mail: jilles@mpi-inf.mpg.de

C. C. Aggarwal, J. Han (eds.), *Frequent Pattern Mining,*
DOI 10.1007/978-3-319-07821-2_16,

As there are many different intuitions on how objects can be similar, there exist many different clustering algorithms for formalizing these intuitions, and extracting such clusters from data [43–45, 51]. There are two main approaches to clustering. On the one hand we find so-called *partitional* algorithms [26, 49, 55, 56], where similarity of objects is directly expressed in a notion of spatial closeness. For example, a smaller Euclidean distance between two points than between other pairs of points in Euclidean space makes them relatively similar. On the other hand we have *density-based* approaches [8, 20, 28, 39, 40, 75, 77], where similarity is expressed in terms of density-connectivity of points. That is, points that find themselves in a densely populated area in the data space are said to be 'connected' and should be assigned to the same cluster, whereas areas of relatively low density separate different clusters.

An important point to note for unsupervised learning in general, and clustering specifically, is that the cluster structure of the data—and hence that discovered by a particular clustering algorithm—does not necessarily have to correlate with class label annotations: clusters 'simply' identify structure that exists in the data [29, 36]. This means both that clustering requires methods different from classification, as well as that for evaluating clusters we cannot rely just on class labels.

Over the last 15 years, a lot of research effort has been invested to develop clustering algorithms that can handle high-dimensional data. Compared to traditional data with only few attributes, high-dimensional data incur particular challenges, most prominently the difficulty of assessing the similarity of objects in a meaningful manner. These issues are generally known as the 'curse of dimensionality'. Important aspects of this infamous 'curse' and its consequences for clustering (and related tasks) have been discussed in various studies, surveys, and overview articles [4, 9, 17, 18, 27, 30, 41, 42, 50, 52, 76, 83, 85].

A special family of adaptations of clustering approaches to high-dimensional data is known as 'subspace clustering'. Here the idea is that clusters do not necessarily exhibit similarity over all attributes, but that their similarity may be restricted to subsets of attributes; the other attributes are not relevant to the cluster structure. In effect, there is a need for algorithms that can measure similarity of objects, and hence detect clusters, over *subspaces*. Different subspaces can be relevant for different clusters while the clusters can be obfuscated by the noise of the remaining, 'irrelevant' attributes. There exist many similarities of this problem setting to that of mining frequent patterns, and in fact algorithmic ideas originally developed for frequent pattern mining form the foundations of the paradigm of *subspace clustering* [7].

As in pattern mining, the general intuition in subspace clustering is that an object may be a member of *several* clusters, over *different* subsets of the attributes. In this manner, it is possible to group the data differently depending on the features that are considered. Figure 16.1 gives an example. As we can see, the projection to different subspaces results in different clusters, but not all dimensions contribute to the patterns. In the leftmost projection to the subspace consisting of dimensions x and y, two groups are visible that are different from the groups seen in the center projection to dimensions w and z (note that symbols are consistent across the projections shown). Interestingly, the subspace y and z does not show any clear subspace clusters. The interesting observation here is that this view of different aspects of the

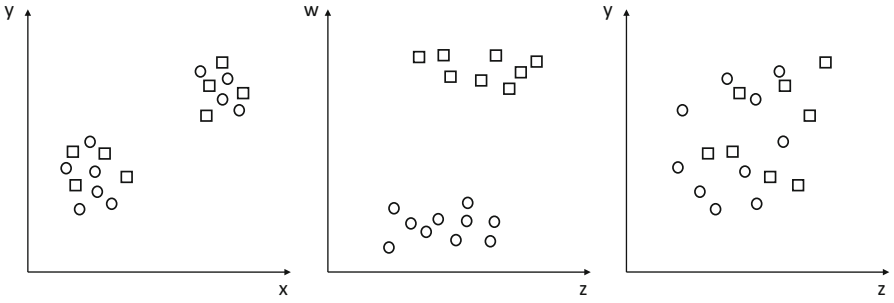


Fig. 16.1 Subspace clustering: two different groupings of the same data are seen when considering the subspace consisting of dimensions x and y (*left*) or the subspace consisting of dimensions z and w (*center*), whereas the subspace projection y and z (*right*) does not show any clear clusters

Fig. 16.2 Frequent itemset mining: transactions for the example are listed (*left*), frequent itemsets are detected when considering just the combination of item a and c, or when considering a and d, but not when considering e.g. c and d

Transactions	
1	a c
2	a c e
3	a d
4	a b c
5	a d
6	a b d
7	a d e

Example frequencies

a c	4 times
a d	4 times
c d	not found

data is present in frequent itemset mining as well (cf. Fig. 16.2): an item can be part of two different patterns such as $\{a, c\}$ or $\{a, d\}$, but the combination of $\{c, d\}$ does not necessarily yield frequent patterns.

There are several surveys and overview articles, discussing specifically subspace clustering [9, 50, 52, 53, 67, 74, 83], some of which also point out the connection to frequent pattern mining algorithms. The first survey to discuss the young field was presented by Parsons et al. [67], putting the research community's attention to the problem and sketching a few early algorithms. In the following years, the problem was studied in much more detail, and categories of similar approaches have been defined [50]. A short discussion of the fundamental problems and strategies has been provided by Kröger and Zimek [53]. Assent gives an overview in the context of high-dimensional data of different provenance, including time series and text documents [9]. Sim et al. [74] discuss 'enhanced' subspace clustering, i.e., they point out particular open problems in the field and discuss methods specifically addressing those problems. Kriegel et al. [52] give a concise overview and point to open questions as well. Based on this overview, an updated discussion was given by Zimek [83]. Recent textbooks by Han et al. [38], and Gan et al. [31], sketch prominent issues and example algorithms. Recent experimental evaluation studies compared some subspace clustering algorithms [60, 63].

The close relationship between the two areas subspace clustering and frequent pattern mining has been elaborated in a broader perspective by Zimek and Vreeken

[84]. Here, we will go more into detail of how the ideas of frequent pattern mining have been transferred and translated to the clustering domain, and how exactly they have found use in various clustering algorithms. To this end, in Sect. 2 we will first discuss the generalization of the reasoning about frequent patterns for the application to the clustering task. We will then, in Sect. 3, detail example algorithms for both subspace clustering and subspace search, discussing the use of ideas proposed in frequent pattern mining in these algorithms. We conclude the chapter in Sect. 4.

2 Generalizing Pattern Mining for Clustering

For a reader of a chapter in this book about frequent pattern mining, we assume familiarity with frequent pattern mining as discussed also in fundamental other chapters in this book. In particular, we assume basic knowledge of the Apriori algorithm [6]. Nevertheless, for the sake of completeness, let us briefly recapitulate the algorithmic ingredients of Apriori that are essential to our discussion.

Considering the example of market basket analysis, we are interested in finding items that are sold together (i.e., itemsets). Naïvely, the search for all frequent itemsets is exponential in the number of available items: we would simply calculate the frequency of all k-itemsets in the database over m items, resulting in $\sum_{k=1}^{m} \binom{m}{k} = 2^m - 1$ tests.

For identification of frequent patterns in a transaction database (i.e., a binary database, where each row does or does not contain a certain item), the idea of Apriori is a level-wise search for itemsets of incremental length, given a frequency threshold. Starting with all frequent itemsets of length 1 (i.e., counting all transactions containing a certain item, irrespective of other items possibly also contained in the transaction), the list of potential candidates for *frequent* itemsets of length 2 can be restricted based on the following observation: An itemset of length 2 can only be frequent if both contained items (i.e., itemsets of length 1) are frequent as well. If neither diapers nor beer is a frequent item in the transaction database, the transaction containing both diapers and beer cannot be frequent either. This holds for itemsets of all lengths n, that can only be frequent if all contained itemsets of length $n-1$ are frequent as well. For example, an itemset may contain items A, B, C, etc. If a 1-itemset containing A is not frequent (i.e., we find such an itemset less often than a given threshold), all 2-itemsets containing A (e.g., $\{A, B\}, \{A, C\}, \{A, D\}$) cannot be frequent either (otherwise itemsets containing A would have been frequent as well) and need not be tested for exceeding the threshold. Likewise, if the itemset $\{A, B\}$ is not frequent, then all 3-itemsets containing $\{A, B\}$ (e.g., $\{A, B, C\}$, $\{A, B, D\}$, $\{A, B, E\}$) cannot be frequent either, etc. Theoretically, the search space remains exponential, yet practically the search is usually substantially accelerated.

This observation is a principle of monotonicity and is the most important ingredient for a heuristic speed-up of the mining for frequent patterns. More concisely, we can express this monotonicity over sets as follows:

$$\mathcal{T} \text{ is frequent} \Rightarrow \forall \mathcal{S} \subseteq \mathcal{T} : \mathcal{S} \text{ is frequent.} \tag{16.1}$$

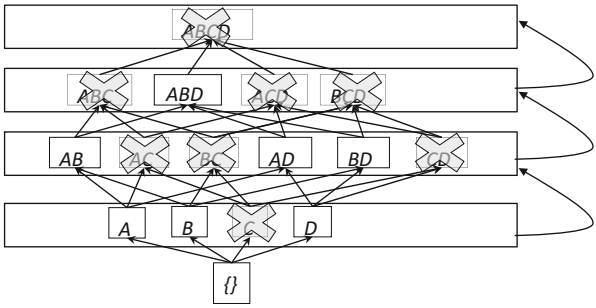


Fig. 16.3 Pruned search space during iterative database scans of Apriori (example): itemset $\{C\}$ has been found infrequent in the first scan, therefore, itemsets $\{A, C\}$, $\{B, C\}$, $\{C, D\}$ do not need to be considered in the second scan, itemsets $\{A, B, C\}$, $\{A, C, D\}$, $\{B, C, D\}$ do not need to be considered in the third scan, etc. In this example, Apriori stops scanning the database after round three, as there is no candidate of length 4 remaining

More precisely, the pruning criterion used in the Apriori algorithm is based on the equivalent anti-monotonic property, describing the opposite direction of deduction:

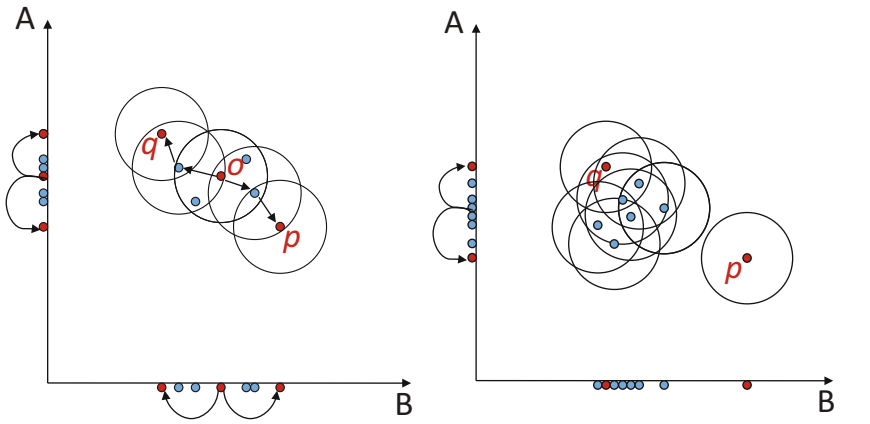
$$\mathcal{S} \text{ is } \textit{not} \text{ frequent} \Rightarrow \forall \mathcal{T} \supseteq \mathcal{S} : \mathcal{T} \text{cannot be frequent either.} \tag{16.2}$$

In the iterative procedure of repeated scans of the database for frequent itemsets, this anti-monotonic property allows to ignore candidates that cannot be frequent and, eventually, this pruning allows stopping at a certain size of itemsets, when no candidates of typically moderate size remain to generate larger itemsets (see Fig. 16.3).

An extension of the Apriori idea for very large itemsets has been termed 'colossal patterns' [82]. The observation is that if one is interested in finding very large frequent itemsets, then Apriori needs to generate many smaller frequent itemsets that are not relevant for the result. This effect can be used positively, in that if large patterns also have a large number of subsets, several of these subsets can be combined in order to obtain larger candidates directly. In this sense, the idea is to avoid the full search, and instead use some results at the bottom of the search space as a shortcut to particularly promising candidates higher up. This approach thus trades some of the accuracy of full search for a much more efficient frequent pattern mining algorithm. As we will see below, both the Apriori algorithm, as well as that of colossal patterns have been employed towards mining subspace clusters.

2.1 Generalized Monotonicity

In data clustering, we typically do not consider binary transaction data, or discrete data in general, but instead most often study continuous real-valued vector data, typically assuming a Euclidean vector space. In this space, attributes may be noisy, or



a Objects p and q are density-connected, i.e., they can build a density-based cluster, in subspaces $\{A, B\}$, $\{A\}$, and $\{B\}$.

b Objects p and q are not density-connected in subspace $\{B\}$. Therefore they cannot be density-connected in subspace $\{A, B\}$ either.

Fig. 16.4 Transfer of anti-monotonicity to subspace clusters

even completely irrelevant for certain clusters. If we measure similarity over the full space, i.e., over all attributes, detecting such 'subspace' clusters becomes increasingly difficult for higher numbers of irrelevant dimensions. To the end of identifying the relevant attributes, and measuring similarity only over these, the fundamental algorithmic idea of Apriori has been transferred to clustering in Euclidean spaces, giving rise to the task of 'subspace clustering', which has been defined as 'finding all clusters in all subspaces' [7].

Over time, this transfer has been done in different ways. The most important variants are to identify subspace clusters that in turn qualify some subspace as 'frequent pattern', or to identify interesting subspaces without direct clustering, but as a prerequisite for subsequent clustering in these subspaces or as an integral part of some clustering procedure.

Subspace Clusters as Patterns Let us consider the case of clusters in different subspaces with an example for density-based clusters [51], as visualized in Fig. 16.4. In the first scenario, depicted in Fig. 16.4a, we see that objects p and q are density-connected with respect to some parameters in subspaces $\{A, B\}$, $\{A\}$, and $\{B\}$. Here, the parameters capturing density are a distance threshold defining the radius of the neighborhood ball and a minimum number of points required to fall within this neighborhood ball in order to qualify as dense. That is, within these subspaces, we can reach both p and q starting at o by 'hopping' from one object with at least n neighbors within ε distance to another. This means that with these parameters, p and q belong to the same density-based cluster in each of these subspaces.

In the second scenario, depicted in Fig. 16.4b, p and q are again density-connected in subspace $\{A\}$, but not in subspace $\{B\}$. As a result from monotonicity, they therefore are also not density-connected in subspace $\{A, B\}$.

Table 16.1 Translating frequent pattern mining concepts to subspace clustering

Frequent pattern mining	Subspace clustering
Item	Dimension (attribute)
Itemset	Subspace (set of attributes)
Frequent itemset	Subspace (unit) containing cluster

Consequently, a set of points cannot form a cluster in some space $\mathcal{T}$, if it does not also form a cluster in every subspace of $\mathcal{T}$. Or, formulated as anti-monotone property that we can use to prune candidate subspaces:

$$\mathcal{S} \text{ does not contain any cluster} \Rightarrow \forall \text{ superspaces } \mathcal{T} \supseteq \mathcal{S}: \quad \mathcal{T} \text{ cannot contain a cluster either.} \tag{16.3}$$

As a result, an algorithm for subspace clustering can identify all clusters in all 1-dimensional subspaces, continue to look for clusters in only those 2-dimensional subspaces that have a 1-dimensional subspace containing some cluster, and so on, following the candidate-pruning heuristic of the Apriori algorithm. Hence we see that the 'items' of Apriori translate to dimensions, 'itemsets' translate to subspaces, and 'frequent itemset' according to some frequency threshold translates to 'subspace contains some cluster' according to some clustering criterion. See Table 16.1 for a summary of this translation. This transfer of concepts requires the anti-monotonicity to hold for the clustering criterion used.

Note that the monotonicity does not hold in general for arbitrary cluster paradigms, but instead depends on the particular cluster model used. The example used here (monotonicity of density-based clusters, Fig. 16.4) has been proven for the subspace clustering approach SUBCLU [48]. However, the very idea of using a monotonicity for some cluster criterion has been used for different clustering models several times, following the seminal approach of CLIQUE [7]. We detail the specific adaptations for different clustering models in the next section.

Subspaces as Patterns In the second main variant, the setup is slightly modified, and the goal is to identify subspaces as a prerequisite for the final clustering result. These subspaces can be used in quite different ways in connection with clustering algorithms. For example, after identification of subspaces, traditional clustering algorithms are applied to find clusters within these subspaces, or distance measures can be adapted to these subspaces in the actual clustering procedure, or clusters and corresponding subspaces are refined iteratively. As such, in contrast to the setting above, here one does not identify whether subspaces are 'interesting' by the clusters they contain (which is specific to a particular clustering model), but rather defines 'interesting' more generally, for example in terms of how strongly these attributes interact.

In subspace search, just as in subspace clustering, the 'items' and 'itemsets' concepts from frequent pattern mining translate nicely to 'dimension' and 'subspace', respectively. The notion of a 'frequent itemset' according to some frequency threshold translates different here, namely to 'interesting subspace' according to some

Table 16.2 Translating frequent pattern mining concepts to subspace search

Frequent pattern mining	Subspace search
Item	Dimension (attribute)
Itemset	Subspace (set of attributes)
Frequent itemset	'Interesting' subspace

measure of 'interestingness' (see Table 16.2 for a summary). How to measure this 'interestingness' in a way to satisfy anti-monotonicity is the crucial question that differs from approach to approach. Let us note that many methods follow the general idea of candidate elimination in subspace search without adhering to a criterion of *strict* anti-monotonicity, i.e., they rely on some observation that anti-monotonicity of their criterion 'usually' holds.

2.2 Count Indexes

Generalized monotonicity is a very useful property towards pruning the search space in both frequent itemset mining and subspace clustering. As part of the Apriori algorithm, however, candidate itemsets or subspaces have to be generated. For large sets of items and high-dimensional subspaces (i.e., subspaces with very many attributes), this can be a performance bottleneck [37].

Taking a different approach, the so-called FP-Growth algorithm uses a specialized index structure to maintain frequency counts of itemsets, the FP-tree [37]. As illustrated in Fig. 16.5, a node in this count index corresponds to the frequency count of a particular item, and following a path from an item to the root corresponds to the frequency count of a particular combination of items into an itemset. The index can be constructed in two data scans, where the first finds all frequent items, and the second creates nodes and updates counts for each transaction.

The FP-Growth algorithm is a depth-first approach. Starting from the most frequent item, the corresponding combinations with other items are 'grown' by recursively extracting the corresponding paths, until the index has been reduced to one path. The advantage of this method is that only frequent itemsets are generated, and that only two scans over the data are necessary in order to do so.

As we will detail in the next section, this idea of compactly representing interesting combinations in a count index and of proceeding in a depth-first traversal of the search space has also been applied to subspace clustering. This application is not straightforward due to the fact that both relevant subspace regions, as well as a notion of similarity between adjacent regions has to be defined; concepts that do not have one-to-one counterparts in frequent pattern mining.

2.3 Pattern Explosion and Redundancy

The downside to the frequency criterion and its monotonicity in frequent itemset mining is that with a threshold low enough to avoid exclusion of all but the most

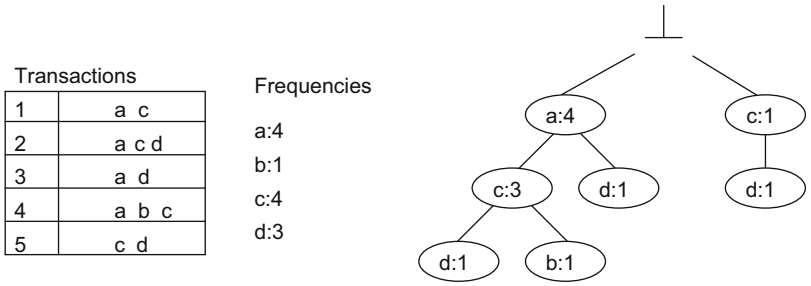


Fig. 16.5 FP-tree example: the tree nodes store items and their counts, paths correspond to combinations of itemsets and their respective counts. The index is built in just two scans over the data, and the frequent itemset mining algorithm FP-Growth works exclusively on the index. Once individual item frequencies are established, the second scan updates counts for each transaction or creates new nodes where necessary

common (and therefore not really interesting) itemsets, the frequent itemsets will usually be abundant and therefore, as a result of data exploration, not be useful either. In frequent pattern mining, this phenomenon is known as the pattern explosion. By the exponential size of possible subspaces, and type of interestingness measures, subspace clustering inherited this problem with the transfer of the techniques from frequent pattern mining. For non-trivial thresholds usually huge sets of subspace clusters are discovered—which are typically quite redundant.

Different means have been studied to condense the result set of patterns or to restrict the search space further in the first place.

One approach among others is mining or keeping only those itemsets that cannot be extended further without dropping below the threshold, i.e., the *maximal* frequent itemsets [16]. An alternative approach uses borders to represent a lossless compression of the result set of frequent patterns, named *closed* frequent itemsets [68]. Another branch of summarization is that of picking or creating a number of representative results. Yan et al. [78] choose a subset of results such that the error of predicting the frequencies in the complete result set is minimized. Mampaey et al. [57] give an information theoretic approach to identifying that subset of results by which the frequencies in either the complete result set, or the data in general, can best be approximated. To this end, they define a maximum entropy model for data objects, given knowledge about itemset frequencies. The resulting models capture the general structure of the data very well, without redundancy.

Just as the basic techniques for frequent pattern mining, also these ideas for condensing the result, as well as restricting the search space, have found corresponding solutions to the problem of redundant results in subspace clustering.

3 Frequent Pattern Mining in Subspace Clustering

3.1 Subspace Cluster Search

As mentioned above, CLIQUE [7] introduced the first subspace clustering algorithm using a monotonicity on the subspace search space. The approach uses an equal width discretization of the input space and a density threshold per cell. A subspace cluster is a maximal set of connected dense cells in some subspace. As a consequence, the approach operates also algorithmically at the cell level. The monotonicity used is that a dense cell in a k-dimensional subspace is also a dense cell in all its $k-1$ dimensional subspaces:

$$\begin{aligned} &\mathcal{C} \text{ is a cluster in subspace } \mathcal{T} \Rightarrow \\ &\mathcal{C} \text{ is part of a cluster in all subspaces } \mathcal{S} \subseteq \mathcal{T} \end{aligned} \tag{16.4}$$

Based on the corresponding anti-monotonicity, Apriori is applied from 1-dimensional dense cells in a straightforward fashion to find all higher-dimensional dense cells. As a variation of this base scheme, an approximation is suggested that prunes subspaces from consideration if their dense cells do not cover a sufficiently large part of the data.

MAFIA [65] extends the cell-based approach by adapting the cell sizes to the data distribution. The general approach is to combine neighboring cells in one dimension if they have similar density values. The monotonicity used is the same as in CLIQUE, but additionally, a parallel algorithm is introduced that processes chunks of the data on local machines that communicate to exchange cell counts at each level of the subspace lattice. XProj [5] is an adaptation of the CLIQUE idea to clustering of graph data based on frequent sub-graphs and was applied to cluster XML data. In contrast to CLIQUE, XProj looks for a hard partitioning, rather than overlapping clusters.

CLIQUE and MAFIA may miss points or subspace clusters depending on location and resolution of the cells (see for example Fig. 16.6), so later works have proposed bottom-up algorithms that do not rely on discretization. SUBCLU [48] follows the density-based subspace clustering paradigm. As already illustrated in Fig. 16.4, subspace clusters are maximal sets of density-connected points. Any subspace cluster projection to a lower dimensional subspace is a density-connected set again (albeit not necessarily a maximal one). Anti-monotonicity is used in that if a subspace does not contain a density-based subspace cluster, then no superspace will either.

Note that this approach means that the notion of frequent patterns is also different than in CLIQUE and MAFIA: in these cell-based approaches, a (frequent) item is a (dense) cell in a particular subspace, whereas in SUBCLU (and later approaches) it is the entire subspace. In SUBCLU, the Apriori principle is used to generate candidate subspaces within which the actual subspace clusters are determined.

The DUSC [10] approach relies on a different definition of density than SUBCLU does. Based on the observation that a fixed density assessment is biased and favors

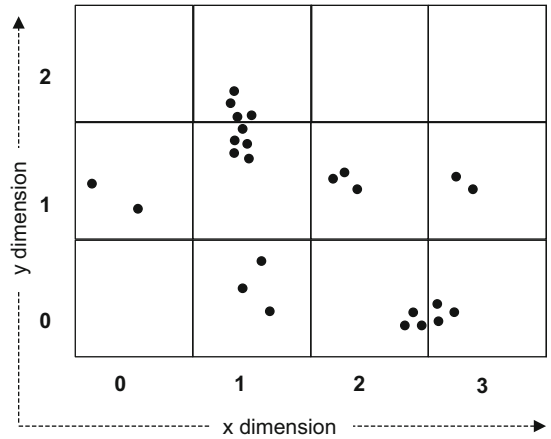


Fig. 16.6 Standard grid-based discretization as used e.g. in CLIQUE: the accuracy of subspace clustering depends on location and resolution of the grid. A minimum cell count of more than three will miss the subspace cluster at the *bottom right*, whereas a minimum cell count of three will also report cells that contain a few isolated noise points (e.g., cell at the *center right*)

low dimensional subspace clusters over high-dimensional ones, the density measure is normalized by the expected density. This means that (anti-)monotonicity is lost, and standard application of Apriori is not possible. However, as proposed in a later extension, it is possible to use the anti-monotonicity as a filtering criterion in a multistep clustering scheme (EDSC) [11]. The idea is to generate a conservative approximation of subspace clusters based on cells that are merged if potentially density-connected. Similar in spirit to the anti-monotonicity in Apriori, pruning is based on the weakest density measure as a filter step.

The idea of avoiding full lattice search in favor of more efficient runtimes (i.e., the colossal pattern idea [82] we saw above) is also found for subspace clustering [64]. Instead of analyzing all subspaces, and the entire value ranges within these subspaces, the idea is to represent subspace clusters at different levels of approximation. Using the number of objects within the current approximation as an indication, potential combinations with other subspaces are used as an indication of higher-dimensional subspace clusters. Priority queues are maintained in order to generate the most promising candidates in the lattice first. As a result, it becomes possible to avoid the generation of many relatively low-dimensional subspace clusters and to steer the search towards high-dimensional subspace clusters directly.

Another interesting connection to frequent pattern mining is discussed with the algorithm INSCY for density-based subspace clustering [12]: subspace clusters are detected based on a frequent cell count data representation, an index structure that is similar in spirit to the FP-tree from frequent itemset mining. As mentioned in the previous section, the challenge here is two-fold: first, to define an adequate representation of subspace regions (the items), and second, to identify similarities among these subspace regions. For the first part, a discretization technique as in

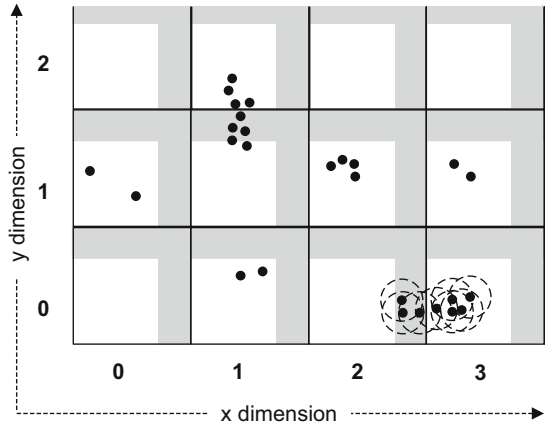


Fig. 16.7 Grid with density-preserving borders: to guarantee detection of all density-based subspace clusters, the grid is enhanced with borders (*gray shaded*) at the top of each cell in each dimension. These borders have exactly the size of the area for the density assessment (circles around points in the clusters at the *bottom right*), so that an empty border means that no cluster extends across these two cells

EDSC [11] is used, which consists of a traditional equal-width grid, plus density-preserving borders. Figure 16.7 illustrates the general idea: the density-preserving borders make it possible to determine whether points in one cell are potentially density-connected to those in a neighboring cell. They are the size of the area used for density assessment (circles around points in the figure). If a subspace cluster extends across one of these borders, this border must be non-empty. If that should be the case, these cells need to be merged during mining.

A SCY-tree is constructed, which similar to item frequency counts in FP-trees contains counts of the number of points in a particular grid cell. In addition, marker nodes are introduced to signal that the border between neighboring cells is non-empty. An example is given in Fig. 16.8. As we can see in this example, the ten points that are in the bottom '0' slice of the y-dimension (leftmost node under the root in the tree), fall into three different intervals in the x-dimension: two in cell '1', three in cell '2', and five in cell '3' (three child nodes). Additionally, a node marks the presence of one or more points in the border of cell '2' by a special node without any count information. Similar to FP-Growth, it is then possible to mine subspace clusters in a depth-first manner. Different levels of the index correspond to the dimensions in which these cells exist. As opposed to frequent itemset mining, neighboring nodes are merged if they contain cells that are potentially part of the same cluster.

3.2 Subspace Search

Subspace search based on frequent pattern mining concepts has been applied both independently of specific clustering algorithms, as well as integrated in some clustering

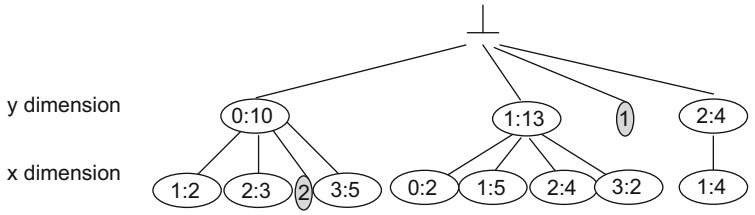


Fig. 16.8 SCY-tree index for depth-first mining of subspace clusters. Nodes contain cell counts as in frequent itemset mining. Levels correspond to different dimensions, and additional marker nodes indicate that a border is non-empty and that cells need to be merged during mining. For example, the *gray shaded node* labeled '2' at the *bottom* corresponds to the *non-empty border* of cell '2' in dimension x in Fig. 16.7

algorithm yet independent of the cluster model. In the first scenario, we can regard subspace search as a global identification of 'interesting' subspaces—subspaces in which we expect clusters to exist—and hence as a restriction of the search space. In the second scenario, we observe a local identification of 'interesting' subspaces. A typical use case of these 'locally interesting' subspaces is to adapt distance measures locally, that is, for different clusters, different measures of similarity are applied.

Global Subspace Search ENCLUS [21] is based on an assessment of the subspace as a whole, i.e., a subspace search step proceeds the actual subspace clustering. In order to determine interesting subspaces, Shannon Entropy [73] is used. Entropy measures the uncertainty in a random variable, where a high value means a high level of uncertainty. A uniform distribution implies greatest uncertainty, so a low entropy value (below some threshold) is used as an indication of subspace clusters. Similar to CLIQUE, the data are discretized into equal-width cells before entropy assessment. Monotonicity is based on the fact that an additional attribute can only increase the uncertainty and thereby the Shannon Entropy:

$$\mathcal{T} \text{ has low entropy} \Rightarrow \forall \mathcal{S} \subseteq \mathcal{T} : \mathcal{S} \text{ has low entropy.} \tag{16.5}$$

Besides this Apriori bottom-up part of the algorithm, an additional mutual information criterion is used for top-down pruning. Interesting subspaces in this sense are those with an entropy that is lower (by some threshold) than the sum of the entropy of each of its one-dimensional subspaces. Using both criteria, the most interesting subspaces for subspace clustering according to ENCLUS are located neither at the top nor at the bottom of the subspace search space, but at some medium dimensionality. This resembles the concept of borders in frequent itemset mining (Sect. 2.3). While there borders are used to derive a condensed representation of the result set, here, the result set is restricted to reduce the redundancy of too many clusters.

For RIS (Ranking Interesting Subspaces) [47], subspaces are 'interesting' if they have a large number of points in the neighborhoods of core points (i.e., points with a high local point density according to some thresholds), normalized by the expected number of points assuming uniform distribution. While this criterion adopts a density-based notion [51] of 'interesting', it is not tied to a specific clustering algorithm.

These subspaces are hence expected to prove interesting for various density-based clustering algorithms. While monotonicity of this quality criterion is not proven in general, we do know that the core point property is anti-monotonic:

$$o \text{ is not a core point in } \mathcal{S} \Rightarrow \\ \forall \mathcal{T} \supseteq \mathcal{S} : o \text{ is not a core point in } \mathcal{T}. \tag{16.6}$$

A similar approach, SURFING (SUbspaces Relevant For clusterING) [15], also following a density-based notion of 'interestingness', assesses the variance of k-nearest neighbor distances. ℓ-dimensional subspaces can be rated as 'interesting', 'neutral', or 'irrelevant' in comparison to $(\ell - 1)$-dimensional subspaces, but this rating is not monotonous. Accordingly, the subspace search of SURFING follows the Apriori idea of early pruning of candidates only heuristically but does not formally implement the strictly anti-monotonic candidate elimination.

CMI (Cumulative Mutual Information) [66] is a measure to assess the correlation among the attributes of some subspace and is used to identify subspaces that are interesting w.r.t. a high contrast, that is, they are likely to contain different clusters. The authors assume monotonicity of this contrast criterion to facilitate a candidate elimination-based search starting with two dimensional subspaces. As a priority search, generating candidates from the top m subspaces only, their algorithm is more efficient than the Apriori search at the expense of completeness of the results. Finally, subspaces contained in another subspace reported as a result are dropped from the resulting set of subspaces, if the higher-dimensional subspace also has higher contrast.

Local Subspace Search Subspace search has also been incorporated locally into clustering algorithms. DiSH [1], similar to its predecessor HiCS citeclu:AchBoeKriKroetal06, follows a pattern of cluster search that is different from the Apriori-based subspace clustering idea discussed so far. Appropriate subspaces for distance computations are learned locally for each point, then the locally adapted (subspace-) distances and the dimensionality of the assigned subspace are used as a combined distance measure in a global clustering schema similar to OPTICS [8] to find hierarchies of subspaces.

For learning the most appropriate subspace for each data point both HiCS and DiSH assign a 'subspace preference vector' to each object, based on the variance of the neighborhood in each attribute. As such, the clustering procedure does not make use of an efficient frequent pattern search algorithm. However, while HiCS uses the full-dimensional neighborhood and studies the variances of the neighborhoods in attribute-wise projections, DiSH starts with attribute-wise neighborhoods and combines those neighborhoods in a bottom-up procedure. Here, an Apriori-like search strategy is one of the suggested alternatives, employing the monotonicity of neighborhoods in projections of the data. If $\mathcal{S}$ is a subspace of $\mathcal{T}$, then the cardinality of the ε-neighborhood of some object o in $\mathcal{T}$ is bound to be at most the cardinality of the ε-neighborhood of the same object o in $\mathcal{S}$:

$$\mathcal{S} \subseteq \mathcal{T} \Rightarrow \left|\mathcal{N}_\varepsilon^{\mathcal{T}}(o)\right| \leq \left|\mathcal{N}_\varepsilon^{\mathcal{S}}(o)\right| \tag{16.7}$$

This holds, e.g., for L_P-type distances ($P \geq 1$, for example the commonly used Euclidean distance), because distances between the points can never shrink when adding more dimensions. Let us note that this is also the reason why the core point property is anti-monotone (cf. Eq. 16.6).

In a similar way, CFPC [80] ('MineClus' in an earlier version [79]) improves by a frequent pattern mining-based approach the subspace search strategy of an earlier projected clustering algorithm, DOC [71]. As projected clustering approaches, both pursue a local subspace search per (preliminary) cluster. A typical projected clustering algorithm, following the seminal approach of PROCLUS [3], starts with some initial assignment of points to clusters. Then, the optimal projection (subspace) of each cluster and the assignment of points are iteratively refined. In DOC, random-sampling was applied to find the most suitable subspace for a potential cluster. CFPC replaces this random sampling strategy by a technique related to FP-growth. A potential cluster is defined by its potential (in both approaches randomly sampled) medoid p. For all points q, an itemset includes those dimensions in which q is close to p. A large, frequent itemset would therefore correspond to a projected cluster with many points and high dimensionality. To find the best cluster and its optimal projection, FP-growth is applied over this modelling of frequent itemsets.

The projected clustering algorithm P3C [58, 59] does also incorporate an Apriori-like local subspace search, but in yet another variant. The basic idea of P3C is to find cluster cores starting with "p-signatures" that are intervals of some subset of p distinct attributes, i.e., subspace regions. Roughly, such a p-signature qualifies as a cluster core if and only if its support, i.e., the number of points falling into this subspace region, exceeds the expected support under some assumptions concerning the point distribution, and if this happens by chance (Poisson probability) less likely than specified by some (Poisson-)threshold. By these conditions, p-signatures qualifying as cluster cores can be generated using an Apriori-like candidate elimination procedure.

3.3 Redundancy in Subspace Clustering

As pointed out above, redundancy of subspace cluster results is a problem inherited from the Apriori strategy for traversing the search space of subspaces. As a consequence, for current research on subspace clustering, reducing redundancy is a major topic. As we have seen, the concept of borders found analogous use already in the early subspace search algorithm ENCLUS [21] for restricting the search space. Some approaches mine or report the most representative clusters as solutions [13, 61]. This is related to picking or creating a number of representative results in frequent pattern mining. Also the idea of restricting results of frequent pattern mining to the maximal frequent itemsets found a correspondence in subspace clustering. For example, nCluster [54], CLICKS [81], or MaPle [69] mine those subspace clusters of maximal dimensionality.

Other variants of clustering algorithms outside subspace clustering that also tackle high-dimensional data face a similar problem. For example, multiview clustering

[19, 22, 34, 46] approaches the problem from the opposite direction. It is based on the notion of semantically different subspaces, i.e., multiple representations for the same data. We cannot generally assume to know the different semantics of subspaces beforehand and, accordingly, could find results in overlapping subspaces. As a consequence, these approaches allow some redundancy between resulting clusters. A certain partial overlap between concepts is allowed in order to not exclude possibly interesting concepts.

A related though distinct way of addressing the problem of redundancy and distinctiveness of different clusters is to seek diverse clusterings by directly assessing a certain notion of distance between different partitions (so-called alternative clustering approaches [14, 23, 24, 25, 32, 33, 35, 70, 72]). Starting with one clustering solution, they search for an alternative clustering solution that provides substantially different insights. Still, alternative clustering solutions are allowed to not be absolutely orthogonal but to show some redundancy with existing clustering solutions.

Apparently, to avoid redundancy as more 'enhanced' [74] subspace clustering algorithms try to do should not be pursued as an absolute goal. Multiview clustering and alternative clustering come from the other extreme and relax the original restriction of 'no redundancy' more and more. Relationships between subspace clustering and other families of clustering approaches have been discussed by Zimek and Vreeken [84].

A question related to the redundancy issue is that of the appropriate density level. Both of these issues have decisive influence on the clusters that are selected. Determining the right density level is a general problem also in full space density-based clustering [51], but for clustering in subspaces, the problem is even more severe. Setting a fixed density threshold for an Apriori style subspace search is not appropriate for all possible subspaces. Consider for example any CLIQUE-style grid approach: the volume of a hypercube increases exponentially with the dimensionality, hence the density decreases rapidly. As a consequence, any chosen threshold introduces a bias to identify clusters of (up to) a certain dimensionality. This observation motivates research on adaptive density thresholds [10, 62]. The algorithmic challenge then comes from loss of monotonicity that would allow efficient traversal of the search space of subspaces.

When using Euclidean distance (L_2), the appropriate choice of an ε-range becomes extremely challenging as well due to the rather counter-intuitive behavior of the volume of the hypersphere with increasing dimensions. Let us note that, for outlier detection, the very same problem occurs in high-dimensional data, which has been discussed in detail by Zimek et al. [85]. Choosing the size of the neighborhood in terms of objects rather than in terms of a radius (i.e., using k nearest neighbors instead of an ε-range query) has been advocated as a workaround for this problem [2], to solve at least certain aspects such as having a well-defined (non-empty) set of objects for the density estimation or spatial properties of the neighborhood.

4 Conclusions

This chapter discusses the close relationship between frequent pattern mining and clustering, which might not be apparent at first sight. In fact, frequent pattern mining was the godfather of subspace clustering, which developed quickly into an independent and influential research area on its own. We showed how certain techniques that have been originally developed for frequent pattern mining have been transferred to clustering, how these techniques changed in their new environment, and how the drawbacks of these techniques—unfortunately transferred along—raised new research questions as well as interesting solutions in the area of data clustering.

Acknowledgments Ira Assent is partly supported by the Danish Council for Independent Research—Technology and Production Sciences (FTP), grant 10-081972. Jilles Vreeken is supported by the Cluster of Excellence 'Multimodal Computing and Interaction' within the Excellence Initiative of the German Federal Government.

References

1. E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, I. Müller-Gorman, and A. Zimek. Detection and visualization of subspace cluster hierarchies. In *12th International Conference on Database Systems for Advanced Applications (DASFAA), Bangkok, Thailand*, pages 152–163, 2007.
2. E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek. Robust, complete, and efficient correlation clustering. In *7th SIAM International Conference on Data Mining (SDM), Minneapolis, MN*, pages 413–418, 2007.
3. C. C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *ACM International Conference on Management of Data (SIGMOD), Philadelphia, PA*, pages 61–72, 1999.
4. C. C. Aggarwal, A. Hinneburg, and D. Keim. On the surprising behavior of distance metrics in high dimensional space. In *8th International Conference on Database Theory (ICDT), London, UK*, pages 420–434, 2001.
5. C. C. Aggarwal, N. Ta, J. Wang, J. Feng, and M. Zaki. Xproj: a framework for projected structural clustering of xml documents. In *13th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Jose, CA*, pages 46–55, 2007.
6. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *20th International Conference on Very Large Data Bases (VLDB), Santiago de Chile, Chile*, pages 487–499, 1994.
7. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *ACM International Conference on Management of Data (SIGMOD), Seattle, WA*, pages 94–105, 1998.
8. M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. In *ACM International Conference on Management of Data (SIGMOD), Philadelphia, PA*, pages 49–60, 1999.
9. I. Assent. Clustering high dimensional data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4):340–350, 2012.
10. I. Assent, R. Krieger, E. Müller, and T. Seidl. DUSC: dimensionality unbiased subspace clustering. In *7th IEEE International Conference on Data Mining (ICDM), Omaha, NE*, pages 409–414, 2007.
11. I. Assent, R. Krieger, E. Müller, and T. Seidl. EDSC: efficient density-based subspace clustering. In *17th ACM Conference on Information and Knowledge Management (CIKM), Napa Valley, CA*, pages 1093–1102, 2008.

12. I. Assent, R. Krieger, E. Müller, and T. Seidl. INSCY: indexing subspace clusters with in-process-removal of redundancy. In *8th IEEE International Conference on Data Mining (ICDM), Pisa, Italy*, pages 719–724, 2008.
13. I. Assent, E. Müller, S. Günnemann, R. Krieger, and T. Seidl. Less is more: Non-redundant subspace clustering. In *MultiClust: 1st International Workshop on Discovering, Summarizing and Using Multiple Clusterings Held in Conjunction with KDD 2010, Washington, DC*, 2010.
14. E. Bae and J. Bailey. COALA: a novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *6th IEEE International Conference on Data Mining (ICDM), Hong Kong, China*, pages 53–62, 2006.
15. C. Baumgartner, K. Kailing, H.-P. Kriegel, P. Kröger, and C. Plant. Subspace selection for clustering high-dimensional data. In *4th IEEE International Conference on Data Mining (ICDM), Brighton, UK*, pages 11–18, 2004.
16. R. Bayardo. Efficiently mining long patterns from databases. In *ACM International Conference on Management of Data (SIGMOD), Seattle, WA*, pages 85–93, 1998.
17. K. P. Bennett, U. Fayyad, and D. Geiger. Density-based indexing for approximate nearest-neighbor queries. In *5th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Diego, CA*, pages 233–243, 1999.
18. K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *7th International Conference on Database Theory (ICDT), Jerusalem, Israel*, pages 217–235, 1999.
19. S. Bickel and T. Scheffer. Multi-view clustering. In *4th IEEE International Conference on Data Mining (ICDM), Brighton, UK*, pages 19–26, 2004.
20. R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *17th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Gold Coast, Australia*, pages 160–172, 2013.
21. C. H. Cheng, A. W.-C. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *5th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Diego, CA*, pages 84–93, 1999.
22. Y. Cui, X. Z. Fern, and J. G. Dy. Non-redundant multi-view clustering via orthogonalization. In *7th IEEE International Conference on Data Mining (ICDM), Omaha, NE*, pages 133–142, 2007.
23. X. H. Dang and J. Bailey. Generation of alternative clusterings using the CAMI approach. In *10th SIAM International Conference on Data Mining (SDM), Columbus, OH*, pages 118–129, 2010.
24. I. Davidson and Z. Qi. Finding alternative clusterings using constraints. In *8th IEEE International Conference on Data Mining (ICDM), Pisa, Italy*, pages 773–778, 2008.
25. I. Davidson, S. S. Ravi, and L. Shamis. A SAT-based framework for efficient constrained clustering. In *10th SIAM International Conference on Data Mining (SDM), Columbus, OH*, pages 94–105, 2010.
26. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 39(1):1–31, 1977.
27. R. J. Durrant and A. Kaban. When is 'nearest neighbour' meaningful: A converse theorem and implications. *Journal of Complexity*, 25(4):385–397, 2009.
28. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *2nd ACM International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR*, pages 226–231, 1996.
29. I. Färber, S. Günnemann, H.-P. Kriegel, P. Kröger, E. Müller, E. Schubert, T. Seidl, and A. Zimek. On using class-labels in evaluation of clusterings. In *MultiClust*: 1st International Workshop on Discovering, Summarizing and Using Multiple Clusterings Held in Conjunction with KDD 2010, Washington, DC, 2010.
30. D. François, V. Wertz, and M. Verleysen. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19(7):873–886, 2007.
31. G. Gan, C. Ma, and J. Wu. *Data Clustering. Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics (SIAM), 2007.

32. D. Gondek and T. Hofmann. Non-redundant data clustering. In *4th IEEE International Conference on Data Mining (ICDM), Brighton, UK*, pages 75–82, 2004.
33. D. Gondek and T. Hofmann. Non-redundant clustering with conditional ensembles. In *11th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Chicago, IL*, pages 70–77, 2005.
34. S. Günnemann, E. Müller, I. Färber, and T. Seidl. Detection of orthogonal concepts in subspaces of high dimensional data. In *18th ACM Conference on Information and Knowledge Management (CIKM), Hong Kong, China*, pages 1317–1326, 2009.
35. S. Günnemann, I. Färber, E. Müller, and T. Seidl. ASCLU: alternative subspace clustering. In *MultiClust: 1st International Workshop on Discovering, Summarizing and Using Multiple Clusterings Held in Conjunction with KDD 2010, Washington, DC*, 2010.
36. S. Günnemann, I. Färber, E. Müller, I. Assent, and T. Seidl. External evaluation measures for subspace clustering. In *20th ACM Conference on Information and Knowledge Management (CIKM), Glasgow, UK*, pages 1363–1372, 2011.
37. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *ACM SIGMOD Record*, 29(2):1–12, 2000.
38. J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3rd edition, 2011.
39. J. A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, New York, London, Sydney, Toronto, 1975.
40. A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *4th ACM International Conference on Knowledge Discovery and Data Mining (KDD), New York City, NY*, pages 58–65, 1998.
41. A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What is the nearest neighbor in high dimensional spaces? In *26th International Conference on Very Large Data Bases (VLDB), Cairo, Egypt*, pages 506–515, 2000.
42. M. E. Houle, H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Can shared-neighbor distances defeat the curse of dimensionality? In *22nd International Conference on Scientific and Statistical Database Management (SSDBM), Heidelberg, Germany*, pages 482–500, 2010.
43. A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.
44. A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, 1988.
45. A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.
46. P. Jain, R. Meka, and I. S. Dhillon. Simultaneous unsupervised learning of disparate clusterings. *Statistical Analysis and Data Mining*, 1(3):195–210, 2008.
47. K. Kailing, H.-P. Kriegel, P. Kröger, and S. Wanka. Ranking interesting subspaces for clustering high dimensional data. In *7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), Cavtat-Dubrovnik, Croatia*, pages 241–252, 2003.
48. K. Kailing, H.-P. Kriegel, and P. Kröger. Density-connected subspace clustering for high-dimensional data. In *4th SIAM International Conference on Data Mining (SDM), Lake Buena Vista, FL*, pages 246–257, 2004.
49. L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analyis*. John Wiley & Sons, 1990.
50. H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1–58, 2009.
51. H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.
52. H.-P. Kriegel, P. Kröger, and A. Zimek. Subspace clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4):351–364, 2012.
53. P. Kröger and A. Zimek. Subspace clustering techniques. In L. Liu and M. T. Ozsu, editors, *Encyclopedia of Database Systems*, pages 2873–2875. Springer, 2009.

54. G. Liu, J. Li, K. Sim, and L. Wong. Distance based subspace clustering with flexible dimension partitioning. In *23rd International Conference on Data Engineering (ICDE), Istanbul, Turkey*, pages 1250–1254, 2007.
55. S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982.
56. J. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematics, Statistics, and Probabilistics*, volume 1, pages 281–297, 1967.
57. M. Mampaey, N. Tatti, and J. Vreeken. Tell me what I need to know: Succinctly summarizing data with itemsets. In *17th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), San Diego, CA*, pages 573–581, 2011.
58. G. Moise, J. Sander, and M. Ester. P3C: A robust projected clustering algorithm. In *6th IEEE International Conference on Data Mining (ICDM), Hong Kong, China*, pages 414–425, 2006.
59. G. Moise, J. Sander, and M. Ester. Robust projected clustering. *Knowledge and Information Systems (KAIS)*, 14(3):273–298, 2008.
60. G. Moise, A. Zimek, P. Kröger, H.-P. Kriegel, and J. Sander. Subspace and projected clustering: Experimental evaluation and analysis. *Knowledge and Information Systems (KAIS)*, 21(3):299–326, 2009.
61. E. Müller, I. Assent, S. Günnemann, R. Krieger, and T. Seidl. Relevant subspace clustering: Mining the most interesting non-redundant concepts in high dimensional data. In *9th IEEE International Conference on Data Mining (ICDM), Miami, FL*, pages 377–386, 2009.
62. E. Müller, I. Assent, R. Krieger, S. Günnemann, and T. Seidl. Dens-Est:density estimation for data mining in high dimensional spaces. In *9th SIAM International Conference on Data Mining (SDM), Sparks, NV*, pages 173–184, 2009.
63. E. Müller, S. Günnemann, I. Assent, and T. Seidl. Evaluating clustering in subspace projections of high dimensional data. In *35th International Conference on Very Large Data Bases (VLDB), Lyon, France*, pages 1270–1281, 2009.
64. E. Müller, I. Assent, S. Günnemann, and T. Seidl. Scalable densitybased subspace clustering. In *20th ACM Conference on Information and Knowledge Management (CIKM), Glasgow, UK*, pages 1077–1086, 2011.
65. H. S. Nagesh, S. Goil, and A. Choudhary. Adaptive grids for clustering massive data sets. In *1st SIAM International Conference on Data Mining (SDM), Chicago, IL*, 2001.
66. H. V. Nguyen, E. Müller, J. Vreeken, F. Keller, and K. Böhm. CMI: an information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In *13th SIAM International Conference on Data Mining (SDM), Austin, TX*, pages 198–206, 2013.
67. L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: A review. *ACM SIGKDD Explorations*, 6(1):90–105, 2004.
68. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *7th International Conference on Database Theory (ICDT), Jerusalem, Israel*, pages 398–416, 1999.
69. J. Pei, X. Zhang, M. Cho, H. Wang, and P. S. Yu. MaPle: A fast algorithm for maximal pattern-based clustering. In *3rd IEEE International Conference on Data Mining (ICDM), Melbourne, FL*, pages 259–266, 2003.
70. J. M. Phillips, P. Raman, and S. Venkatasubramanian. Generating a diverse set of high-quality clusterings. In *2nd MultiClust Workshop: Discovering, Summarizing and Using Multiple Clusterings Held in Conjunction with ECML PKDD 2011, Athens, Greece*, pages 80–91, 2011.
71. C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali. A Monte Carlo algorithm for fast projective clustering. In *ACM International Conference on Management of Data (SIGMOD), Madison, WI*, pages 418–427, 2002.
72. Z. J. Qi and I. Davidson. A principled and flexible framework for finding alternative clusterings. In *15th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Paris, France*, pages 717–726, 2009.

73. C. E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.
74. K. Sim, V. Gopalkrishnan, A. Zimek, and G. Cong. A survey on enhanced subspace clustering. *Data Mining and Knowledge Discovery*, 26(2):332–397, 2013.
75. P. H. A. Sneath. The application of computers to taxonomy. *Journal of General Microbiology*, 17:201–226, 1957.
76. M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. In *8th International Work-Conference on Artificial Neural Networks (IWANN), Barcelona, Spain*, pages 758–770, 2005.
77. D. Wishart. Mode analysis: A generalization of nearest neighbor which reduces chaining effects. In A. J. Cole, editor, *Numerical Taxonomy*, pages 282–311, 1969.
78. X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: a profile-based approach. In *11th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), Chicago, IL*, pages 314–323, 2005.
79. M. L. Yiu and N. Mamoulis. Frequent-pattern based iterative projected clustering. In *3rd IEEE International Conference on Data Mining (ICDM), Melbourne, FL*, pages 689–692, 2003.
80. M. L. Yiu and N. Mamoulis. Iterative projected clustering by subspace mining. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):176–189, 2005.
81. M. J. Zaki, M. Peters, I. Assent, and T. Seidl. CLICKS: an effective algorithm for mining subspace clusters in categorical datasets. *Data & Knowledge Engineering*, 60(1):51–70, 2007.
82. F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Cheng. Mining colossal frequent patterns by core pattern fusion. In *23rd International Conference on Data Engineering (ICDE), Istanbul, Turkey*, pages 706–715, 2007.
83. A. Zimek. Clustering high-dimensional data. In C. C. Aggarwal and C. K. Reddy, editors, *Data Clustering: Algorithms and Applications*, chapter 9, pages 201–230. CRC Press, 2013.
84. A. Zimek and J. Vreeken. The blind men and the elephant: On meeting the problem of multiple truths in data from clustering and pattern mining perspectives. *Machine Learning*, 2013.
85. A. Zimek, E. Schubert, and H.-P. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5):363–387, 2012.