Subspace Clustering for Complex Data

Stephan Günnemann

Data Management and Data Exploration Group
RWTH Aachen University, Germany
guennemann@cs.rwth-aachen.de

Abstract: Clustering is an established data mining technique for grouping objects based on their mutual similarity. Since in today's applications, however, usually many characteristics for each object are recorded, one cannot expect to find similar objects by considering all attributes together. In contrast, valuable clusters are hidden in subspace projections of the data. As a general solution to this problem, the paradigm of *subspace clustering* has been introduced, which aims at automatically determining for each group of objects a set of relevant attributes these objects are similar in.

In this work, we introduce novel methods for effective subspace clustering on various types of complex data: vector data, imperfect data, and graph data. Our methods tackle major open challenges for clustering in subspace projections. We study the problem of redundancy in subspace clustering results and propose models whose solutions contain only non-redundant and, thus, valuable clusters. Since different subspace projections represent different views on the data, often several groupings of the objects are reasonable. Thus, we propose techniques that are not restricted to a single partitioning of the objects but that enable the detection of multiple clustering solutions.

1 Introduction

The increasing potential of storage technologies and information systems over the last decades has opened the possibility to conveniently and affordably gather large amounts of complex data. Going beyond simple descriptions of objects by some few characteristics, such data sources range from high dimensional vector spaces over imperfect data containing errors to network data describing relations between the objects. While storing these data is common, their analysis is challenging: the human capabilities of a manual analysis are quickly exhausted considering the mere size of the data. Thus, automatic techniques supporting the user in the process of knowledge extraction are required to gain a benefit from the collected data.

The concept of Knowledge Discovery in Databases (KDD) [HK01] has been evolved as a possible solution for the above challenge and it is coherently described by a multilevel process the user has to follow (cf. Figure 1). Given the raw data, which is rarely perfect since, e.g., missing entries, inconsistencies, or uncertain values are prevalent during the data acquisition phase, the KDD process starts with a preprocessing step to clean the data. This step is often referred to data cleansing and tries to increase the data quality to support the subsequent data mining step. The goal of data mining, as the key component of the

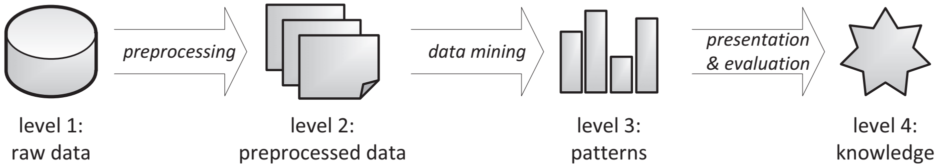


Figure 1: Knowledge Discovery in Databases (KDD) process

KDD process, is to extract previously unknown and useful patterns from the data using automatic or semi-automatic algorithms. Finally, the KDD process concludes with the presentation and evaluation of the detected patterns, enabling the user to understand and interpret the results.

In this work we focus on the development of novel models and algorithms for the central step of the KDD process: data mining. Out of the several mining tasks that exist in the literature, this work centers on the important method of *clustering*, which aims at grouping similar objects while separating dissimilar ones. Clustering, as an unsupervised learning task, analyses data without given labels but automatically reveals the hidden structure of the data by its aggregations. For today's data, however, it is known that traditional clustering methods fail to detect meaningful patterns. The problem originates from the fact that traditional clustering approaches consider the full space to measure the similarity between objects, i.e. all characteristics of the objects are taken into account. While collecting more and more characteristics, however, it is very unlikely that two objects are similar with respect to the full space and often some dimensions are not relevant for clustering. A continuative aspect is the decreasing discrimination power of distance functions with increasing dimensionality of the data space due to the "curse of dimensionality" [BGRS99]. The distances between objects grow more and more alike, thus all objects seem equally similar based on their attribute values. Since clusters are strongly obfuscated by irrelevant dimensions and distances are not discriminable any more, searches in the full space are futile or lead to very questionable clustering results.

Global dimensionality reduction techniques, e.g., based on Principal Component Analysis (PCA [Jol02]), try to mitigate these effects, but they do not provide a solution to this problem. Since they reduce all objects to a single projection, they cannot detect clusters with locally relevant dimensions. In complex data sets, however, different groups of objects may have different relevant dimensions. In Figure 2, the objects depicted as rectangles are similar in a 2-dimensional subspace, while the objects depicted as triangles show only similar values in a single dimension.

As a solution to this problem, the paradigm of *subspace clustering* [PHL04, KKZ09] has been introduced. Subspace clustering detects clusters in arbitrary subspace projections of the data by automatically determining for each group of objects a set of relevant dimensions these objects are similar in. Thus, in Figure 2 the objects grouped in cluster C_1 would correspond to a subspace cluster in subspace $\{$fast food consumption, sport activity$\}$, while the cluster C_2 is only located in subspace $\{$sport activity$\}$. Since different subspaces may lead to different groupings, each object can naturally belong to multiple clusters as illus-

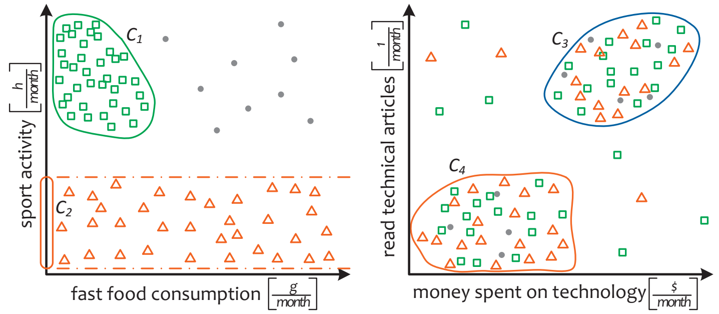


Figure 2: Exemplary subspace clustering of a 4-dimensional database

trated in Figure 2 (right). The subspaces individually assigned to each group provide the reasoning why such multiple solutions are meaningful. Thus, in the example of Figure 2, each of the four clusters $\{C_1, \dots, C_4\}$ is useful and should be provided to the user.

In this work we describe novel methods for effective subspace clustering on complex data including high-dimensional vector spaces (Section 2), imperfect data (Section 3), and graph data (Section 4). Such clustering methods are beneficial for various applications: In customer and social network analysis, persons can be grouped according to their similarity based on some product relevant attributes. In bioinformatics, groups of genes that show similar expression levels in a subset of experimental medical treatments can be identified. In sensor network analysis, different environmental events can be described by similarly behaving sensors with respect to specific measured variables. For all of these domains objects are characterized by many attributes, while the clusters appear only in subspace projections of the data.

2 Subspace Clustering on Vector Data

In high-dimensional vector spaces, clusters rarely show up in the full dimensional space but are hidden in subspace projections of the data. Subspace clustering methods try to detect these patterns by analyzing arbitrary subspaces of the data for their clustering structure. In general, a subspace cluster $C = (O, S)$ is defined by a set of objects $O \subseteq DB$ that are similar in a subset of dimensions $S \subseteq Dim$.

Traditional subspace clustering approaches report clusters in any possible subspace projection. However, besides the high computational demand due to the exponential number of subspaces w.r.t. the number of dimensions that have to be analyzed, this principle generates results with a tremendous amount of redundant clusters [MGAS09]: often the objects grouped in a cluster $C = (O, S)$ are also similar in the subspace projections $S' \subseteq S$. In Figure 2 for example, the objects of the 2-dimensional subspace cluster C_1 are also similar in the 1-dimensional projections {sport activity} and {fast food consumption}, resulting in already three clusters. Most of these groups, though, do not provide novel knowledge about the data's structure. Even worse, such redundant information hinders an easy interpretation of the mining result. Consequently, traditional subspace clustering approaches fail to detect only the relevant subspace clusters.

To tackle the above challenge we propose novel subspace clustering methods avoiding redundant information in the final clustering result. In contrast to existing approaches that simply exclude lower dimensional projections of clusters, our methods perform an optimization of the final clustering to select the most interesting clusters. Furthermore, unlike to projected clustering methods, which avoid redundancy by enforcing disjoint clusters, our methods allow overlapping clusters in general.

2.1 Subspace Clustering using Combinatorial Optimization

In one line of research, we exploit the principle of combinatorial optimization to detect non-redundant subspace clustering results. The general idea can be described as follows: Assuming the set *All* of all possible subspace clusters according to a specific cluster definition is given (e.g. the set of clusters when applying DBSCAN [EKSX96] to any subspace projection). Since this set, however, contains highly redundant clusters, we are only interested in finding an *optimal*, *non-redundant* subset $M \subseteq \mathit{All}$ of clusters. To formally define the set M, we have to specify an appropriate *objective function* which should be minimized or maximized and necessary *constraints* that need to be fulfilled by M.

In our RESCU approach [MAG+09] we extend the Set Cover optimization problem to handle subspace clustering. The basic idea is that each cluster $C \in M$ of a non-redundant clustering $M \subseteq \mathit{All}$ needs to cover sufficiently many objects not contained in other clusters. That is, we do not include clusters whose grouped objects are already represented by the remaining clusters. To realize this aim, RESCU introduces the notion of cluster gain:

Definition 1 *Cluster gain*
Let $M = \{(O_1, S_1), \ldots, (O_n, S_n)\}$ be a clustering, $C = (O, S)$ a subspace cluster, and k a cost function for subspace clusters. The cluster gain of cluster C w.r.t. to M is:

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

where $Cov(M) = \bigcup_{i=1}^{n} O_i$ are the objects covered by the clustering M.

The cost function k flexibly models the (un-)interestingness of clusters and can be specified by the user. For example, high-dimensional clusters are often be regarded as more interesting and therefore might get lower cost values. For a cluster to be included in the final result the cluster gain according to Definition 1 needs to be sufficiently high. Two important aspects contribute to this fact. First, the cluster covers many new objects, i.e. only few objects are already contained in other clusters. And second, the cost of the cluster is low, i.e. the cluster is interesting according to the user's rating. Based on the above definition, the optimal clustering M as specified in the RESCU model is defined as:

Definition 2 *Relevant subspace clustering (RESCU)*
Given the set All of all possible subspace clusters and a minimal cluster gain $\Delta \in \mathbb{R}^{\geq 0}$*,* $M \subseteq$ *All is the optimal, non-redundant clustering if and only if*

- *constraints:*
  - *M is redundancy-free, i.e.* $\forall C \in M : clus\_gain(C, M \backslash \{C\}) > \Delta$
  - *M is concept-covering, i.e.* $\forall C \in All \backslash M : clus\_gain(C, M) \leq \Delta$
- *objective: M minimizes the relative cost of the clustering, i.e. for all redundancy-free and concept-covering clusterings $N \subseteq$ All it holds*

$$\frac{1}{|Cov(M)|} \sum_{(O_i, S_i) \in M} k(O_i, S_i) \leq \frac{1}{|Cov(N)|} \sum_{(O'_i, S'_i) \in N} k(O'_i, S'_i)$$

The above *constraints* ensure that the optimal clustering M contains all but only non-redundant clusters. By minimizing the *objective function*, the best clustering according to the selected interesting criterion is chosen. Overall, based on this combinatorial optimization problem a small set of interesting and non-redundant clusters is determined.

In [MAG$^+$09] we prove that the computation of the RESCU model is NP-hard, and we propose an algorithm determining an approximate solution showing high clustering accuracy. Thorough experiments demonstrate that RESCU reliably outperforms existing subspace and projected clustering algorithms. Figure 3 shows the clustering quality (computed via the F1 and accuracy measure [GFM$^+$11]) for six real world data sets [FA10, AKMS08]. In addition to the absolute values we note the relative quality compared to the best measurement on each data set. Best 95% results are highlighted in gray. RESCU achieves top quality results for *all* data sets with respect to *both* quality measures. The competing approaches show highly varying performance. None of them achieves top quality allover. Although some of the approaches achieve slightly better results on some of the data sets, RESCU reliably shows top results on all data sets.

Detecting Multiple Clustering Views. Eliminating redundancy from subspace clustering results has to be regarded carefully: overlapping clusters are not necessarily a sufficient criterion for redundancy. Since different subspaces represent different views on the data, objects are allowed to be contained in several clusters without inducing redundancy (cf. Figure 2). The subspace clusters of each view provide novel information about the data's characteristic, and their grouping into views enables further interpretations about the clusters' interrelations. These aspects are considered in our OSCLU model [GMFS09]. In the OSCLU model we propose a global optimization of the overall clustering exploiting the clusters' similarities regarding their sets of objects as well as their subspace projections. The combinatorial optimization method of OSCLU actively includes novel knowledge of (almost) orthogonal subspaces into the final clustering result. Therefore, it overcomes major limitations of existing approaches in the detection of multiple views. The formal definition of the combinatorial optimization performed by OSCLU can be found in [GMFS09].

While the OSCLU model provides a general and flexible solution to detect subspace clusters hidden in multiple views, we prove its complexity to be NP-hard and propose an efficient algorithm to compute an approximate solution. We approximate the optimization

	Glass (214; 9)				Vowel (990; 10)				Diabetes (768; 8)			
	F1		Accuracy		F1		Accuracy		F1		Accuracy	
RESCU	60	100%	62	100%	44	100%	64	96%	71	100%	69	100%
INSCY	56	93%	54	87%	37	84%	67	100%	58	82%	65	94%
FIRES	30	50%	49	79%	10	23%	12	18%	33	46%	65	94%
SCHISM	45	75%	49	79%	24	55%	53	79%	69	97%	69	100%
PROCLUS	39	65%	54	87%	32	73%	30	45%	44	62%	65	94%
P3C	17	28%	39	63%	8	18%	16	24%	44	62%	65	94%
STATPC	19	32%	47	76%	17	39%	47	70%	39	55%	64	93%

	Shape (160; 17)				Liver-Dis. (345; 6)				Breast (198; 33)			
	F1		Accuracy		F1		Accuracy		F1		Accuracy	
RESCU	60	100%	75	100%	62	97%	61	98%	67	100%	76	97%
INSCY	56	93%	61	81%	62	97%	59	95%	65	97%	70	90%
FIRES	56	93%	62	83%	50	78%	53	85%	46	69%	75	96%
SCHISM	38	63%	59	79%	64	100%	58	94%	65	97%	71	91%
PROCLUS	60	100%	62	83%	46	72%	62	100%	47	70%	77	99%
P3C	39	65%	45	60%	36	56%	58	94%	63	94%	77	99%
STATPC	31	52%	62	83%	57	89%	58	94%	41	61%	78	100%

Figure 3: Quality (F1 & accuracy) on real world data. Captions: data set (size; dimensionality)

problem by pruning similar subspaces ensuring efficient cluster detection since only orthogonal subspaces are analyzed. Overall, our OSCLU approach is the first method for detecting multiple clustering views in subspaces of high dimensional data.

2.2 Subspace Clustering using Bayesian Generative Models

Besides combinatorial optimization, we analyzed a second line of research for multi-view subspace clustering: in [GFS12] we propose a method exploiting the principle of Bayesian generative models. We extend the established concept of mixture models to handle data containing multiple clustering views. Our MVGen model represents the data's multiple views by different subspace projections, thus, avoiding the problem of full-space clustering. Each view describes an individual partitioning of the objects. Accordingly, our model is able to represent multiple groupings and it simultaneously prevents redundant information since highly overlapping clusters in similar subspace projections are avoided. Figure 4 shows an exemplary result as reported by our method: In the example, two different clustering views are detected. The first grouping is found in the subspace $\{1, 2\}$, while a second and completely different grouping is detected in subspace $\{3, 4\}$. For each of these views an individual mixture distribution is fitted to the data. Please note that our method automatically detects the groupings as well as the dimensions supporting this grouping. Additionally, our method allows each mixture component to be located in an individual subspace. For example, as shown in Figure 4 left, the mixture component in the back is noisy in dimension 1, while the component in the front is located in both dimensions.

In [GFS12], we formally introduce the generative process that models data containing multiple views. Since views and clusters are located in subspace projections, we distinguish between relevant and irrelevant dimensions. Thus, unlike to traditional mixture model, in our model we have to tackle the challenge of model selection. In our method, we use

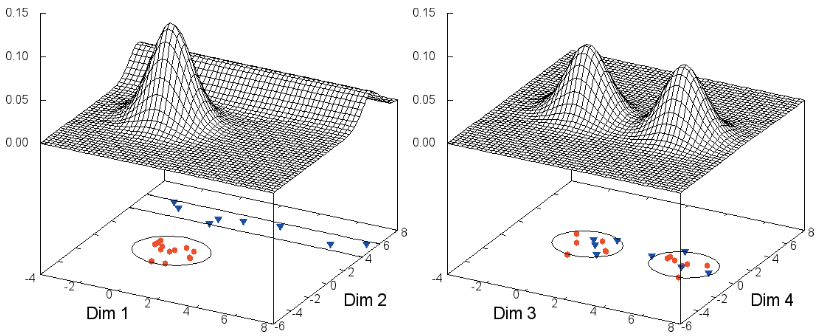


Figure 4: Mixture models located in subspace projections

Bayesian model selection to decide which sets of dimensions are relevant for the clusters and views. For an efficient learning, we exploit the principle of iterated conditional modes and we derived the required update equations.

The comparison of MVGen with competing approaches demonstrated the strengths of detecting views in multiple subspace projections. In the following we exemplarily show the results of MVGen on two datasets. In the first experiment we analyze the clustering result on the CMUFace data. This data is interesting for multi-view clustering since it consists of images taken from persons showing varying characteristics as their facial expressions (neutral, happy, sad, angry), head positions (left, right, straight), and eye states (open, sunglasses). We randomly select 3 persons with all their images and applied PCA retaining at least 90% of the data's variance as a pre-processing step. The clustering result of MVGen for two views each with three clusters is illustrated in Figure 5. The images correspond to the means of each detected cluster. By visual inspection, we can easily find the reason for detecting these two views: The first view, describes the grouping based on the different persons, while the second view, corresponds to a grouping based on their head positions.

In the second experiment, we perform image segmentation on Escher images, which are known to have multiple interpretations to the human eye. For clustering, each pixel is

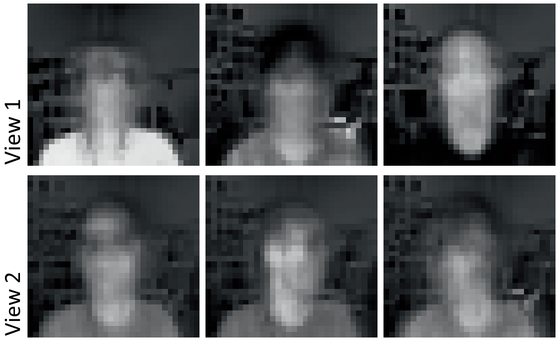
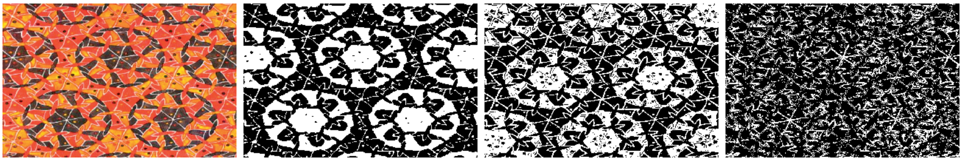


Figure 5: MVGen on face data



Original image View 1 View 2 View 3

Figure 6: Result of MVGen on an Escher image

regarded as an object with RGB and HSV values as features. In Figure 6 (left), such an image is depicted (followed by the three views detected by MVGen). Focusing on the dark regions, there is a segmentation of the image as given by the first view of MVGen. This segmentation is dominant since the dark parts clearly deviate from the orange/yellow parts. However, MVGen is also able to discover the more subtle view where the yellow parts are decoupled from the others. Most interesting is the third view detected by MVGen: it corresponds to only the background of the image.

Overall, MVGen successfully detects the multi-view clustering structure on a variety of data sets. Especially the explicit modeling of the views' relevant subspaces has proven to be very valuable for interpreting the final clustering results.

2.3 Subspace Correlation Clustering

While the previous methods focus on subspace clusters corresponding to dense areas in the data space, we introduced in [GFVS12] the novel paradigm of subspace correlation clustering: we analyze *subspace projections* to find *subsets of objects* showing *linear correlations* among this subset of dimensions. While existing correlation clustering methods are limited to almost disjoint clusters, our model allows each object to contribute to several correlations due to different subspace projections. For example, by considering the 2-dimensional subspace $\{d1, d2\}$ in Figure 7, two different (local) correlations can be detected: The objects indicated by a cross are positively correlated on a line, while the objects indicated by a circle are negatively correlated on a different line. Considering the subspace $\{d3, d4\}$, two different correlations supported by different sets of objects can be detected. Thus, objects may contribute to several correlations due to different subspace projections. In our paradigm, we permit multiple overlapping clusters but simultaneously avoid redundant clusters deducible from already known correlations originating from collinearity or induction. More details about this work can be found in [GFVS12].

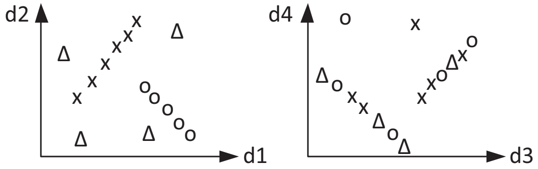


Figure 7: 4-d database with 15 objects and 4 subspace correlation clusters

3 Subspace Clustering on Imperfect Data

Most subspace clustering algorithms assume perfect data as input. Imperfect information, however, is ubiquitous where data is recorded: Missing values, for example, occur due to sensor faults in sensor networks, or uncertainty about attribute values is present due to noise or privacy issues. There is a need to handle such imperfect data for the task of subspace clustering.

Naively, traditional data cleansing techniques could be applied to preprocess the data before clustering. Data cleansing, however, has several limitations. First, data cleansing is accompanied by high cost since the methods are rarely completely automatic but the user has to be involved. Second, the storage overhead can be huge since besides the original data also the preprocessed data have to be stored. And last, preprocessing the data usually results in an information loss. On the one hand the preprocessing step is not aware of the special characteristics of the subsequent subspace clustering task as, e.g., the occurrence of objects in multiple clusters due to different subspace projections. On the other hand the mining method cannot distinguish between a precise object and an imperfect but cleaned object. Overall, valuable information is no longer available due to preprocessing.

Consequently, we propose integrated mining methods that direct handle imperfect data for the task of subspace clustering as illustrated in Figure 8. By joining the preprocessing step with the actual mining task, we are able to account for the special characteristics of subspace clustering leading to a better handling of imperfect information. Instead of mining the preprocessed data, the mining method directly analyzes the raw data and, e.g., instantiates missing values based on the currently detected subspace clusters.

Directly operating on imperfect data leads to novel requirements for subspace clustering models and definitions ranging from the accurate determination of similarity values between individual objects to the overall coherence of a cluster in an imperfect setting. The underlying challenge to be tackled by our models is their robustness against ’errors’ in the data. Even for a high-degree of imperfect information, reliably detecting high quality patterns should be possible. In our work, we describe two scenarios: our first method provides a solution for the case of data containing incomplete information and our second method covers the topic of attribute uncertainty.

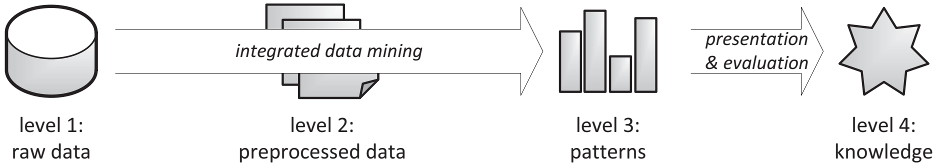


Figure 8: Enhanced KDD process by integrating the preprocessing step into the mining step for better handling of imperfect data

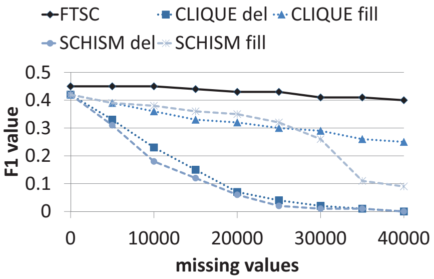


Figure 9: Clustering quality on pendigits

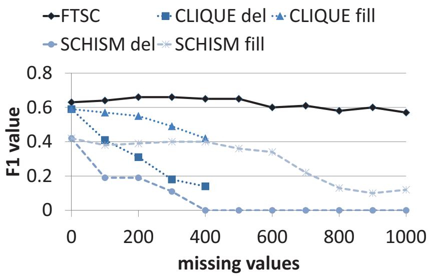


Figure 10: Clustering quality on shape

3.1 Subspace Clustering for Incomplete Data

Incompleteness describes imperfect information due to the absence of individual measurements. While the absence of specific information regarding a single object is denoted as existential incompleteness, the absence of objects as a whole is denoted as universal. In our work we tackle the challenges of existential incompleteness. Incomplete information, often referred to as data containing *missing values*, occurs for example due to faulty sensors or incomplete customer questionnaires.

In [GMRS11] we introduce a general fault tolerance definition enhancing subspace clustering models to handle missing values. Our model handles missing values based on the currently considered subspace and set of objects. Intuitively, missing values should be tolerated within a subspace cluster when the remaining objects still provide sufficient information about the relevant dimensions and the object groupings. Since a meaningful fault tolerance has to consider the varying object and attribute characteristics for each subspace cluster individually, we introduce a fault tolerance notion that adapts to the characteristics of subspace clusters. We abstract from concrete subspace clustering definitions and propose a general fault tolerance principle applicable to multiple instantiations. Thus, grid-based subspace clustering approaches as CLIQUE [AGGR98], paradigms based on the density-based clustering idea [KKK04], and several other definitions can benefit from our approach. In addition to our general model, we present a concrete instantiation – the algorithm FTSC – to the well-established grid-based subspace clustering.

As there are no competing subspace clustering approaches that handle missing values, we compare FTSC with methods working on (complete) data, cleaned by statistical preprocessing techniques. In one case we use complete case analysis and in the second case mean value imputation. To ensure a fair comparison, we apply the grid-based clustering methods CLIQUE [AGGR98] and SCHISM [SZ04] on these data since FTSC is also grid-based. In the experiments depicted in Figure 9 & 10 we analyze the methods' clustering quality on the real world datasets *pendigits* and *shape*. We increase the number of randomly distributed missing values to analyze the methods' robustness to faults in the data. For both datasets the following observations become apparent: By adding 0 missing values, the qualities of all approaches are nearly identical. The small differences can be explained by slightly different clustering definitions. Our FTSC achieves the highest clustering qualities and shows robust behavior with increasing number of missing values. Even for a huge amount of missing values the quality is high and only for some datasets a small decrease can be observed. The methods based on pre-processing show a stronger decrease of their clustering qualities. Especially, the deletion methods (CLIQUE/SCHISM del) are consistently worse than the methods based on filling up missing values by mean value imputation (CLIQUE/SCHISM fill). Summarized, our FTSC gets highest clustering qualities even if the data is polluted by a huge amount of missing values.

3.2 Subspace Clustering for Uncertain Data

In many scenarios uncertainty about the given information does exist. In the case of uncertainty, one is just provided with an estimate how likely the observed value is equal to (or may differ from) the true value. For example, the measured GPS signal of a mobile phone is highly uncertain information for determining its true position and one is only provided with an estimate about this position by, e.g., incorporating probability distributions. Similar to incomplete information, one distinguishes uncertainty about specific attributes – so called attribute uncertainty – and uncertainty about the existence of whole objects – tuple uncertainty. We consider the case of attribute uncertainty. Besides uncertainty due to the data recording step, artificial uncertainty due to privacy issues is present, i.e. before providing a data set sensitive information is obfuscated.

Data mining on uncertain databases is critical since attribute values with a large error are less reliable for data mining purposes than ones with small errors. Our novel subspace clustering method [GKS10] considers these aspects to ensure robust clustering results. Since often uncertain objects are represented by probability density functions (*pdfs*), our subspace clustering methods analyses data objects modeled as (multi-dimensional) probability density functions. Additionally, since subspace clustering tackles the challenge of clustering objects in projections of the data space, our method has to consider for each *pdf* multiple subspace projections:

Definition 3 *Projection of an uncertain object*
Given an uncertain object i represented by the pdf p_i and a subspace $S \subseteq Dim = \{1, \ldots, d\}$, the projection of p_i to S is the marginal distribution *of p_i for S. The obtained pdf is denoted as*

$$p_i^S(x) \text{ with } x \in \mathbb{R}^{|S|}$$

For example and w.l.o.g. $S = \{1, \ldots, s\}$, then

$$p_i^S(x) = p_i^S(x_1, \ldots, x_s) = \underbrace{\int \cdots \int}_{x_{s+1}, \ldots, x_d \in \mathbb{R}} p_i(x_1, \ldots, x_d)$$

i.e., we marginalize over the dimensions $\{s+1, \ldots, d\}$.

In our method we exploit the principle of grid-based subspace clustering [PJAM02]. Established for (certain) vector data, this principle groups objects into the same cluster if their distance to each other in a specific subspace is sufficiently small. Since our method has to cope with uncertain objects represented by *pdf*s, we do not calculate an actual distance value but we calculate the probability that two objects are near to each other. Formally, the probability that the distance between two independent objects i and j (represented by the *pdf*s p_i and p_j) in a subspace S is smaller than a maximal distance w is

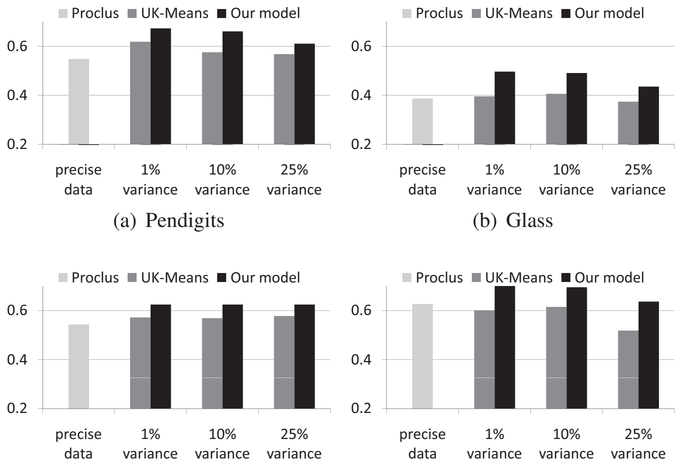
$$P_{\leq w}(p_i, p_j, S) = \int_{\substack{x,y\in\mathbb{R}^{|S|} \\ d_\infty^S(x,y)\leq w}} p_i^S(x) \cdot p_j^S(y)\, dx\, dy \tag{1}$$

We have to integrate over all possible vectors whose distance to each other is small enough and multiply the corresponding densities of the underlying *pdfs*.

Please do not confuse this probability with the values computed when considering, for example, mixture models. In mixture models, we compute for each object the likelihood of belonging to the cluster, i.e. we evaluate the density of a *single pdf* (the cluster's component) at a *given* realization (the observed data point). Here, we compute the probability that *any two* realizations of the *two given pdf*s are close to each other.

A subspace clusters in subspace S can finally be detected by randomly selecting an uncertain object m and determining all objects i whose probability for the event $P_{\leq w}(p_m, p_i, S)$ is high enough. Since in an uncertain setting each object naturally might belong to multiple clusters with different probabilities, partitioning clustering approaches are obviously out of place. Therefore, we additionally introduce a new non-partitioning clustering method by augmenting the clusters with membership degrees of their assigned objects. This improves the quality of clusterings and enables users to extract more useful information. Since our proposed model is computationally expensive, we introduce an efficient solution that uses Apriori-based pruning and heuristic sampling while still providing high quality results.

The performance of our model on real world data is analyzed in Figure 11. We present the results for the 4 datasets pendigits, glass, breast cancer, and shape. Because there exist no



(a) Pendigits

(b) Glass

(c) Breast cancer

(d) Shape

Figure 11: Clustering quality on real world data

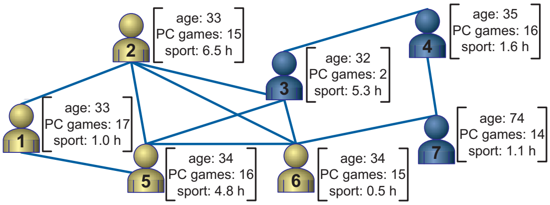


Figure 12: Exemplary social network represented by vector and graph data; highlighted in yellow: one potential twofold cluster with two relevant dimensions

direct competitors in the domain of subspace clustering for uncertain data, we compare our approach with UK-Means [CCKN06] and Proclus [AWY$^+$99]. UK-Means is chosen as a representative for fullspace clustering on *uncertain* data while Proclus identifies *subspace* clusters on certain data. Proclus is executed on the original precise data. Our model and UK-Means use the uncertain variants of the data; the variance of the underlying Gaussian distributions is set to 1%, 10%, and 25% of the data range.

The results on the pendigits dataset are presented in Figure 11(a). We can see that our model outperforms the competing algorithms. Interestingly, the results of Proclus, operated on precise data, are worse than the results of the approaches that have to cope with uncertain information. For higher variances, however, we can see a decrease in quality; the clustering structure is obfuscated by the high variance and hence a detection of clusters is difficult. On the remaining datasets similar results are obtained. Only the shape dataset (Figure 11(d)) is an exception: the quality of Proclus is slightly better than the quality of UK-Means. Nevertheless, for every dataset the effectiveness of our model is higher compared to the competing methods.

4 Subspace Clustering on Graphs with Feature Vectors

Traditional data mining algorithms process just a single type of data; e.g., objects embedded into a vector space. Today's applications, however, can acquire multiple, diverse, and heterogeneous data sources. Besides characterizing single objects by vector data, network information, for example, is a ubiquitous source to indicate the relations between different objects. Such type of heterogeneous data can be observed in various domains including social networks, where friendship relationships are available along with the users' individual interests (cf. Figure 12); systems biology, where interacting genes and their specific expression levels are recorded; and sensor networks, where connections between the sensors as well as individual measurements are given. To realize the full potential for knowledge extraction, mining techniques should consider all available information sources.

A sequential process for heterogeneous data, which first mines each type independently and then compares the detected patterns, is problematic since the results of each source might differ or even contradict. Thus, for an effective clustering, again an integrated mining promises more meaningful and accurate results. By simultaneously mining different

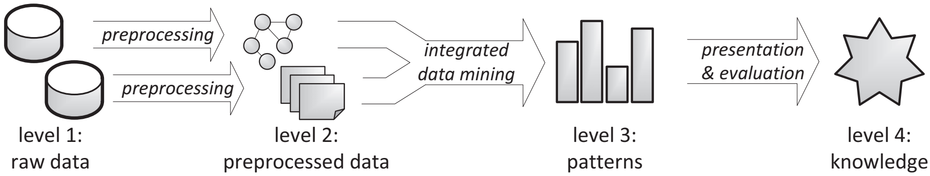


Figure 13: Enhanced KDD process by simultaneously mining multiple information types for better handling of heterogeneous data

types of information, as illustrated in the adapted KDD process of Figure 13, inaccurate information in one source can be mitigated by the other sources and an overall coherent result is possible.

In the past years, multiple integrated approaches for clustering graphs with feature vectors have been introduced. The main problem of almost all these approaches, however, is the consideration of *all* attribute dimensions for determining the similarity. As known from the previous sections, some dimensions, however, might not be relevant, which is why clusters are located in subsets of the dimensions. E.g. in social networks, it is very unlikely that people are similar within all of their characteristics. Since this aspect is not adequately considered by the existing models, we propose novel methods joining the mining task of subspace clustering and graph mining.

4.1 A Synthesis of Subspace Clustering and Dense Subgraph Mining

The GAMer approach [GFBS10] combines graph data and attribute data to identify groups according to their density of connections as well as similarity of attribute values in subsets of the dimensions. In Fig. 12 for example we are able to identify the cluster $\{1, 2, 5, 6\}$ because the objects are similar in 2 attributes and the density of the subgraph is high. A clustering procedure like this is advantageous for a variety of applications: Besides the already mentioned example of gene analysis, highly connected groups of people in social networks (graph density) can be used for target and viral marketing based on their specific preferences (attribute subset). In sensor networks, an aggregated transmission of specific sensor measurements (attribute subset) of communicating sensors (graph density) leads to improved energy efficiency and, thus, to longer lifetime of the network.

A sound combination of the paradigms *subspace clustering* and *dense subgraph mining* has to be unbiased in the sense that none of the paradigms is preferred over the other. Most integrated clustering models focus on graph properties as determining maximal sets whose density is large enough. In Fig. 12 for example the largest clique (a certain type of dense subgraphs) is $\{2, 3, 5, 6\}$; however, the vertices of this clique show similar behavior only in one of their three attributes. Even worse, preferring just high dimensional clusters leads to $\{1, 4, 6\}$; this cluster cannot be reconciled with the graph structure. Obviously the cluster properties ’density/connectedness’, ’dimensionality’, and ’size’ are usually contradictory and a clustering model has to realize a reasonable trade-off. The challenge tackled by our approach is the optimization of all three goals simultaneously to ensure their equal consideration. This enables each paradigm to be on a par with the other one in order to obtain meaningful and consistent clusters. Vertex group $\{1,2,5,6\}$ and vertex group $\{2,3,5\}$ could be possible clusters for such an optimization. In both clusters all vertices have similar values in 2 attributes, and the density of the subgraphs is negligibly smaller than in cliques.

A further important observation is that overlaps between clusters are quite reasonable. While the cluster $\{1,2,5,6\}$ might be of interest for video game producers, the cluster $\{2,3,5\}$ might be of interest for sports wear retailers. Persons thus can be assigned to more than one product target group. Also for the application of gene interaction networks and sensor networks it holds that genes can belong to more than one functional module and sensors to more than one aggregation unit. Highly overlapping clusters, however, often imply nearly the same interpretations and, thus, a strong overlap usually indicates redundancy. As shown in the previous sections of this work, considering redundancy is indispensable for subspace clustering methods. Also in the field of graph mining, avoiding redundant patterns is studied [HCS$^+$07]. The importance of a proper treatment of redundancy is hence increased for the combined consideration of subspace clustering and subgraph mining albeit rarely treated accurately in the past. Our model successfully avoids redundancy in the clustering result, while generally allowing the clusters to overlap.

Formally, the input for our model is a vertex-labeled graph $G = (V, E, l)$ with vertices V, edges $E \subseteq V \times V$ and a labeling function $l : V \to \mathbb{R}^d$ where $Dim = \{1, \ldots, d\}$ is the set of dimensions. As an abbreviation we use $l(O) = \{l(o) \mid o \in O\}$ to denote the set of vectors associated to the set of vertices $O \subseteq V$ and $x[i]$ to refer to the i-th component of a vector $x \in \mathbb{R}^d$.

The clusters detected in GAMER should represent meaningful subspace clusters and at the same time meaningful dense subgraphs. To achieve this aim, the notion of twofold clusters is introduced:

Definition 4 *Twofold cluster*
Given a graph $G = (V, E, l)$, a twofold cluster $C = (O, S)$ with respect to the thresholds $s_{min}, \gamma_{min}, n_{min}$ is a set of vertices $O \subseteq V$ and a set of dimensions $S \subseteq Dim$ with the following properties

- *$(l(O), S)$ fulfills the subspace cluster property, i.e.*

$$\forall d \in S : \ \forall x, y \in l(O) : \ |x[d] - y[d]| \leq w$$
$$\forall d \in Dim \backslash S : \ \exists x, y \in l(O) : \ |x[d] - y[d]| > w$$

- *O fulfills the quasi-clique property, i.e.*

$$\min_{v \in O}\{deg^O(v)\} \geq \lceil \gamma_{min} \cdot (|O| - 1) \rceil$$

where $deg^O(v)$ is the degree of vertex v within vertex set O

- *the induced subgraph of O is connected, $|O| \geq n_{min}$, and $|S| \geq s_{min}$*

With the beforehand introduced definition we are able to determine the set of all valid twofold clusters $Clusters$. Without any constraints this set can be large and would represent many redundant clusters. Similar to Section 2.1 we are interested in finding a non-redundant subset $Result \subseteq Clusters$ of the most interesting clusters. The interestingness of individual clusters is evaluated in GAMER via a quality function $Q(C)$. It incorporates the density, size and dimensionality of a cluster and, thus, realizes a sound and unbiased synthesis of subspace clustering and subgraph mining.

The quality function is important to identify the redundant clusters. A cluster C can only be redundant compared to a cluster C' if C's quality is lower. If the cluster C had a higher quality, then it should not be reported as redundant w.r.t. C'; the user is more interested in C. Thus, $Q(C) < Q(C')$ must hold for the redundancy of C w.r.t. C'. Furthermore, the cluster C induces redundancy w.r.t. C' if it does not describe novel structural information. In our context, this aspect means that the objects as well as the relevant dimensions of $C = (O, S)$ have already been covered to most parts by the cluster $C' = (O', S')$. If the fraction $\frac{|O \cap O'|}{|O|}$ is large, only a small percentage of C's objects are not contained in C'; we do not have a large information gain based on the object grouping of C. The same holds for the set of relevant dimensions. If all three conditions hold, the cluster C is redundant w.r.t. C'. We denote this by $C \prec_{red} C'$ and we formally define:

Definition 5 *Binary redundancy relation*
Given the redundancy parameters $r_{obj}, r_{dim} \in [0, 1]$, *the binary redundancy relation* $\prec_{red}$ *is defined by:*

$$\text{For all twofold clusters } C = (O, S), C' = (O', S'):$$
$$C \prec_{red} C' \Leftrightarrow \left[Q(C) < Q(C') \quad \wedge \quad \frac{|O \cap O'|}{|O|} \geq r_{obj} \quad \wedge \quad \frac{|S \cap S'|}{|S|} \geq r_{dim} \right]$$

Based on this relation, the optimal clustering can be defined as follows:

Definition 6 *Optimal twofold clustering*
Given the set of all twofold clusters $Clusters$, *the optimal twofold clustering* $Result \subseteq Clusters$ *fulfills*

- *redundancy-free property:* $\neg \exists C_i, C_j \in Result : C_i \prec_{red} C_j$
- *maximality property:* $\forall C_i \in Clusters \backslash Result : \exists C_j \in Result : C_i \prec_{red} C_j$

As in Section 2.1 we perform a combinatorial optimization to detect the non-redundant clustering result. Please note, though, that the above definition introduces *constraints* only and does not specify an *objective function* to be minimized/maximized. As shown in [GFBS10], the clustering fulfilling the above constraints is unique. Thus, any objective function would lead to the same result. Overall, also for the synthesis of subspace clustering with dense subgraph mining, a combinatorial optimization method can be used to find a non-redundant clustering solution.

Figure 14 shows the experimental results on a dataset comprising gene expressions and gene interactions [S+06, S+05]. The data contains 3548 vertices, 8334 edges, and 115 dimensions. For evaluating the clustering quality we use the Go-Miner tool that assigns

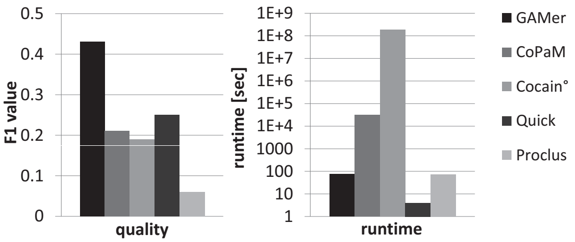


Figure 14: Clustering quality and runtime on gene data

genes to biological categories. These classes are used as hidden clusters as also done in [MCRE09]. For the experiment in Fig. 14, GAMER obtains the highest quality results. CoPaM [MCRE09] and Cocain° [ZWZK06] are not able to detect meaningful clusters. Furthermore, we calculate for this experiment the results of approaches that consider only one data source, i.e. subgraph mining (maximal quasi cliques, Quick [LW08]) or subspace clustering (Proclus [AWY$^+$99]). The quality of these two algorithms is low, indicating that a synthesis of subspace clustering and dense subgraph mining can effectively increase the clustering quality. Considering the runtime, we see that our approach is more than 100 times faster than CoPaM and even better compared to Cocain°.

Extending the GAMER method, we propose in [GBFS13] our EDCAR model. We prove the model's complexity and identify the critical parts inhibiting an efficient execution. Based on this analysis, we develop an efficient and effective algorithm that approximates the optimal clustering solution. By interweaving the process of cluster generation and cluster selection, which both make use of the GRASP (Greedy Randomized Adaptive Search Procedure) principle, we determine high quality clusters and ensure low runtimes. Figure 15 shows that EDCAR is orders of magnitudes faster than all competing approaches.

4.2 Density-Based Subspace Clustering for Graphs with Feature Vectors

The previously proposed approaches GAMER and EDCAR successfully overcome the problems of full-space clustering when analyzing graphs with feature vectors. Though,

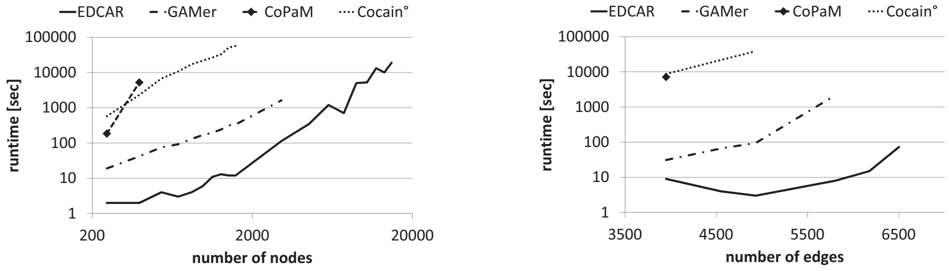


Figure 15: Scalability w.r.t. number of nodes and number of edges

the twofold cluster definition (cf. Def. 4) is restricted to clusters of certain shapes. Similar to grid-based subspace clustering [PJAM02], a cluster (w.r.t. the attributes) is defined by taking all objects located within a given grid cell, i.e. whose attribute values differ by at most a given threshold w. The methods are biased towards small clusters with little extend. This drawback is worsened by considering the used notions of dense subgraphs: e.g. by using quasi-cliques the diameter is a priori constrained to a fixed threshold [PJZ05]. For real world data, such a cluster definition might be too restrictive since clusters can exhibit more complex shapes.

In our DB-CSC model [GBS11, GBS12], we combine dense subgraph mining with subspace clustering based on a more sophisticated cluster definition; thus solving the drawbacks of previous approaches. Established for other data types, density-based clustering techniques [EKSX96, SEKX98] have shown their strength in many scenarios. They do not require the number of clusters as an input parameter and are able to find arbitrarily shaped clusters. We introduce the first approach exploiting a density-based clustering principle to join the paradigms of subspace clustering and dense subgraph mining. Our clusters correspond to dense regions in the attribute space as well as in the graph. Based on the novel notion of local densities, our DB-CSC model uses a fixed point iteration to find the desired clusters. Further pruning techniques, allow an efficient calculation of the overall clustering solution. In thorough experiments we demonstrate the strength of DB-CSC in comparison to related approaches. A more detailed discussion can be found in [GBS11, GBS12].

5 Conclusion

In this work, we proposed novel models for an effective subspace clustering of complex data. We analyzed three different types of data: vector data, imperfect data, and network data in combination with vector data. For each of these different data sources, we introduced enhanced mining models and efficient algorithms. In thorough experiments, we demonstrated the strengths of our novel clustering approaches. Overall, for the first time, meaningful subspace clustering results can be obtained for these types of complex data.

References

[AGGR98] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 94–105, 1998.

[AKMS08] I. Assent, R. Krieger, E. Müller, and T. Seidl. EDSC: Efficient Density-Based Subspace Clustering. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 1093–1102, 2008.

[AWY$^+$99] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 61–72, 1999.

[BGRS99] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When Is ”Nearest Neighbor” Meaningful? In *International Conference on Database Theory (ICDT)*, pages 217–235, 1999.

[CCKN06] M. Chau, R. Cheng, B. Kao, and J. Ng. Uncertain Data Mining: An Example in Clustering Location Data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 199–204, 2006.

[EKSX96] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.

[FA10] A. Frank and A. Asuncion. UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, 2010.

[GBFS13] S. Günnemann, B. Boden, I. Färber, and T. Seidl. Efficient Mining of Combined Subspace and Subgraph Clusters in Graphs with Feature Vectors. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2013.

[GBS11] S. Günnemann, B. Boden, and T. Seidl. DB-CSC: A density-based approach for subspace clustering in graphs with feature vectors. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 565–580, 2011.

[GBS12] S. Günnemann, B. Boden, and T. Seidl. Finding density-based subspace clusters in graphs with feature vectors. *Data Mining and Knowledge Discovery Journal (DMKD)*, 25(2):243–269, 2012.

[GFBS10] S. Günnemann, I. Färber, B. Boden, and T. Seidl. Subspace Clustering Meets Dense Subgraph Mining: A Synthesis of Two Paradigms. In *IEEE International Conference on Data Mining (ICDM)*, pages 845–850, 2010.

[GFM$^+$11] S. Günnemann, I. Färber, E. Müller, I. Assent, and T. Seidl. External Evaluation Measures for Subspace Clustering. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 1363–1372, 2011.

[GFS12] S. Günnemann, I. Färber, and T. Seidl. Multi-View Clustering Using Mixture Models in Subspace Projections. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 132–140, 2012.

[GFVS12] S. Günnemann, I. Färber, K. Virochsiri, and T. Seidl. Subspace Correlation Clustering: Finding Locally Correlated Dimensions in Subspace Projections of the Data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 352–360, 2012.

[GKS10] S. Günnemann, H. Kremer, and T. Seidl. Subspace Clustering for Uncertain Data. In *SIAM International Conference on Data Mining (SDM)*, pages 385–396, 2010.

[GMFS09] S. Günnemann, E. Müller, I. Färber, and T. Seidl. Detection of orthogonal concepts in subspaces of high dimensional data. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 1317–1326, 2009.

[GMRS11] S. Günnemann, E. Müller, S. Raubach, and T. Seidl. Flexible Fault Tolerant Subspace Clustering for Data with Missing Values. In *IEEE International Conference on Data Mining (ICDM)*, pages 231–240, 2011.

[HCS^{+}07] M. A. Hasan, V. Chaoji, S. Salem, J. Besson, and M. J. Zaki. Origami: Mining representative orthogonal graph patterns. In *IEEE International Conference on Data Mining (ICDM)*, pages 153–162, 2007.

[HK01] J. Han and M. Kamber. *Data mining: Concepts and techniques*. Morgan Kaufmann, 2001.

[Jol02] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

[KKK04] K. Kailing, H.-P. Kriegel, and P. Kröger. Density-Connected Subspace Clustering for High-Dimensional Data. In *SIAM International Conference on Data Mining (SDM)*, pages 246–257, 2004.

[KKZ09] 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.

[LW08] G. Liu and L. Wong. Effective Pruning Techniques for Mining Quasi-Cliques. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 33–49, 2008.

[MAG^{+}09] 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 *IEEE International Conference on Data Mining (ICDM)*, pages 377–386, 2009.

[MCRE09] F. Moser, R. Colak, A. Rafiey, and M. Ester. Mining Cohesive Patterns from Graphs with Feature Vectors. In *SIAM International Conference on Data Mining (SDM)*, pages 593–604, 2009.

[MGAS09] E. Müller, S. Günnemann, I. Assent, and T. Seidl. Evaluating Clustering in Subspace Projections of High Dimensional Data. *PVLDB*, 2(1):1270–1281, 2009.

[PHL04] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explorations*, 6(1):90–105, 2004.

[PJAM02] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali. A Monte Carlo algorithm for fast projective clustering. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 418–427, 2002.

[PJZ05] J. Pei, D. Jiang, and A. Zhang. On mining cross-graph quasi-cliques. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 228–238, 2005.

[S^{+}05] R. Shyamsundar et al. A DNA microarray survey of gene expression in normal human tissues. *Genome Biology*, 6, 2005.

[S^{+}06] C. Stark et al. BioGRID: a general repository for interaction datasets. *Nucleic acids research*, 34, 2006.

[SEKX98] J. Sander, M. Ester, H.-P. Kriegel, and Xiaowei Xu. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. *Data Mining and Knowledge Discovery Journal (DMKD)*, 2(2):169–194, 1998.

[SZ04] K. Sequeira and M. J. Zaki. SCHISM: A New Approach for Interesting Subspace Mining. In *IEEE International Conference on Data Mining (ICDM)*, pages 186–193, 2004.

[ZWZK06] Z. Zeng, J. Wang, L. Zhou, and G. Karypis. Coherent closed quasi-clique discovery from large dense graph databases. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 797–802, 2006.