Subspace Correlation Clustering: Finding Locally Correlated Dimensions in Subspace Projections of the Data

Stephan Günnemann, Ines Färber, Kittipat Virochsiri, and Thomas Seidl
RWTH Aachen University, Germany
{guennemann, faerber, virochsiri, seidl}@cs.rwth-aachen.de

ABSTRACT

The necessity to analyze subspace projections of complex data is a well-known fact in the clustering community. While the full space may be obfuscated by overlapping patterns and irrelevant dimensions, only certain subspaces are able to reveal the clustering structure. Subspace clustering discards irrelevant dimensions and allows objects to belong to multiple, overlapping clusters due to individual subspace projections for each set of objects. As we will demonstrate, the observations, which originate the need to consider subspace projections for traditional clustering, also apply for the task of correlation analysis.

In this work, we introduce the novel paradigm of subspace correlation clustering: we analyze *subspace projections* to find *subsets of objects* showing *linear correlations* among this subset of dimensions. In contrast to existing techniques, which determine correlations based on the full-space, our method is able to exclude locally irrelevant dimensions, enabling more precise detection of the correlated features. Since we analyze subspace projections, each object can contribute to several correlations. Our model allows multiple overlapping clusters in general but simultaneously avoids redundant clusters deducible from already known correlations. We introduce the algorithm SSCC that exploits different pruning techniques to efficiently generate a subspace correlation clustering. In thorough experiments we demonstrate the strength of our novel paradigm in comparison to existing methods.

Categories and Subject Descriptors: H.2.8 Database management: Database applications [Data mining]

Keywords: linear correlations, overlapping clusters

1. INTRODUCTION

The goal of correlation clustering[1] is to identify groups of objects that exhibit dependencies between the features of the dataset. Considering correlations, the attribute values in one dimension depend on the attribute values of other dimensions. In contrast to clustering, the objects do not have to be closely located together but should describe the same regression, e.g., they are located near the same line or plane.

[1]Not to be confused with the term as used in the machine learning community, which refers to a different task [9].

KDD'12, August 12–16, 2012, Beijing, China.

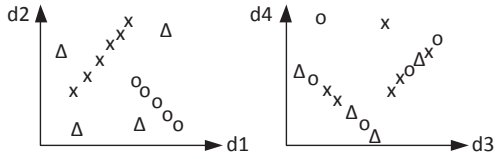

Fig. 1: 4-d database with 15 objects and 4 subspace correlation clusters

Knowing about the dependencies of attributes provides a motive to reason about causalities and is advantageous for, e.g., trend analysis, decision strategies, and prediction.

Mistakenly, in the literature correlation clustering is characterized as generalized subspace clustering [16]. Contrarily, we show in the following how traditional correlation clustering falls short of the fundamental observations and assumptions which originated the research field of subspace clustering and that the actual transfer of the correlation principles towards subspace clustering still needs to be done.

It have been two crucial observations that led the subspace clustering community to the conclusion that solely considering the full-space might deny access to a meaningful clustering structure of the data. First, for high-dimensional data, objects do not necessarily show cohesion regarding all attributes but only w.r.t. a subset of attributes. As each cluster can have its individual set of relevant attributes, global dimensionality reduction does not do the trick. Second, if objects are clustered based on differing characteristics, they can easily belong to multiple clusters when considering different views, i.e., attribute subsets of the data. Given a multitude of overlapping clusters in different subspaces, an immediate consequence is that the clustering structure might be totally obscured in the full-space [10]. This is the key challenge to be solved by subspace clustering and multi-view clustering [18, 16], where clusters highly overlap and multiple partitionings in different views can be determined. Both aspects, the obfuscation of clusters in the full-space and highly overlapping clusters, are not considered by existing correlation clustering methods. Nonetheless, these observations do also hold for correlation clustering as Fig. 1 illustrates. By considering the 2-dim. subspace $\{d1, d2\}$ in Fig. 1, two different (local) correlations can be detected: The objects denoted by a cross are positively correlated on a line, while the objects denoted by a circle are negatively correlated on a different line[2]. Considering the subspace $\{d3, d4\}$, different correlations supported by different sets of objects can be detected. Thus, observable correlations might be fulfilled only for a subset of objects and

[2]Of course our paradigm is not restricted to lines but also detects linear correlations with higher degrees of freedom.

a subset of attributes and also objects might contribute to several correlations of different attributes.

Why do we highlight this problem? One might argue that *given* an appropriate set of objects, correlation clustering can easily determine the subset of correlated dimensions and excludes the irrelevant dimensions by a post-processing. The crux, however, is to *find* the 'appropriate' set of objects in the first place. If patterns are hidden in subspace projections of the data this is an almost impossible task for approaches operating in the full-space. Since correlation clustering analyses the full-space, noisy dimensions and overlapping clusters will obfuscate the local correlations of objects leading to nearly arbitrary groupings in the full-space (similar to traditional full-space clustering). In Fig. 1 it is very unlikely to find a group of objects in the full-space that corresponds to a strong correlation in a subspace. If, however, already wrong sets of objects have been clustered, determining the true subset of correlated dimensions in a post-processing is almost impossible. Thus, correlation clustering will not just miss some clusters, but the resulting clusters are highly questionable as for the traditional full-space clustering approaches.

A novel mining paradigm. In this paper, we show that the observations made for (subspace) clustering also have to be transferred to the task of correlation analysis and we introduce the *novel paradigm of subspace correlation clustering*: First, correlations between dimensions can be restricted to subspace projections, i.e., not the whole set of dimensions is correlated. Second, correlations can be hidden in local and possibly overlapping patterns, i.e., only subsets of the objects may be correlated and objects may contribute to several correlations. With our paradigm we detect locally correlated dimensions in subspace projections. This enables to find more precise local patterns and unfolds the potential of correlation clustering also for higher dimensional datasets.

Challenges for subspace correlation clustering. By analyzing subspace projections, we inherit two crucial challenges of traditional subspace clustering. The first one arises from the exponential number of subspaces to be analyzed. Obviously, a naive exploration of all possible projections is not recommendable. Therefore, we present an efficient algorithm that uses already acquired information to apply pruning strategies. The second challenge is originated by allowing each object to contribute to several correlations. While overlapping clusters are beneficial for detecting multiple patterns in the data, analyzing each subspace projection and simply reporting any pattern, results in highly redundant information. For example a line detected in subspace S is also a line in any of its projections $S' \subseteq S$ (with $|S'| \geq 2$). Since these projections represent similar correlations, they are not beneficial for the user. Even worse, some correlations just exist due to other patterns and, thus, they represent misleading (and redundant) information. Therefore, we have to develop a redundancy model handling these scenarios to ensure interpretable results. Overall, the contributions of our work for the novel paradigm of subspace correlation clustering are:

- we analyze *subspace projections* to find *correlated dimensions* supported by a *subset of objects*
- we allow multiple overlapping clusters, i.e., each object can contribute to several correlations due to different subspace projections
- we avoid redundancy in the clustering result, i.e., we remove clusters representing similar and misleading correlation information

2. RELATED WORK

In the following, we review paradigms related to the topic of subspace correlation clustering. The general clustering objective is to group objects based on their mutual similarity. For traditional clustering, similarity of vector data is determined based on all attributes of the feature space.

Subspace Clustering [16], in contrast, determines clusters that excel by a high compactness or density in projections of the original feature space. The main challenges are to handle the exponentially many subspaces and to avoid redundancy in the result. To reduce the number of potentially relevant subspaces, several techniques exploit the apriori principle [6, 17]. Redundancy is mostly evaluated based on the clusters' similarity w.r.t. objects and attributes [6, 7, 14].

Projected Clustering is related to subspace clustering. However, projected clustering performs an (almost) partitioning of the objects, i.e., for each partition a set of relevant dimensions is detected. As consequence, it misses many of the hidden clusters or the resulting solution is of low quality since it tears multiple clusters apart to fit one partitioning.

Correlation Clustering aims at identifying object groups describing correlations between different features. Since the clusters' dimensions are not restricted to subsets of the original attributes but correspond to arbitrarily oriented subspaces, correlation clustering is often denoted as generalized *subspace* clustering [16]. This, however, is inaccurate; existing methods are rather generalized *projected* clustering methods and have the same limitations: They are not able to find multiple overlapping clusters, since they are limited to find only disjoint or, in the case of [8], nearly disjoint clusters. Even more serious is the ignorance of the obfuscation provoked by highly overlapping clusters in different subspaces, causing most approaches to fail in detecting the true correlation clusters. For completeness, we review existing correlation clustering methods and discuss further limitations. We will restrict to linear correlation clustering in the following. The basic technique utilized by most approaches is PCA. Agglomerative methods (ORCLUS [5], 4C [11], COPAC [4], ERiC [3]) assume that local neighborhoods are in line with a global trend and perform PCA on those neighborhoods, which, however, are strongly influenced by the similarity in the full-space. Therefore these techniques are not appropriate for noise or multiple views in the data as both negatively influence the choice of the neighborhood. To avoid the problem of defining a proper local neighborhood, divisive methods, e.g., [8], were proposed, which, however, usually require the number of clusters to be specified beforehand. The method of [15] is even restricted to find disjoint lines. Instead of PCA, the work of [1] utilizes the Hough transform to detect correlations within the attributes. The paradigm of pattern-based clustering [20, 16] such as co-clustering or biclustering is related; though, it is limited to positive correlations and to one-dimensional correlations.

Overall, none of the existing correlation clustering methods is able to find multiple overlapping subspace correlation clusters as it is possible with our method.

Subspace correlation analysis, e.g. CARE [21] and REDUS [22], tries to identify subsets of attributes in which the *majority* of data entries exhibit a correlation. This differs from clustering in that it does not analyze *several local subsets* of objects. Therefore these methods are restricted to finding only a single correlation per subspace and the datasets have to contain a small degree of noise concerning objects.

3. SUBSPACE CORRELATION CLUSTERS

Generally speaking, a subspace correlation cluster is a set of objects $O \subseteq DB$ that exhibits a correlation in a set of dimensions $S \subseteq Dim$. We call this set of attributes the subspace of the cluster. In Fig. 2 for example, the whole set of objects is correlated in the 2-dim. subspace $\{x, y\}$; the objects form a line. Considering the 3-dim. space, however, we do not have a valid subspace correlation cluster since the attribute z is not correlated to the other ones. Correlation is a measure of the effect of independent variables on a dependent variable [12]. As shown in Fig. 2, however, we cannot draw any conclusion about the attribute values in dimension z given the attribute values in the remaining dimensions. Thus, the plane does not represent a valid subspace correlation cluster. Unlike existing correlation clustering approaches we will not report such invalid correlation clusters as they encourage highly misleading interpretations. Instead we will use this principle of induced clusters for redundancy modeling and efficiency improvements in Section 4.

In Fig. 3, the illustrated plane consisting of all objects in the 3-dim. space, however, is a valid correlation. Given the attribute values of an object in dimension x and y, the attribute value in dimension z is dependent. An object in this correlation is described by *two independent variables*. If we just consider a subset of the objects, we are even able to find a line in the 3-dim. subspace, i.e., given the attribute value in dimension x for example, the attribute values of dimension y and z are completely dependent. For a line we have *one independent variable*, i.e., one degree of freedom.

As shown in the examples, for subspace correlation clusters we have two different types of dimensionalities: First, the *subspace dimensionality*: e.g., the 3-dim. subspace of $\{x, y, z\}$. Second, the *cluster dimensionality*: Within each subspace each cluster/correlation has its own intrinsic dimensionality. E.g., a plane corresponds to a 2-dim. cluster (two degrees of freedom) while a line is a 1-dim. cluster. Thus, in general a subspace correlation cluster is defined by a three-tuple (O, S, λ) with a set of objects $O \subseteq DB$, a set of correlated dimensions $S \subseteq Dim$, and the clusters dimensionality λ, representing the degrees of freedom. We now formalize the definition of such subspace correlation clusters.

3.1 Cluster definition

The basic idea to describe correlations in the data is by considering the data's principal components. We adapt the notions of [2] to define our clusters. Since we design our model to detect linear correlations we will use the term 'correlation' in place of 'linear correlation'.

DEFINITION 1. *Basic notions*
We assume a database $DB \subseteq \mathbb{R}^d$ of d-dimensional objects is given. $Dim = \{1, \ldots, d\}$ is the set of dimensions. With $o_{|S}$ we denote the projection of an object $o \in DB$ to the subspace $S \subseteq Dim$. Accordingly, $O_{|S}$ refers to the projection of a set of objects $O \subseteq DB$. The (sample) covariance matrix for $O_{|S}$, i.e., for a set of objects $O \subseteq DB$ in a subset of dimensions $S \subseteq Dim$, is denoted by $\Sigma_{O,S} \in \mathbb{R}^{|S|} \times \mathbb{R}^{|S|}$. The eigendecomposition of $\Sigma_{O,S}$ is $\Sigma_{O,S} = V_{O,S} \cdot E_{O,S} \cdot V_{O,S}^T$. The eigenvalue matrix $E_{O,S}$ is a diagonal matrix storing the $|S|$ eigenvalues in decreasing order, i.e., $E_{O,S} = diag(e_1, \ldots, e_{|S|})$ with $e_1 \geq e_2 \geq \ldots \geq e_{|S|}$. The eigenvector matrix $V_{O,S}$ is an orthonormal matrix storing the unit eigenvectors v_i corresponding to e_i. Therefore, the i-th principal component is given by $e_i \cdot v_i$.

Given a set of objects O projected to the subspace S, i.e., $O_{|S}$, the minimum number of principal components needed to retain a significant level α of the data's variance corresponds to the cluster's intrinsic dimensionality. Considering the line (defined by a subset of the objects) in the 3-dim. subspace in Fig. 3, we just need one principal component to describe most of the variance. This is indicated by one large eigenvalue and two small ones. In contrast, for the plane we need two principal components. Thus, by using the eigenvalues of the covariance matrix, we are able to determine the cluster dimensionality.

DEFINITION 2. *Cluster dimensionality*
The cluster dimensionality of the (projected) set of objects $O_{|S}$ w.r.t. a significance level α is

$$\lambda(O, S) = \min\{k \in \mathbb{N}^+ \mid \frac{\sum_{i=1}^{k} e_i}{\sum_{i=1}^{|S|} e_i} \geq \alpha\}$$

using the entries e_i of the eigenvalue matrix $E_{O,S}$.

In real world scenarios we cannot expect to observe a set of objects that perfectly fits a line or another linear regression model. Due to errors and noise in the data, the objects slightly deviate from the perfect model. We can measure the strength of a correlation by the degree of variation *not* explained by the regression model. Since the regression model corresponds to the hyperplane spanned by the first $\lambda(O, S)$ principal components with the largest eigenvalues (strong principal components), the distance along the last $|S| - \lambda(O, S)$ principal components with the smallest eigenvalues (weak principal components) determines the strength of a correlation. Intuitively, this corresponds to the distance between an object and, e.g., the perfect line. The smaller these distances, the stronger the correlation. Formally, the correlation distance can be computed by determining the Euclidean distance after projecting the objects and the cluster mean onto the weak principal components (please note: the objects are first projected to the subspace S, i.e., the weak components are linear combinations of the dimensions in S):

DEFINITION 3. *Subspace correlation distance*
The subspace correlation distance of an object $p \in DB$ to a correlation defined by a set of objects O in subspace S is:

$$scdist(p, O, S) = \sqrt{(p_{|S} - \mu_{|S})^T \cdot V_{O,S} \cdot \hat{E} \cdot V_{O,S}^T \cdot (p_{|S} - \mu_{|S})}$$

with mean vector μ of the objects in O, $V_{O,S}$ is the eigenvector matrix, and $\hat{E}$ is a diagonal matrix where the first $\lambda(O, S)$ entries are 0 and the remaining $|S| - \lambda(O, S)$ entries are 1.

Thus, we obtain a strong correlation if each object of the cluster has a small subspace correlation distance. Using these basic ideas, we first introduce our novel cluster definition and afterwards we highlight the important characteristics.

DEFINITION 4. *Subspace correlation cluster*
A subspace correlation cluster $C = (O, S, \lambda)$ is a set of objects $O \subseteq DB$, a set of dimensions $S \subseteq Dim$, and its cluster dimensionality $\lambda = \lambda(O, S)$, such that

1. *the cluster is sufficiently large, i.e., $|O| \geq minSize$*
2. *the subspace correlation distance is small for each object in the cluster, i.e., $\forall o \in O : scdist(o, O, S) \leq \epsilon$*
3. *the set of objects is maximal: any object not in the cluster has a larger subspace correlation distance, i.e., $\forall p \in DB \backslash O : scdist(p, O, S) > \epsilon$*
4. *the cluster dimensionality is smaller than the subspace dimensionality, i.e., $\lambda < |S|$*
5. *uncorrelated dimensions are not included, i.e., $\forall d \in S : \lambda(O, S \backslash \{d\}) = \lambda(O, S)$.*

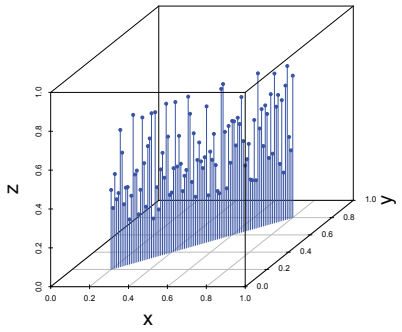

Fig. 2: Valid correlation in subspace $\{x, y\}$, invalid in $\{x, y, z\}$

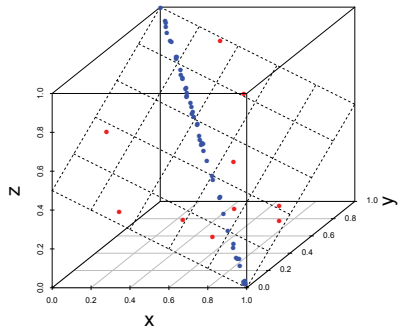

Fig. 3: Redundant plane (induced by the non-redundant line)

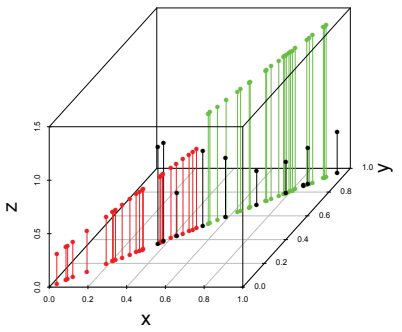

Fig. 4: Redundancy (line in subspace $\{x, y\}$) due to collinearity

The first property ensures that each cluster has sufficiently large support since, e.g., a line composed by only two points is not interesting. By the second property, we ensure that each cluster exhibits a strong correlation; we do not include objects with large subspace correlation distances. Accordingly, we include any object with a small distance, i.e., the cluster should be as large as possible (third property).

The last two properties are the most important ones: With Prop. 4 the case $\lambda = |S|$ is prohibited ($\lambda > |S|$ is not possible by definition). For $\lambda = |S|$, the cluster dimensionality and the subspace dimensionality would be identical. This is not meaningful since each dimension in the current subspace would be an independent variable. Thus, none of the attribute values depends on the others, which does not represent a correlation.

Property 5 is even more important to avoid "meaningless" correlations (and is one large advantage in contrast to traditional correlation clustering). As shown in Fig. 2, the plane is not a valid correlation; dimension z is not correlated to the other ones. This, however, cannot be detected by considering just the eigenvalues. The cluster dimensionality is 2 as for any other plane. Thus, the first 4 properties of the previous definition hold. However, we can detect this invalid correlation by the following principle: Lets assume we have detected a set of objects with dimensionality λ' in subspace $S\backslash\{d\}$. By adding a further dimension, i.e., to get the subspace S, the novel cluster dimensionality obviously has to be either $\lambda' + 1$ or λ'. The first case occurs if we add an uncorrelated/noisy dimension to $S\backslash\{d\}$, e.g., a line in a 2-dim. subspace becomes a plane in a 3-dim. subspace (cf. Fig. 2). The latter case occurs if we add a correlated dimension, e.g., a line still is a line (Fig. 3). Thus, Property 5 checks for each lower-dimensional subspace $S\backslash\{d\}$ whether the cluster dimensionality remains stable. Otherwise ($\lambda(O, S\backslash\{d\}) < \lambda(O, S)$), at least one dimension d would be uncorrelated and the cluster is not valid.

Overall, our cluster definition enables us to detect groups of objects O that are correlated in the set of dimension S with a specific degree of freedom λ by simultaneously excluding uncorrelated dimensions. Note that the *smaller* λ the more interesting is a cluster; a line is better than a plane. Contrarily, the *larger* S the more interesting is a cluster; a 3-dim. subspace is better than a 2-dim. subspace.

3.2 Clustering definition

Based on Def. 4 we are able to determine all valid clusters, which are allowed to group diverse object sets in diverse subspaces. Thus, we allow overlapping clusters in general and we are able to detect multiple different correlations per object, e.g., due to different attribute subsets. We are not limited to disjoint clusters as previous methods. However, simply using the set *All* of all valid subspace correlation clusters according to Def. 4 would lead to an overwhelming result size, containing highly redundant information. This redundancy occurs due to two reasons:

Collinearity: When one dimension $d \in S$ is highly correlated with some other dimension x, the remaining dimensions $S\backslash\{d\}$ will be correlated to x too. This phenomenon is called (multi)collinearity [13]. When this happens, we can derive many clusters describing the same information, e.g., if $\{d1, d2\}$, $\{d2, d3\}$ and $\{d3, d4\}$ are correlated, we directly can infer that $\{d1, d3\}$, $\{d1, d4\}$, ... are correlated as well. Instead of reporting several collinear correlations, we represent them by just one cluster (in a subspace of higher cardinality) and thus avoid redundancy. To handle collinearity, we have to ensure that a correlation cluster in subspace S is not simply a projection of another cluster in a supersubspace $S' \supset S$. As illustrated in Fig. 4, the correlation cluster in subspace $\{x, y\}$ is a projection of two clusters in subspace $\{x, y, z\}$ (the green and red clusters) and a few noise objects (black dots). Such a cluster does not provide much additional insight and therefore should be excluded from the final result.

Induced clusters: With a few extra points, a correlation cluster can induce a cluster of higher dimensionality (e.g., a line induces a plane). In Fig. 3, the 1-dim. correlation cluster (blue line) induces the 2-dim. correlation cluster (plane) which has only a few extra objects. Induced clusters can be misleading as they mainly take credit for another cluster's objects. In this example, the plane is mainly supported by the objects of the line. If we took the line's objects out of consideration, the plane would have too little support to be a valid correlation cluster. The plane is just valid because of the line but not by itself. Thus, we should exclude induced clusters from the final result.

The 'potentially redundant' rule. The main question is, which clusters could lead to a redundancy of other clusters either due to collinearity or due to induction. The following definition clearly states this:

DEFINITION 5. *'Potentially redundant' rule*
A subspace correlation cluster $C = (O, S, \lambda)$ is potentially redundant to $C' = (O', S', \lambda')$, for short $C \prec_{red}^{pot} C'$, iff

$$\lambda - \lambda' \geq |S\backslash S'| \vee \exists S^* \subset S : |S^*| = \lambda + 1 \wedge \lambda - \lambda' > |S^*\backslash S'|$$

We continue by discussing why the above rule covers all clusters C' to which C is potentially redundant to. We distinguish three cases: *(1)* $\lambda < \lambda'$ (the cluster dimensionality of C is lower (better) than the one of C'): In this case, the

rule is always evaluated to $false$ as the left sides of both inequalities are negative while their right sides are at least 0. This means a lower dimensional cluster (e.g., a line) is never redundant w.r.t. a higher dimensional cluster (e.g., a plane). This is desirable because high-dim. clusters cannot induce low-dim. ones.
(2) $\lambda = \lambda'$ (both clusters have the same dimensionality): In this case, the rule will be true if and only if $S' \supseteq S$. Otherwise the right side of the first inequality will be greater than 0, and the second part of the rule is violated in any case since a strict $>$ is required. This means a cluster in a higher-dimensional subspace is more important than its projections. As we can derive clusters in lower-dimensional subspaces by simply projecting the cluster of a higher-dimensional subspace, clusters in lower-dimensional subspaces need not to be presented explicitly (cf. collinearity).
(3) $\lambda > \lambda'$ (the cluster dimensionality of C is higher (worse) than the ones of C'): This case is the most complex one and corresponds to induced correlation clusters, e.g., C is a plane while C' is a line. As before, the rule is true for the case $S' \supseteq S$. Though, it can even hold for $S' \not\supseteq S$.

Let us consider an example: C is a 4-dim. cluster in subspace $S = \{1, 2, 3, 4, 5\}$ and C' is a 1d-cluster (line) in subspace S'=$\{1, 2, 3, 8, 9\}$. How does C' look like in S? If we project C' to $\{1, 2, 3\}$=$S \cap S'$, it has to be still a line. By now adding two further dimensions, e.g., $\{4, 5\}$ to reach S, the dimensionality of the 1d-cluster increases at most by two (if both dimensions are uncorrelated). Thus, in the worst case the dimensionality of C' in S is 1+2=3, which is better than the 4d-cluster C. Hence C' could potentially induce C. In general, this holds if $\lambda' + |S| - |S' \cap S| < \lambda \Leftrightarrow |S \setminus S'| < \lambda - \lambda'$.

But there is more: Assume C to be a 4-dim. cluster in subspace $S = \{1, 2, \ldots, 7\}$. Obviously, the line C' does look like a 5-dim. cluster in S. It cannot induce the cluster C. However, in our model the cluster C represents all of its collinear projections, e.g., also the 4-dim. cluster in subspace $\{1, \ldots, 5\}$=$S^* \subset S$. As discussed, the cluster in this subspace may be induced by C' and hence could be redundant. Consequently, if the projection S^* is redundant then this information should not be contained in the 'parent' cluster. We also have to denote C as potentially redundant to C'. In general, the equation $\lambda - \lambda' > |S^* \setminus S'|$ has to be checked for any subset $S^* \subset S$ with $|S^*| = \lambda + 1$ since these are the lowest dimensional subspaces represented by collinear information of C.

Overall model. Based on the 'potentially redundant' rule, we define the overall redundancy of a cluster. As discussed and illustrated in Fig. 3, the plane should be discarded as redundant because its support is too small after removing the line. Similarly, in Fig. 4, the line in subspace $\{x, y\}$ should be discarded, since the support is mainly due to the two lines in subspace $\{x, y, z\}$. Thus, we define a cluster C as redundant w.r.t. a set of other clusters *Result* if C's support is too small after removing all objects that are already grouped in clusters $C' \in Result$ to which C is potentially redundant.

Definition 6. *Redundancy of clusters*
A subspace correlation cluster $C = (O, S, \lambda)$ is redundant to a set of clusters Result, for short $C \prec_{red} Result$, iff

$$|O \setminus \bigcup_{C'=(O',S',\lambda') \in Red} O'| < minSize$$

with $Red = \{C' \in Result \setminus \{C\} \mid C \prec_{red}^{pot} C'\}$

Finally, we define the overall subspace correlation clustering. The final clustering should be redundancy free, i.e., it should not contain induced clusters or clusters present due to collinearity. Moreover, it should be maximal, i.e., should contain as many clusters as possible without introducing redundancy. Based on these two principles, we define the overall clustering model as follows:

Definition 7. *Subspace correlation clustering*
Given the set All of all valid subspace correlation clusters, a subspace correlation clustering Result $\subseteq$ All fulfills the following conditions:

- *redundancy-free:* $\forall C \in Result : C \not\prec_{red} Result$
- *maximality:* $\forall D \in All \setminus Result : Result \cup \{D\}$ *is not redundancy-free*

Our novel clustering model enables the detection of correlations in subspace projections of the data, it allows objects to contribute to multiple, overlapping correlations, and simultaneously prevents redundant information in the result.

4. THE SSCC ALGORITHM

In the following section we briefly introduce our algorithm SSCC, which determines a clustering result according to Definition 7. We refer to an efficient approximation since, as known from traditional subspace clustering, the number of cluster candidates is exponential in the number of objects and the number of dimensions. Furthermore, generating all cluster candidates in a first step and selecting the final clustering afterwards is highly inefficient since most of the clusters will be rejected as redundant anyway. Thus, our SSCC avoids the bottleneck of generating all possible clusters by trying to directly generate only non-redundant clusters.

The general processing of SSCC is shown in Algo. 1. We first describe the general idea of our approach. Three major principles are used to avoid generating redundant clusters: *(1)* Based on the redundancy definition, a λ-dim. cluster can only be redundant to clusters with dimensionality $\lambda' \leq \lambda$. Thus, we can first mine all (non-redundant) low-dimensional clusters (e.g., lines) before mining higher-dimensional ones (e.g., planes). This is shown in line 2 of the algorithm. *(2)* The support of a non-redundant λ-dim. cluster in subspace S has to be high enough *after* removing the objects contained in clusters that make him potentially redundant (cf. Def. 6). We use this idea for pruning objects: Since we already know the non-redundant λ'-dim. clusters with $\lambda' \leq \lambda$ in similar subspaces, we remove all objects of the database (in this subspace) that are already clustered (line 5). Thus, the set of objects to be considered is dramatically reduced (or even too small; line 6) and finding clusters is more efficient. *(3)* We exploit the fact of collinearity to avoid analyzing the exponential number of possible subspaces. Each λ-dim. cluster in subspace S is also a λ-dim. cluster in any subspace $S' \subseteq S$ with $|S'| = \lambda + 1$. Thus, we only mine λ-dim. clusters in subspaces with cardinality $\lambda + 1$ (cf. lines 4, 7) and we *merge* these clusters to obtain clusters with higher subspace cardinality (line 8). For example, if we have already found similar clusters in subspace $\{d1, d2\}$ and $\{d1, d3\}$, we merge them to $\{d1, d2, d3\}$. If the cluster is still valid here, we do not have to analyze $\{d2, d3\}$ since in our model the collinearly correlated dimensions are represented by a single cluster.

Based on these strategies, the lines 3-10 of Algo. 1 efficiently generate a set of λ-dim. clusters (located in subspaces of arbitrarily high cardinality) that is non-redundant with high probability. To finally guarantee a non-redundant result, this set is refined (lines 11-14): To remove redundant

```
1  Result = ∅ // current result containing non-red. clusters
2  for λ from 1 to d − 1 do // low-dim. clusters first
3      λClusters = ∅
4      for S ⊆ Dim with |S| = λ + 1 do
5          NonPruned_S=PruneObjects(DB,S,Result,λClusters)
6          if |NonPruned_S| < minSize then continue;
7          Tmp=FindλClusters(NonPruned_S, S, λ)
8          λClusters=MergeClusters(Tmp, λClusters) // Alg. 2
9          // λClusters may now contain clusters with
10         // subspace cardinality |S_C| > λ+1
11     for x from d to λ + 1 do // high-d subspaces first
12         for C ∈ λClusters with |S_C| = x do
13             if C is non-redundant to Result then // Def. 6
14                 Result = Result ∪ {C}
```

Algorithm 1: The SSCC algorithm

```
1  input: new clusters Tmp, already known λClusters
2  while Tmp ≠ ∅ do
3      select C = (O, S, λ) ∈ Tmp, CIsRedundant = false
4      PotColl = {C_i ∈ λClusters | |S ∩ S_i| ≥ λ ∧ S_i ⊈ S}
5      for C_i ∈ PotColl do
6          generate C_merge = (O ∩ O_i, S ∪ S_i, λ)
7          if C_merge is valid cluster then
8              Tmp = Tmp ∪ {C_merge}
9              if C_i is redundant to C_merge then
10                 λClusters = λClusters\{C_i}
11             if C is redundant to C_merge then
12                 CIsRedundant = true
13     Tmp = Tmp\{C}
14     if CIsRedundant == false then λClusters.add(C)
```

Algorithm 2: Generate high-dim. subspaces by merging

clusters, we can use an efficient incremental approach since based on Def. 5 we can process the clusters of highest *subspace* cardinality first (line 11,12) and we just have to compare these against the current result set (line 13).

Finding λ-dim. clusters in a (λ + 1)-dim. subspace. As described above, we just have to mine the λ-dim. clusters in (λ + 1)-dim. subspaces. To achieve this, we apply the method of COPAC [4] on the current subspace projection; however, with two important differences: First, COPAC partitions the objects according to their local correlation dimensionality and finds clusters in any of these partitions. Since we are just interested in λ-dim. clusters, we just have to analyze a single partition; the one corresponding to λ. This is far more efficient. Note that we still find clusters of higher dimensionality due to our overall processing. Second, our approach avoids uncorrelated dimensions, i.e., clusters containing principal components that are nearly parallel to any axis of the current subspace are rejected. Overall, we efficiently generate the desired set of λ-dim. clusters.

Merging clusters to higher-dimensional subspaces. By our merging principle we avoid to analyze any possible subspace projection. Given the set Tmp of newly detected λ-dim. clusters (cf. line 7), we try to merge these with the already known $\lambda Clusters$ to reach subspaces of higher cardinality. A pseudo-code for this subroutine is given in Algo. 2. For each cluster $C = (O, S, \lambda) \in Tmp$ we first determine those $C_i = (O_i, S_i, \lambda) \in \lambda Clusters$ that fulfill $|S \cap S_i| \geq \lambda$ and $S_i \not\subseteq S$. Only these clusters are potentially collinear to C. If no such C_i exists, we can simply add C to the current set of $\lambda Clusters$. Otherwise, for each potentially collinear cluster C_i we do the following steps:

We generate the candidate $C_{merge} = (O \cap O_i, S \cup S_i, \lambda)$ and check if it is a valid cluster. If so, we first add C_{merge} to Tmp since it potentially can be merged with further clusters later on (Note: Tmp acts as a queue, where we successively remove and add elements; the method stops if Tmp is empty). Second, we check whether C_i is redundant to C_{merge}. Since the 'potentially redundant' rule is automatically fulfilled, we simply have to test $|O_i| - |O \cap O_i| < minSize$. If C_i is redundant, we remove it from $\lambda Clusters$, ensuring a manageable number of clusters at any time. Similarly, we check the redundancy of C w.r.t. C_{merge}. If it is redundant, we mark C.

If each C_i is processed, i.e., its merging with C has been analyzed, we remove C from Tmp. If C is *not* marked as redundant, we finally add C to the set $\lambda Clusters$.

Note that the termination of the merging principle is guaranteed due to the condition $S_i \not\subseteq S$. Since the subspace cardinality of the merged clusters increases, at some point in time no further merging partners can be found and the set Tmp will become empty. Also note that the merging is invoked *within* the for-loop (Alg. 1, line 4). Thus, the set Tmp is usually small and the clusters in $\lambda Clusters$ may already be of much higher subspace cardinality than $\lambda + 1$. Overall, our merging principle efficiently generates λ-dimensional clusters of subspace cardinality larger than $\lambda + 1$.

Object pruning exploiting the redundancy model. Our pruning lowers the number of objects of the current subspace S that have to be analyzed for clustering structure. According to Def. 6, a non-redundant cluster C_S of dim. λ must have sufficiently high support even if the object sets of some other clusters (based on Def. 5) are removed. Since we already know the non-redundant clusters with dimensionality $\lambda' < \lambda$, we can select those clusters $C' \in Result$ that fulfill the 'potentially redundant' rule w.r.t. the current subspace S, i.e., those clusters C' for which $C_S \prec_{red}^{pot} C'$ holds[3]. Clusters fulfilling the rule might be the reason for *induced* correlation clusters in S and hence their object sets can safely be removed. If C_S is a valid non-redundant cluster in S, it will still be so even after removing the objects determined above.

But we can remove even more objects: The merging step generates clusters with subspaces of cardinality larger than λ+1. For example, based on clusters in $\{d1, d2\}$ and $\{d1, d3\}$ we may get a cluster C_m in $\{d1, d2, d3\}$; it has the same cluster dimensionality λ but higher subspace cardinality. Thus, before analyzing the subspace $\{d2, d3\}$ we can also remove the objects of C_m since the 'potentially redundant' rule holds for this cluster. In general, we can prune objects contained in clusters $C' \in \lambda Clusters$ with $C_S \prec_{red}^{pot} C'$. This prevents the detection of redundant *collinear* clusters with lower subspace cardinality. Hence, our merging principle introduced above also leads to an efficiency gain within this subroutine. Formally, the set of non-pruned objects is given as

$$NonPruned_S = \{o \in DB \mid \neg\exists C' \in Result \cup \lambda Clusters : C' = (O', ., .) \wedge o \in O' \wedge (DB, S, |S| - 1) \prec_{red}^{pot} C'\}$$

Please note that dependent on the current subspace S different sets of objects are pruned since the 'potentially redundant' rule may be evaluated differently. Thus, objects can still contribute to several correlations due to different subspace projections as desired by our model.

Overall, for each subspace S a large amount of objects might be already removed based on the clusters in $Result$ (to

[3]At this point we do not know which objects O are clustered in C_S. However, since Def. 5 just uses λ and S, and we have $\lambda = |S| - 1$, all necessary information is given. We can, e.g., simply set $C_S = (DB, S, |S| - 1)$.

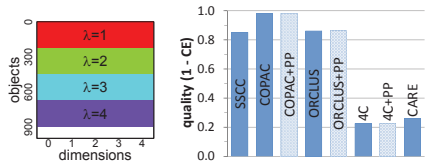

Fig. 5: Scenario A

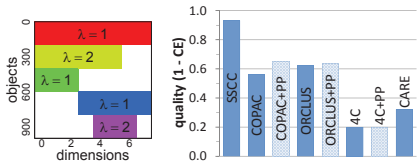

Fig. 6: Scenario B

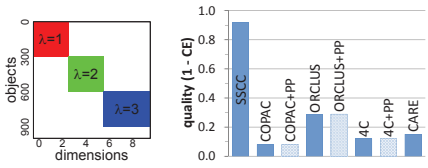

Fig. 7: Scenario C

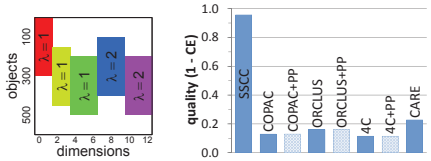

Fig. 8: Scenario D

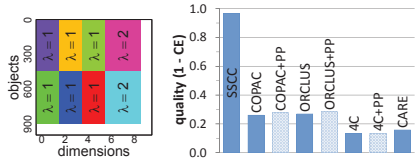

Fig. 9: Scenario E

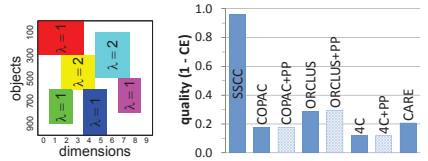

Fig. 10: Scenario F

prevent induced clusters) and $\lambda Clusters$ (to prevent collinear clusters). Only in the remaining set of objects novel clusters have to be detected.

Summary. SSCC generates clusters bottom-up w.r.t. their dimensionality and simultaneously removes redundant clusters top-down w.r.t. their subspace cardinality. Due to the redundancy model and based on the increasing set of known clusters, we are able to prune a large amount of objects that needs not to be considered at all. By utilizing the collinearity phenomenon and merging several clusters, we avoid analyzing many subspace projections. Overall, SSCC efficiently determines a non-redundant subspace correlation clustering.

5. EXPERIMENTAL ANALYSIS

Setup. We compare SSCC against all (non-hierarchical) algorithms implemented in the framework ELKI[4], namely ORCLUS [5], COPAC [4], and 4C [11]. For subspace correlation analysis we choose CARE [21] since, unlike REDUS [22], it is exclusively designed for linear correlations. To revisit our statement of the introduction that simple post-processing is not sufficient in most subspace scenarios to reveal the clustering structure, we implement a post-processing method that discards all dimensions being approximately parallel to the principal components of a cluster and recalculates the cluster's dimensionality. We name this step PP and apply it to all results of full-space correlation clustering methods. We measure clustering quality by the CE measure (clustering error) [19], which also considers the subspaces in its evaluation. For easier interpretation we depict the results of 1-CE, where 1 indicates perfect quality and 0 lowest possible quality. For each algorithm we determine optimal parameter settings w.r.t. the CE value. Efficiency is measured by the approaches' runtime. For comparability all experiments were conducted on Opteron 2.3GHz CPUs using Java6 64bit.

We start by evaluating the approaches based on synthetic data to analyze their performance for the different subspace scenarios. We continue with different scalability experiments and will confirm important observations for the real world datasets 'Wages'[5] and 'Image Segmentation'[6].

Test scenarios for subspace clustering. For Figures 5-10 we examine the algorithms' results for different correlation scenarios, especially for subspace settings. Due to space limitations we provide only visual descriptions of the used datasets attached to the left of the evaluation results of each particular test scenario. Each colored region corresponds to one cluster covering a specific set of objects in a specific subset of dimensions. λ indicates the cluster's dimensionality. Attribute values in white regions are noise.

For the first simple test *scenario A* (Fig. 5), with only full-space clusters, the full-space approaches COPAC and ORCLUS perform better than SSCC. Since clusters are well separated in the full-space but are likely to be merged in subspace projections, the merging strategy of SSCC tends to assign few objects to wrong clusters. Surprisingly, 4C does not yield good results, which probably is originated by its sensitivity to the setting of the neighborhood range, which however might be different for clusters of different dimensionality λ, and its requirement of spatially connectedness of the clusters. Since the CARE approach requires a cluster to comprise the majority of a dataset's objects, it does not perform well for datasets with more than one cluster per subspace, which will be confirmed by the other test cases.

In this scenario, we also examine the importance of *redundancy handling* when considering subspace projections: to this aim, we applied COPAC on each subspace projection. It generated 52 clusters, instead of the four hidden clusters. This matches the observation that each cluster $C = (O, S, \lambda)$ appears as projection in each subspace $S' \subseteq S$ with $|S'| \geq max\{2, \lambda\}$, yielding for this data to the theoretical result of 49 redundant clusters. Clearly, results obfuscated with redundancy would be unmanageable for higher dimensional data; naively using existing correlation clustering in subspace projections is not a choice to detect subspace correlation clusters. In contrast, SSCC detects only the four non-redundant clusters.

Test case B (Fig. 6) shows a simple subspace scenario, where clusters are disjoint and only few dimensions per cluster are noisy. SSCC manages to achieve even better clustering results for this setting. The results of COPAC and ORCLUS are significantly lower compared to scenario A. Noise dimensions obfuscate the clustering structure in the full-space and even *post-processing* only marginally improves the quality. In contrast, CARE is able to achieve better results than for test case A, as clusters overlap less per dimension.

Although *scenario C* (Fig. 7) seems to be the easiest subspace setting, as neither dimensions nor objects of clusters do overlap, it poses severe challenges for full-space algorithms. Since the degree of noise dimensions exceeds the one of relevant dimensions for all clusters, COPAC, ORCLUS, and 4C do not manage to reveal the true clustering structure. Even

[4]Achtert, Kriegel, Zimek. ELKI: A Software System for Evaluation of Subspace Clustering Algorithms, SSDBM, 2008

[5]http://lib.stat.cmu.edu/datasets/CPS_85_Wages

[6]UCI Machine Learning Repos., http://archive.ics.uci.edu/ml

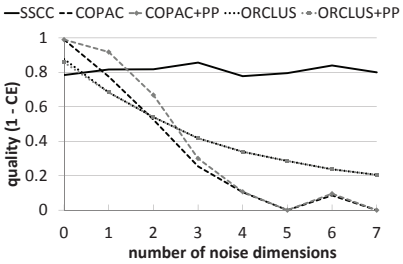

Fig. 11: Effect of noise dimensions

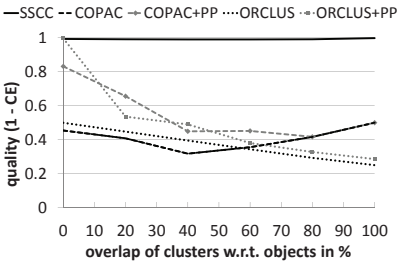

Fig. 12: Effect of cluster overlaps

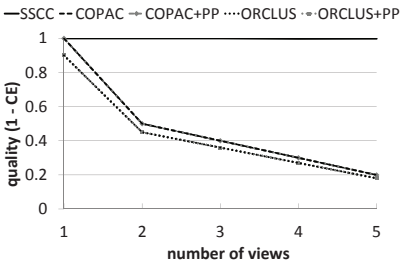

Fig. 13: Effect of multiple views

post-processing achieves no further improvement. SSCC is the only approach being able to uncover the correlations. For the clusters of *scenario D* (Fig. 8) not objects but subspaces are disjoint. As all approaches besides SSCC partition the data, they will not recover the clustering structure by design. However, since the clusters' overlap is arbitrary, the correlations interfere with each other and no clear statement about correlations is possible in the full space. Still, SSCC nearly perfectly recovers the hidden correlations. Although, the degree of overlap of clusters w.r.t objects is increased in *scenario E* (Fig. 9) compared to scenario D, clustering results of COPAC and ORCLUS are better. Clusters either agree perfectly or not at all regarding their objects or dimensions. Therefore the projection into full-space still enables the detection of a certain clustering structure. Thus not only the partitioning of the data hinders the cluster detection for full-space algorithms but also the arbitrary interference shown in scenario D. The last *test case F* (Fig. 10) shows a scenario typically described for subspace clustering. For these combined characteristics of the previous cases, we once more observe that none of the examined approaches but SSCC is able to reveal the underlying correlation clustering structure.

For further experiments we will restrict our comparison of SSCC to COPAC and ORCLUS, as the remaining methods have shown to be not effective to typical subspace scenarios.

Number of noise dimensions. To confirm our observations of scenario B, in Fig. 11 we gradually add noise dimensions to the dataset of Fig. 5. While SSCC is mostly unaffected by noise dimensions, COPAC and ORCLUS increasingly fail to discover the hidden correlations. For few noise dimensions, post-processing slightly increases the quality of COPAC. In general, however, post-processing cannot eliminate the problems of full-space clustering.

Overlap of clusters. In Fig. 12 we increase the percentage of overlapping objects between clusters. For a 4-dim. dataset with 1000 objects, 2 clusters each with 500 objects in disjoint 2-dim. subspaces, we increase the clusters' overlap without varying the clusters' sizes. SSCC shows perfect result in any case. ORCLUS and COPAC with post-proc. are able to reveal the true correlations for small overlap. However, with increasing overlap, both algorithms are destructed by the interference of the two clusters in the full-space, shown by worse post-proc. quality. Even worse, COPAC groups the overlapping objects into a separate cluster, which is more beneficial according to CE than to regard them as noise.

Number of views. In Fig. 13 we increase the number of views for a dataset of 1000 objects and 10 dimensions. Comparably to scenario E (Fig. 9), all views have disjoint, nearly equally sized subspaces and contain two 1-dim. correlation clusters. We observe a fast decrease of clustering quality for COPAC and ORCLUS with increasing number of views. SSCC constantly gets high quality results.

Scalability w.r.t. database size. Fig. 14 (left) shows the results for a varying number of objects in a dataset with 5 dimensions and 4 equally sized full space clusters without noise. All three algorithms scale linearly with the database size. Note the logarithmic scale of *both* axes. Although SSCC has to cope with an exponential number of subspaces, its runtime is still in range of COPAC's runtime.

Scalability w.r.t. dimensionality. For a dataset similar to scenario E (Fig. 9), except that all clusters comprise 500 objects, we consecutively concatenate the dataset to generate datasets of higher dimensionality in Fig. 14 (right). Although the number of subspaces grows exponentially with the number of dimensions, SSCC scales linearly. COPAC shows super linear behavior and thus exceeds SSCC's runtime. We again applied COPAC to any subspace projection to evaluate the *redundancy* and efficiency challenge. Already for the 10-dim. dataset COPAC needed over 4 hours and reported 1013 (redundant) clusters. Thus, removing redundancy, as done by SSCC, is also important for the efficiency.

Real world data. Due to space limitations, we only refer to the results of SSCC and COPAC in the following. For *Wages* (534 objects, 4 numerical attributes, 7 categories) we only use the four numerical attributes. Both algorithms only detected a single 2-dim. cluster, which is visualized in Fig. 15. All clustered objects are colored red, noise is colored blue. For the 2-dim. correlation in the Wages data, SSCC ($\epsilon = 0.002$, $\alpha = 0.85$, $minSize = 80$) captures the objects much better than COPAC. COPAC misses many objects since in the full-space they do not belong to this correlation.

For the *Image Segmentation* data we have 19 numerical attributes describing pixel regions and one class attribute. We removed the class attribute and the constant region-pixel-count attribute. Although, trying a wide range of parameter settings for COPAC, it was not able to detect any cluster for this dataset. For SSCC ($\epsilon = 0.002$, $\alpha = 0.85$, $minSize = 50$) two exemplary clusters out of the eight found ones are plotted in Fig. 16. Clearly, the clusters detected by SSCC correspond to strong correlations, which can be visually verified and are explainable from the dataset's description: the first two dimensions are the measures of excess green and excess blue that are defined as $exgreen.mean = (2G+(R+B))$ and $exblue.mean = (2B+(R+G))$. Here R denotes the average

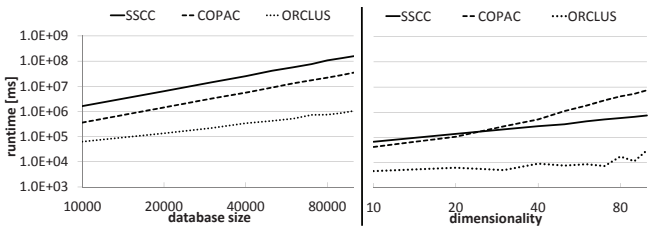

Fig. 14: Scalability: database size & dimensionality

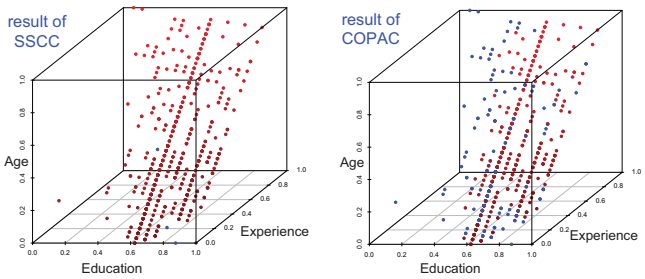

Fig. 15: Results of SSCC and COPAC for Wages

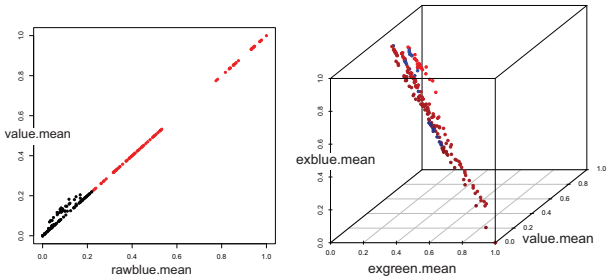

Fig. 16: SSCC results for Image Segmentation

over the raw values of red, B for blue, and G for green. The dimension *value.mean* is a 3-dim. non-linear transformation of the values R, G, and B. Obviously, from the definition of these dimensions we can expect some correlation in this subspace. This correlation is successfully identified by SSCC. Note that these correlations appear only for subsets of the attributes. Thus, full-space approaches are not able to detect these results, confirming the need for our novel subspace correlation clustering paradigm.

6. CONCLUSION

In this work, we have demonstrated that the observations of traditional subspace clustering analogously apply for the paradigm of correlation clustering. A simple post-processing to refine the clustering result determined in the full-space is not sufficient for typical subspace scenarios. Instead we have to analyze *subspace projections* of the data to find meaningful *strong correlations* supported by *subsets of objects*. Our introduced approach reveals linear correlations in subspace projections, allows objects to contribute to multiple correlations, and simultaneously ensures a result of manageable size containing only non-redundant subspace correlation clusters. For this, we carefully differentiate between non-redundant correlation clusters and ones originated due to collinearity or induction. We designed the efficient algorithm SSCC exploiting various pruning techniques. The experiments demonstrate that transferring ideas from the subspace clustering paradigm leads to more precise correlation clustering results compared to state of the art techniques in this domain.

As future work, we will extend our method to also handle non-linear correlations.

Acknowledgment. This work has been partly funded by the DFG grant SE1039/6-1 and by the UMIC Research Centre, RWTH Aachen University, Germany.

7. REFERENCES

[1] E. Achtert, C. Böhm, J. David, P. Kröger, and A. Zimek. Global correlation clustering based on the hough transform. *SADM*, 1(3):111–127, 2008.

[2] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek. Deriving quantitative models for correlation clusters. In *KDD*, pages 4–13, 2006.

[3] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek. On exploring complex relationships of correlation clusters. In *SSDBM*, page 7, 2007.

[4] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek. Robust, complete, and efficient correlation clustering. In *SDM*, 2007.

[5] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In *SIGMOD*, pages 70–81, 2000.

[6] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD*, pages 94–105, 1998.

[7] I. Assent, R. Krieger, E. Müller, and T. Seidl. Inscy: Indexing subspace clusters with in-process-removal of redundancy. In *ICDM*, pages 719–724, 2008.

[8] M. S. Aziz and C. K. Reddy. A robust seedless algorithm for correlation clustering. In *PAKDD (1)*, pages 28–37, 2010.

[9] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

[10] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is nearest neighbor meaningful? In *ICDT*, pages 217–235, 1999.

[11] C. Böhm, K. Kailing, P. Kröger, and A. Zimek. Computing clusters of correlation connected objects. In *SIGMOD*, pages 455–466, 2004.

[12] J. Cohen, P. Cohen, S. G. West, and L. S. Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge, 3rd edition, 2002.

[13] R. Freund, W. Wilson, and P. Sa. *Regression analysis: statistical modeling of a response variable*. Academic Press, 2006.

[14] S. Günnemann, E. Müller, I. Färber, and T. Seidl. Detection of orthogonal concepts in subspaces of high dimensional data. In *CIKM*, pages 1317–1326, 2009.

[15] R. Harpaz and R. M. Haralick. Mining subspace correlations. In *CIDM*, pages 335–342, 2007.

[16] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *TKDD*, 3(1), 2009.

[17] P. Kröger, H.-P. Kriegel, and K. Kailing. Density-connected subspace clustering for high-dimensional data. In *SDM*, pages 246–257, 2004.

[18] D. Niu, J. G. Dy, and M. I. Jordan. Multiple non-redundant spectral clustering views. In *ICML*, pages 831–838, 2010.

[19] A. Patrikainen and M. Meila. Comparing subspace clusterings. *TKDE*, 18(7):902–916, 2006.

[20] J. Yang, W. Wang, H. Wang, and P. S. Yu. delta-clusters: Capturing subspace correlation in a large data set. In *ICDE*, pages 517–528, 2002.

[21] X. Zhang, F. Pan, and W. Wang. Care: Finding local linear correlations in high dimensional data. In *ICDE*, pages 130–139, 2008.

[22] X. Zhang, F. Pan, and W. Wang. Redus: finding reducible subspaces in high dimensional data. In *CIKM*, pages 961–970, 2008.