In Proc. 20th Int. Conf. on Scientific and Statistical Database Management (SSDBM), Hong Kong, China, 2008

A General Framework for Increasing the Robustness of PCA-based Correlation Clustering Algorithms

Hans-Peter Kriegel, Peer Kröger, Erich Schubert, Arthur Zimek

Institute for Informatics, Ludwig-Maximilians-Universität München
http://www.dbs.ifi.lmu.de
{kriegel,kroegerp,schube,zimek}@dbs.ifi.lmu.de

Abstract. Most correlation clustering algorithms rely on principal component analysis (PCA) as a correlation analysis tool. The correlation of each cluster is learned by applying PCA to a set of sample points. Since PCA is rather sensitive to outliers, if a small fraction of these points does not correspond to the correct correlation of the cluster, the algorithms are usually misled or even fail to detect the correct results. In this paper, we evaluate the influence of outliers on PCA and propose a general framework for increasing the robustness of PCA in order to determine the correct correlation of each cluster. We further show how our framework can be applied to PCA-based correlation clustering algorithms. A thorough experimental evaluation shows the benefit of our framework on several synthetic and real-world data sets.

1 Introduction

Finding clusters in arbitrarily oriented subspaces is an important data mining task for many applications. The motivation behind this task is that in high dimensional data, one probably cannot find clusters due to several properties of high dimensional feature spaces. In contrast, clusters can usually be found in arbitrarily oriented subspaces of the original data space. The points of a subspace cluster are then located on a common lower dimensional hyperplane and exhibit a common correlation among a subset of the attributes. The task of finding clusters in arbitrarily oriented subspaces is also called correlation clustering.

The major challenge of correlation clustering is identifying the correct subspace of a cluster. Most correlation clustering algorithms [1–6] apply principal component analysis (PCA) to a subset of points in order to define the correct subspace in orientation and weighting of the transformed axes. PCA is a mature technique and allows the construction of a broad range of similarity measures grasping local correlation of attributes and, therefore, allows to find arbitrarily oriented subspace clusters. It is easy to see that the more points of this subset are cluster members that are located on the common hyperplane, the more accurate the procedure of determining the correct subspace (i.e. hyperplane) will be. However, a drawback common to all those approaches is the notorious *locality assumption*. Since cluster memberships of points are obviously not known

beforehand, it is assumed that the local neighborhood, e.g. the ε-neighborhood or the k-nearest neighbors, of cluster points or cluster centers represents the correct subspace suitably well in its orientation and variance along axes. This assumption is widely accepted but it boldly contradicts the basic problem statement, i.e. "find clusters in a high-dimensional space", because high dimensional spaces are typically doomed by the *curse of dimensionality*. The term "curse of dimensionality" refers to a bundle of problems occurring in high dimensional spaces. The most important effect in the sight of clustering is that concepts like "proximity", "distance", or "local neighborhood" become less meaningful with increasing dimensionality of a data set (as elaborated e.g. in [7–9]). As a consequence of these findings, the discrimination between the nearest and the farthest neighbor becomes rather poor with increasing data dimensionality. This is by far a more fundamental problem than the mere performance degradation of algorithms on high dimensional data: The higher the dimensionality of a data set is, the more outliers will be placed inevitably in the set of neighboring objects.

As we will see in this paper, PCA is very sensitive to outliers. In other words, if the local neighborhood of cluster members or cluster centers to which PCA is applied in order to find the correct subspace of the corresponding cluster contains noise points that do not belong to the cluster, the subspace determination process will be misled. Thus, in view of the "curse of dimensionality", to successfully employ PCA in correlation clustering in high-dimensional data spaces may therefore require more sophisticated techniques of selecting a representative set of neighbors.

In this paper, after shortly reviewing existing approaches to correlation clustering (cf. Section 2), we evaluate the influence of outliers on PCA in general (cf. Section 3) and propose a general framework to determine the correct local subspace dimensionality and orientation for cluster members and cluster centers in a more robust way (cf. Section 4). In Section 5, we show how to apply the proposed framework for increasing the robustness the subspace determination process on existing correlation clustering approaches. Section 6 demonstrates the impact of the increased robustness of PCA on several data sets. The paper is concluded in Section 7.

2 Related Work

The first approach to *generalized projected clustering*, called ORCLUS [1], is a K-means like approach. It picks $K_c > K$ seeds at first and assigns the data base objects to these seeds according to a distance function that is based on an eigensystem of the corresponding cluster assessing the distance along the small eigenvectors only (i.e., the distance in the projected subspace where the cluster objects exhibit high density). The eigensystem is iteratively adapted to the current state of the updated cluster (i.e., based on the current neighborhood of the cluster center). The number K_c of clusters is reduced iteratively by merging closest pairs of clusters until the user-specified number K is reached. The method proposed in [10] is a slight variant of ORCLUS designed for enhancing multi-

dimensional indexing. Initially, however, the eigensystems in both methods are based on the local neighborhood in the Euclidean space.

The algorithm 4C [2] is based on a density-based clustering paradigma [11]. Thus, the number of clusters is not decided beforehand but clusters grow from a seed as long as a density criterion is fulfilled. Otherwise, another seed is picked to start a new cluster. The density criterion is a required minimal number of points within the neighborhood of a point, where the neighborhood is ascertained based on distance matrices computed from the eigensystems of two points. The eigensystem of a point is based on the covariance matrix of the ε-neighborhood of the point in Euclidean space.

As a hierarchical approach, HiCO [4] defines the distance between points according to their local correlation dimensionality and subspace orientation – thus again based on a local neighborhood query – and uses hierarchical density-based clustering [12] to derive a hierarchy of correlation clusters.

COPAC [5] is based on similar ideas as 4C but disposes of some problems like meaningless similarity matrices due to sparse ε-neighborhoods instead taking a fixed number k of neighbors — which raises the question how to choose a good value for k but at least choosing $k > \lambda$ ensures a meaningful definition of a λ-dimensional hyperplane. Still, the Euclidean neighborhood critically influences the results.

The latest PCA-based correlation clustering algorithm is ERiC [6], also deriving a local eigensystem for a point based on the k nearest neighbors in Euclidean space. Here, the neighborhood criterion for two points in a DBSCAN-like procedure is an approximate linear dependency and the affine distance of the correlation hyperplanes as defined by the largest eigenvectors of each point. In finding and correctly assigning complex patterns of intersecting clusters, COPAC and ERiC improve considerably over ORCLUS and 4C.

Another approach based on PCA said to find even non-linear correlation clusters, CURLER [3], seems not restricted to correlations of attributes but, according to its restrictions, finds any narrow trajectory and does not provide a model describing its findings. However, even in this approach the PCA is applied to the local neighborhood of points in Euclidean space.

Note that the term "correlation clustering" relates to a different task in the machine learning community, where a partitioning of the data shall correlate as much as possible with a pairwise similarity function learned from past data [13].

3 Problem Analysis

To the best of our knowledge, all correlation clustering algorithms that use PCA as the method to determine the correct subspace of a cluster face the following problem. In order to determine the correct subspace of a cluster, a (considerably large) number of cluster members needs to be identified first such that PCA can be applied to them. On the other hand, in order to identify points of a particular cluster, the subspace of this cluster needs to be determined first. To escape from this vicious circle all algorithms rely on the locality assumption, i.e. it is

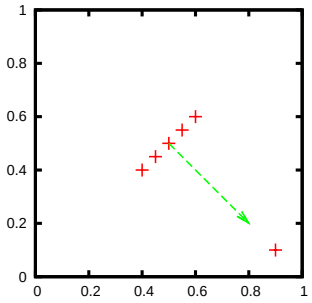


Fig. 1. Simple data set with 6 points and largest eigenvector after PCA.

assumed that the points in the local neighborhood of cluster members or cluster representatives sufficiently reflect the correct subspace of the corresponding cluster such that applying PCA to those neighboring points reports the cluster hyperplane.

As stated above, selecting a meaningful neighborhood becomes more and more difficult with increasing data dimensionality. A neighboring set of points will almost certainly contain outliers, i.e. points that do not belong to the cluster and, thus, are not located on the hyperplane of the cluster. Obviously, these outliers are not helpful to assign a meaningful local correlation dimensionality and orientation. On the other hand, all correlation clustering approaches available (cf. Section 2) rely on an arbitrarily chosen set of neighboring points. We therefore argue to choose a neighboring set of points in a more sophisticated way to enhance the robustness of local correlation analysis and, consequently, to enhance the robustness of correlation clustering algorithms.

3.1 Impact of Outliers on PCA

Correlation analysis using PCA is a *least squares fitting* of a linear function to the data. By minimizing the *mean square error*, outliers are emphasized in a way that is not always beneficial, as can bee seen in Figure 1. This data set consists of 5 points in a 2D space that are strictly positively correlated and, thus, are located on a common 1D hyperplane plus one additional outlier that is not located on that 1D hyperplane. When applying PCA on these six points and computing the strongest eigenvector of the corresponding covariance matrix, the resulting vector is directed towards the outlier (cf. Figure 1). This implies that in certain situations, adding only one single extra point to the correlation computation can cause the resulting strongest eigenvector(s) to flip into a completely different direction. Let us note that if the outlier point would have been closer to the other points it would, at a certain distance, not have made any difference on the vector orientation, but this distance threshold for the flip is rather small.

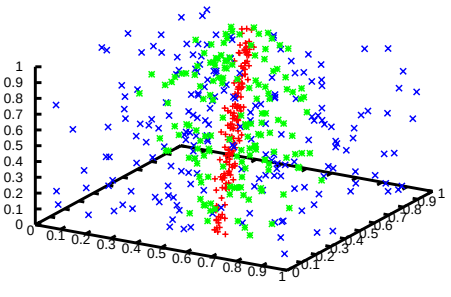


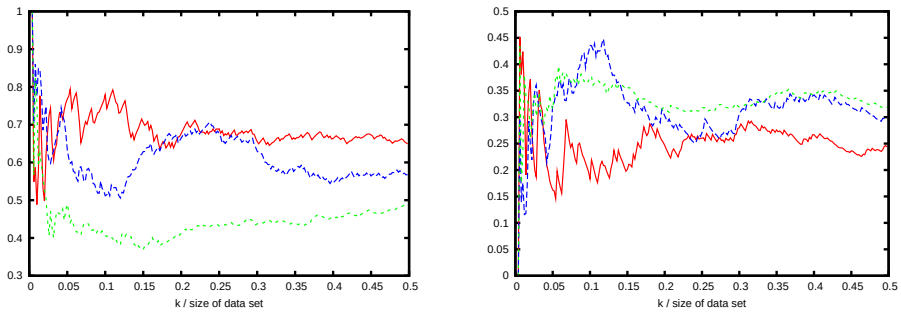
Fig. 2. Data set with a 2D plane and an embedded 1D line.

As a consequence, one needs to carefully select the points that are included into the computation of the cluster hyperplane. In addition, one can consider using a modified correlation analysis procedure which is less sensitive to the effect of outliers. In fact, there are obviously multiple strategies to handle these issues. The most obvious one – using outlier detection to remove outliers from the computation – can usually not be applied to this problem because we face the same vicious circle when searching for outliers as we face when detecting cluster points: in order to identify outliers that do not belong to any clusters, the subspaces of the clusters need to be determined first; in order to determine the correct subspace of a cluster, a (considerably large) number of cluster members needs to be identified first such that PCA can be applied to them; etc. Instead, we introduce two ideas to stabilize PCA for correlation clustering. First, we explore a local optimization strategy that handles the problem of picking appropriate neighboring points in a way that is easy to integrate in many correlation clustering algorithms. Second we will add a modified correlation analysis to further stabilize results which is based on the integration of a suitable weighting function into PCA.

3.2 Statistic Observations on Data Correlation

Without loss of generality, we assume that the points on which PCA is applied to find the correct subspace of a particular cluster are selected as the k-nearest neighbors (kNN) of cluster members or cluster representatives. Later, we will discuss the extension of our ideas to methods like ORCLUS that use neighborhood concepts other than kNN.

When comparing the relative strength of the normalized eigenvalues (i.e. the part of the total variance explained by them) computed for the kNN of a particular point w.r.t. increasing values of k (ranging from 0 to 50% of the data set), a behavior similar to that shown in Figure 3 can usually be observed. We used a 3D data set shown in Figure 2, with a set of 200 outlier points (noise), a



(a) First eigenvector (b) Second eigenvector

Fig. 3. Relative strength of eigenvectors.

correlation cluster of 150 points sharing a common 2D hyperplane (plane), and a correlation cluster of 150 points that are located on a common 1D hyperplane (line) that is embedded into the hyperplane of the 2D cluster. In Figure 3 there are three plots in this graph representing the behavior of the eigenvalues of a sample noise point, of a sample point on a 2D, and of a sample point on a 1D line in the data set (embedded within the 2D plane), respectively.

Examining the noise point (green dotted lines in Figure 3) we observe a minimum relative strength of the first eigenvalue of about 0.4 for $k = 10\% - 15\%$ (cf. Figure 3(a)). Since the minimum possible value for the strongest eigenvector in a 3D data set is $1/3 = 0.33$, the noise point shows approximately no correlation when looking at its kNN with $k = 10\% - 15\%$ of the data set. The second eigenvector (cf. Figure 3(b)) shows similar behavior in that particular range of k confirming our conclusions.

For the point in the 1D cluster (red solid lines in Figure 3), the first eigenvector (cf. Figure 3(a)) explains 80% of the complete variance at around $k = 7\%$, i.e. using this value for k, the kNN of the particular point form the 1D line of the cluster. It is worth noting that the amount of variance explained for the 1D cluster case drops quickly when increasing k beyond this point. The reason for this is that – since the line is embedded in a plane – with increasing k more and more points of the kNN are points from the 2D cluster. As a consequence, the variance explained by the first eigenvector decreases, whereas the variance explained by the second eigenvector increases simultaneously (cf. Figure 3(b)). Then, at $k \approx 10\%$, we have again a very high strength of the first eigenvector (less points from the 2D cluster and more points from the 1D cluster are considered), etc. In other words, depending on the value of k, the kNN of the point form the 1D cluster line or the 2D cluster plane.

For evaluating the 2D cluster, the relevant graph (depicting the behavior of the second eigenvector) is shown in Figure 3(b). In a 3D data set, a value of around 1/3 would be typical for uncorrelated data and is observable on noise points. For the sample point from the 2D cluster it peaks at almost 45% for about $k = 10\%$. Together with the first graph, this means that the first two

eigenvectors explain almost the complete variance at that particular value for k. In other words, for $k = 10\%$, the kNN of this point reflect the 2D plane of the cluster sufficiently. Compared to this observation, the variance of the sample point from the 1D cluster embedded in the 2D cluster (red dotted line) along the first two eigenvectors is significantly below the expected value (which is not surprising, having seen that the first eigenvector reaches 80%).

These simple examples illustrate that it is essential to select a sufficient set of points by choosing a suitable value for k. A slight change in k can already make a large difference. Moreover, we have seen that it is rather meaningful to choose even significantly different values of k for different points.

4 A General Framework for Robust Correlation Analysis

The above presented considerations induce two important aspects. First, since PCA is a least square fitting and we cannot assume that there are no outliers in the kNN of a point, adjusting the weighting of the points during PCA should improve the results. Second, the selection of points to which PCA is applied can be improved by both micro-adjusting the value of k (to avoid sudden drops in the explained variance) as well as choosing significantly different k for different points in the data set. In the following, we will discuss both aspects in more detail. In fact, our framework for making PCA-based correlation analysis more robust uses both ideas.

4.1 Increasing the Robustness of PCA Using Weighted Covariance

As mentioned above, PCA is a common approach to handling correlated data. It is also commonly used for dimensionality reduction by projecting onto the λ strongest (i.e. highest) components. In correlation clustering, PCA is a key method to finding correlated attributes in data.

PCA operates in two steps. In the first step, for any two attributes, i.e. dimensions, d_1 and d_2 the covariance $\mathrm{Cov}(X_{d_1}, X_{d_2})$ of these two dimensions is computed. In the second step, the eigenvectors and eigenvalues of the resulting matrix (which by construction is positive, symmetric and semi-definite) are computed. The computation of eigenvectors and eigenvalues on a symmetric matrix is a standardized procedure which cannot be altered to make the overall process more robust. Instead, the stabilization has to be implemented during the first step.

Given an attribute X, we can model the values of k points in that particular attribute, denoted by x_i for the i-th point, as a random variable. Then, the covariance between two attributes X and Y is mathematically defined as

$$\mathrm{Cov}(X, Y) := E((X - E(X)) \cdot (Y - E(Y))), \tag{1}$$

where E is the expectation operator. Usually, one uses the mean of all values of the corresponding attribute as expectation operator, i.e.

$$E(X) = \frac{1}{k}\sum_{i=1}^{k} x_i =: \hat{x}, \tag{2}$$

so we have

$$\mathrm{Cov}(X,Y) := \frac{1}{k}\sum_{i=1}^{k}(x_i - \hat{x})(y_i - \hat{y}). \tag{3}$$

Obviously, all data points are treated equally in this computation. But given that we want to reduce the effect of outliers, it is more appropriate to use a different expectation operator. Given arbitrary weights ω_i for all points i ($1 \leq i \leq k$ and $\Omega := \sum_{i=1}^{k} \omega_i$), we can define a new expectation operator

$$E_\omega(X) := \frac{1}{\Omega}\sum_{i=1}^{k} \omega_i x_i =: \hat{x_\omega}. \tag{4}$$

With this new expectation operator, we can give each point in kNN a different weight. In particular, we can give potential outliers a smaller weight. Using $E_\omega(X)$, we can compute the covariance as given below.

$$\mathrm{Cov}_\omega(X,Y) := \frac{1}{\Omega}\sum_{i=1}^{n} \omega_i(x_i - \hat{x}_\omega)(y_i - \hat{y}_\omega). \tag{5}$$

Steiner's translation still applies, which leads to the following slightly simpler equation.

$$\mathrm{Cov}_\omega(X,Y) = (\frac{1}{\Omega}\sum_{i=1}^{n} \omega_i x_i y_i) - (\frac{1}{\Omega}\sum_{i=1}^{n} \omega_i x_i)\cdot(\frac{1}{\Omega}\sum_{i=1}^{n} \omega_i y_i). \tag{6}$$

This form is particularly nice for computation. It is also trivial to prove that if $\omega_i = 1$ for all i, we have $\mathrm{Cov}(X,Y) = \mathrm{Cov}_\omega(X,Y)$. If a point i is assigned the weight $\omega_i = 2$, the result would be the same as if we had two points with the same coordinates as i. If a point i is weighted by $\omega_i = 0$, the result is the same as if point i had not been included in the computation at all.

We can now use arbitrary weighting functions to calculate the weights to be used. Obviously, we again have the dilemma that we do not know which points are outliers and need to get assigned a lower weight. However, since all algorithms use the locality assumption, we can make the following considerations: On the one hand, it is usually very likely that taking the local neighborhood of points includes a lot of outliers. But on the other hand, the neighbors that are near to the query point will more likely be cluster members than the neighbors that are farther apart from the query point. So a distance-based weighting function will most likely weight cluster points higher and outliers lower.

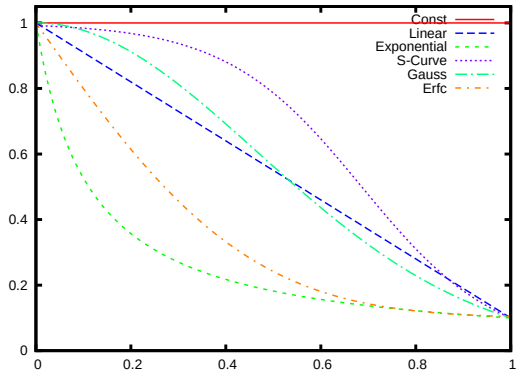
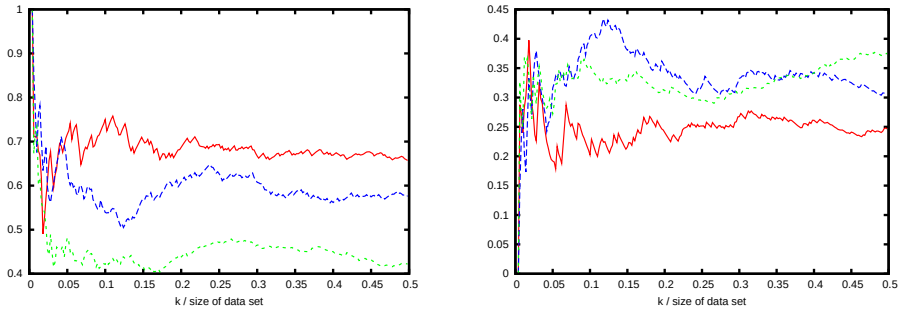


Fig. 4. Some weight functions.

Some examples of distance-based weighting functions are given in Figure 4. We have chosen parameters such that the value at $x = 0.0$ is about $f(0.0) \sim 1.0$ and at $x = 1.0$ it is about $f(1.0) \sim 0.1$. Weights too close to 0.0 are not very useful, because then, these points are not considered for the computation at all. The example weighting functions we have used in our experiments (cf. Figure 4) include a constant weighting of 1.0 (solid red line in Figure 4), a linearly decreasing function ranging from 1.0 to 0.1 (dashed blue line in Figure 4), an exponential fall-off (green dashed line in Figure 4), a sigmoid-curved fall-off (violet dotted line in Figure 4), a Gauss function (green dashed-dotted line in Figure 4), and the complementary Gauss Error Function *Erfc* (red dashed-dotted line in Figure 4). The last one is a function well-known from statistics related to normal distributions and, thus, probably the most sound choice.

In our experiments, all of the alternative weighting functions (except the constant weight) lead to similar improvements so there is no reliable measure or significance to establish a ranking between the different weighting functions. In fact, it is plausible that different functions are appropriate for different underlying causes in the data or assumptions in the clustering process (e.g. clustering algorithms assuming a Gauss distribution might benefit best from a Gaussian weighting function).

For distance-based weighting functions, several tasks arise. We have chosen to scale distances such that the outermost point has a distance of 1.0, i.e. a weight of 0.1, ensuring that this point has still some guaranteed influence on the result. This choice is somehow arbitrary, but it has at least the benefit of fairness. On the other hand, this fairness comes at the cost that all weights depend on the outermost point. When points are selected using a range query, the query range could offer a better normalization. When an incremental computation is desired, a completely different choice might be appropriate. Additionally, we are computing weights based on the distance to a query point. This is appropriate for situations where the data is obtained via kNN or ε-neighborhoods. When



(a) First eigenvector. (b) Second eigenvector.

Fig. 5. Relative strength of eigenvectors (with *Erfc* weight).

computing the correlation for an arbitrary set of points, the distance might need to be computed from the centroid or medoid of that set.

In the above described toy example of five cluster points plus one outlier (cf. Figure 1), the observed sensitivity to that outlier is significantly decreased, given that the outlier will only be weighted at around 0.1. Applying the weighting function to the 3D example data set of Figure 2 we also observe an increased robustness of the correlation analysis. Figures 5(a) and 5(b) depict the effect of a weighted covariance on the relative strength of the first eigenvector and the normalized sum of the first two eigenvectors, respectively, using the *Erfc* weighting. Compared to Figures 3(a) and 3(b) we can derive that many of the sudden drops have been erased, while the overall shape is well preserved. Especially for higher values of k, sudden jumps have mostly disappeared. Therefore, this measure is useful to avoid choosing a particularly bad value of k, i.e. a k where the kNN of the particular point do not reflect the correct subspace of the corresponding cluster, by somewhat averaging with neighbors. Peaks usually are shifted towards a slightly higher value of k. This is natural since the added points are weighted low at first.

4.2 Auto-tuning the Local Context of Correlation Analysis

Graphs such as Figure 3(a) show that even small differences in k can lead to significantly different results. Therefore, it is reasonable not to use a fixed value of k, i.e. a fixed number of neighboring points, but rather to adjust the value of k for each point separately. For example, one can use a globally fixed number of neighbors k_{max} and then individually select for each point the $k \leq k_{max}$ neighbors that are relevant for the particular point. As far as k_{max} is sufficiently large, we should in general be able to select a reasonable k, so that this strategy produces accurate results. Of course there are different strategies of selecting k. Since there are $O(2^{k_{max}})$ subsets of the given k_{max} points that could be used, simply trying all combinations of subsets of k points ($1 \leq k \leq k_{max}$) is not feasible. Probably the easiest strategy of $O(k_{max})$ complexity is to test for any

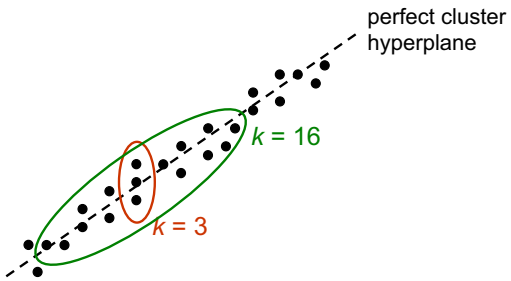


Fig. 6. Problems with jitter.

k ($1 \leq k \leq k_{max}$) only the k nearest points, resulting in k_{max} tests. The next question that arises is how to evaluate the results of the k_{max} tests in order to report the best value for k. The obvious strategy of returning the result that maximizes the relative strength of eigenvalues has shown to be not very reliable because of jitter: one particular k value could result in a "perfect" hyperplane consisting mainly of points that form a subspace completely different to the subspace of the cluster. Figure 6 illustrates this effect: using only the three points in the red ellipsoid, we will hardly find the correct hyperplane of the cluster although all those three points are cluster members because they do not fit the subspace perfectly. Rather, the three points perfectly form a different line so the relative strength of the first eigenvalue will be very high (appr. 100%). In fact, we are more interested in a range of k values where we have a high and stable relative strength of eigenvalues, so we need a more elaborate filtering.

In our evaluations, we have chosen the strategy to use the k nearest points for correlation analysis, with $k_{min} \leq k \leq k_{max}$, where k_{min} is a minimum number of points such that the PCA is at least somewhat sensible at this data set dimension. The motivation behind the introduction of the lower bound k_{min} is that we need at least λ points to span a λ-dimensional hyperplane and $3 \cdot \lambda$ has been considered as a lower bound of points such that the detection of a λ-dimensional hyperplane by PCA is trustworthy rather than arbitrary. To avoid jitter and outlier effects, we use a sliding window to apply a dimensionality filter and average the variance explained by the largest eigenvalues.

Let

$$\mathrm{ex}(E, \lambda) := \frac{\sum_{i=1}^{\lambda} e_i}{\sum_{i=1}^{d} e_i} \tag{7}$$

be the relative amount of variance explained by λ eigenvalues $E = \{e_i\}$ representing a hyperplane of dimensionality λ. Most correlation clustering algorithms rely on a level of significance $\alpha \leq 1$ to decide how many eigenvectors explain a significant variance and, thus, span the hyperplane of the cluster. Intuitively, the eigenvectors are chosen such that the corresponding eigenvalues explain more than α of the total variance. The number of those eigenvectors is called *local di-*

mensionality (of a cluster), denoted by λ_E, formally

$$\lambda_E = \min_{\lambda \in \{1 \ldots d\}} \{\lambda \mid \mathrm{ex}(E, \lambda) \geq \alpha\}. \tag{8}$$

Let us note that almost all correlation clustering algorithms use this notion of local dimensionality. Typical values for α are 0.85, i.e. the eigenvectors that span the hyperplane explain 85% of the total variance along all eigenvectors.

As indicated above, for filtering out the best value of k, we are intuitively interested in a value where (i) the local dimensionality λ is stable, i.e. increasing or decreasing k by a small degree does not affect the value of λ, and (ii) $\mathrm{ex}(E, \lambda)$ is maximal and stable, i.e. increasing or decreasing k by a small degree does not affect the value of $\mathrm{ex}(E, \lambda)$. The motivation behind these considerations is that the value of k that fulfills both properties leads to the determination of a robust hyperplane, that maximizes the variance along its axis. In other words, using the neighbors determined by k, the hyperplane reflects all of these neighbors in a best possible way and there are most likely only very few neighbors that are outliers to this hyperplane. In addition, increasing or decreasing k, i.e. adding or deleting few neighbors, does not affect the correlation analysis.

To find the value of k that meets both properties, we determine $\mathrm{ex}(E, \lambda)$ for all $k_{min} \leq k \leq k_{max}$. We then use a sliding window $W = [k_l, k_u]$ and choose $k = (k_l + k_u)/2$ such that for all k' in W (i.e. $k_l \leq k' \leq k_u$) the local dimensionality λ is the same and the average of $\mathrm{ex}(E_{k'}, \lambda)$ is maximized. Additionally, if this maximum is at the very beginning or end of our search range (i.e. $k_l = k_{min}$ or $k_u = k_{max}$), we discard it. We can still obtain multiple maxima, one for each dimensionality λ. In this case we pick the lowest like all correlation clustering algorithms aiming at finding the lowest dimensional subspace clusters. Those are the most interesting ones since they involve the largest set of correlations among attributes.

5 Application to Existing Approaches

In the following, we discuss how our concepts can be integrated into existing correlation clustering algorithms in order to enhance the quality of their results. Exemplarily, we show this integration with two different types of algorithms, the latest density-based algorithm ERiC and the k-means-based algorithm ORCLUS.

5.1 Application to Density-based Correlation Clustering Algorithms

The integration of our concepts into ERiC is rather straightforward. ERiC determines for each data point p the subspace of the cluster to which p should be assigned (hereafter called the subspace of p). The subspace of p is computed by applying PCA to the kNN of p where k needs to be specified by the user.

Using our concepts, we can simply replace the parameter k by the global maximum k_{max} of neighbors that should be considered. Both the weighting and

the auto-tuning can then be applied directly when computing the subspace of p. First, from the k_{max}NN of p, the optimal $k_p \leq k_{max}$ for detecting the subspace of p is determined as described in Section 4.2 based on a weighted covariance as described in Section 4.1. Second, the subspace of p is computed by applying PCA using a weighted covariance on the k_pNN of p (cf. Section 4.1).

The integration of our concepts into other density-based algorithms like COPAC, HiCO, and 4C can be done analogously.

5.2 Application to Partitioning Correlation Clustering Algorithms

ORCLUS determines the subspace of each cluster C by applying PCA to the local neighborhood of the center of C, denoted by r_C. The local neighborhood of r_C includes the set S_C of all points that have r_C as their nearest cluster representative.

Using our concepts, we can simply consider S_C as the maximum set of points that should be considered for PCA, i.e. $k_{max} = |S_C|$. Both the weighting and the auto-tuning can then be applied directly when computing the subspace of C. First, from the S_C, the optimal $k_C \leq k_{max}$ for detecting the subspace of C is determined as described in Section 4.2 based on a weighted covariance as described in Section 4.1. Second, the subspace of C is computed by applying PCA using a weighted covariance on the k_C points in S_C that are closest to r_C (cf. Section 4.1).

6 Experiments

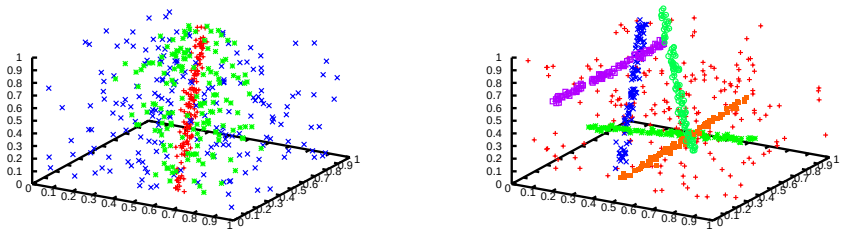
6.1 Evaluation Methodology

In order to evaluate the results of our novel concepts integrated into ERiC and ORCLUS, we generated artificial data sets with a well-defined gold standard, i.e. we defined certain data distributions and all points in our data set are assigned to the distribution with the maximum density in that particular point. Since both ERiC and ORCLUS have different properties and, here, we are not interested in judging which algorithm is better for which data set, we generated different data sets for each algorithm.

To evaluate the quality of the clustering, we employ a pair-counting F-measure, considering the noise points to be a cluster on its own. This means that any two points in the data set form a pair if they belong to the same cluster (or noise). Let $C = \{C_i\}$ be a clustering (with C_i being the clusters in C, including the noise cluster). Then $P_C := \{(a, b) \mid \exists C_i : a \in C_i \wedge b \in C_i\}$ is the set of pairs in clustering C. The F-measure to evaluate how good a clustering C matches the gold standard D is then defined as

$$F(C, D) := \frac{2 \cdot |P_C \cap P_D|}{2 \cdot |P_C \cap P_D| + |P_C \setminus P_D| + |P_D \setminus P_C|}.$$

Obviously, $F(C, D) \in [0, 1]$, where $F(C, D) = 1.0$ means that the clustering C is identical to the gold standard D.



(a) DS1: a 1D lines (150 points) embedded within a 2D plane (150 points) plus 200 points noise.

(b) DS2: five 1D lines (100 points each) plus 200 points noise.

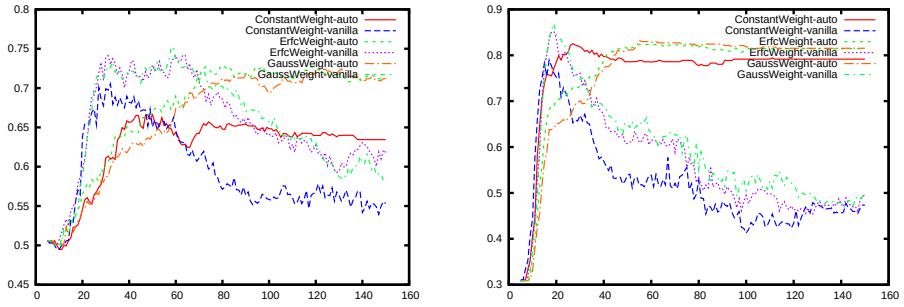
Fig. 7. 3D synthetic data sets used for evaluating ERiC.

6.2 Synthetic Data

For evaluating the influence of our novel methods on both ORCLUS and ERiC, we used several synthetic data sets ranging from 3 to 100 dimensions. In the following discussion, we focus on some lower dimensional data sets for a clear presentation.

ERiC. We first focus on two 3D synthetic data sets that can be seen in Figure 7. Figure 8(a) gives the results for data set DS1 shown in Figure 7(a). We plotted the F-measure of the compared algorithms along the y-axis and varied the parameters k and k_{max} along the x-axis. The blue line represents the results of the unmodified ERiC algorithm. Obviously the choice of k is nontrivial, a value of about $k = 34$ gives the best results. The violet dotted line is the result when using the Erfc weight in PCA. Obviously, the results are significantly better, and any k in $35 < k < 65$ gives good results. Therefore choosing a good k has become a lot easier using only the weighting approach. The green line with the short dash-dot pattern depicts the result of ERiC using a Gauss weight. As it can be seen, the results using a simple Gaussian weighting do not significantly differ from the Erfc weighting results.

The remaining three lines show the results of ERiC when using the auto-tuning of the parameter k, i.e. for each point the optimal $k \leq k_{max}$ is determined separately (for these graphs, the x-axis represents the chosen k_{max} value). The red line is using the traditional PCA without any weighting, while the dashed green and the orange line with the long dash-dot pattern represent the results using the Erfc and Gauss weights, respectively. The results show that k_{max} simply needs to be chosen high enough in order to achieve reasonably good results. While these results do not reach the results of choosing the optimum k (which is not possible without knowing the gold standard), they approach the optimal value quite well. This observation dramatically simplifies the choice of the k/k_{max} parameter.



(a) DS1. (b) DS2.

Fig. 8. Results of ERiC with different weight functions and auto-tuning.

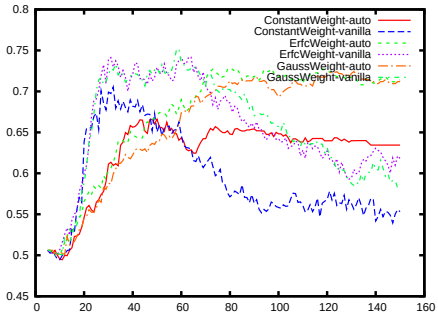


Fig. 9. Results of ERiC with different weight functions and auto-tuning on a sample 10D synthetic data set.

Figure 8(b) depicts the results on DS2 shown in Figure 7(b). Overall, the results on DS1 and DS2 are comparable. However, given that every point has just one "sensible" dimensionality – the other data set had points that had both a sensible 1D and 2D context – and the noise level is not as high, the effect of the weighted PCA on DS2 is not as high as on DS1. Since increasing the noise level will increase the difference between the non-weighted and weighted graphs, the weighting is especially interesting for noisy (e.g. higher dimensional) data.

All observations that could be made for the two 3D data sets could also be made for higher dimensional data sets. For example, Figure 9 shows the results of ERiC with different extensions for a sample 10D data set. Again, the version of ERiC using an Erfc weighted PCA in combination with the auto-tuned selection of k achieved the best overall F-measure. Also, as long as k_{max} is chosen sufficiently high, we get rather accurate results.

In summary, we observed that in all cases, the combination of the Erfc weighted PCA and the auto-tuned selection of k considerably increased the F-measure of the resulting clustering and significantly reduced the complexity of selecting sufficient input parameters compared to the original ERiC algorithm.

Table 1. Impact of the integration of our novel concepts into ORCLUS.

Variant	**Average F-measure**	**Standard Deviation**
ORCLUS	0.667	0.046
ORCLUS + Gauss weight	0.684	0.055
ORCLUS + Exponential weight	0.676	0.054
ORCLUS + Erfc weight	0.683	0.061
ORCLUS + Linear weight	0.686	0.056
ORCLUS + Auto	0.751	0.070
ORCLUS + Auto + Gauss	0.763	0.069
ORCLUS + Auto + Exponential	0.754	0.075
ORCLUS + Auto + Erfc	0.754	0.075
ORCLUS + Auto + Linear	0.771	0.078

Table 2. Results on NBA data using ERiC with autotuning and Erfc weighting.

cluster ID	dim	Description
1	4	"go-to-guys"
2	4	guards
3	4	reserves
4	5	small forwards

ORCLUS. The results of ORCLUS are harder to evaluate, because the results of ORCLUS depend on the order in which the data points are processed. Therefore, we generated 100 permutations of the original data, applied ORCLUS with optimal parameters to all of them, and averaged the results. The data set used in these computations was a 10-dimensional data set, containing 10 clusters of dimensionalities 2 to 5. The results are given in Table 1.

Each of these values was obtained by running ORCLUS and its variants on the same 100 permutations of the input data set and averaging the resulting F-measure values. The standard deviation over the 100 resulting F-measure values is given to show the dependence of ORCLUS on picking good seeds. It can be observed that the benefits of using a weighted PCA are smaller (≈ 0.02) than those of using an auto-tuning PCA (≈ 0.09) and the combination of both actions further improves the results. Interestingly, in this experiment, a linear weighting function is slightly better (by up to 0.02) than a Gaussian or Erfc weighting. However, in general on different data sets, there is no significant difference observable comparing different weighting functions. In summary, using our novel concepts, the F-measure on this data set is improved by approximately 0.1 corresponding to a 10% quality boost.

6.3 Real-world Data

We applied the enhanced version of ERiC (using autotuning and Erfc weighting) on a data set containing average career statistics of current and former NBA

Table 3. Results on Metabolome data using ERiC with autotuning and Erfc weighting.

cluster ID	dim	Description
1	10	PKU
2	10	controll
3	11	PKU
4	12	PKU
5	13	PKU

players[1]. The data contains 15 features such as "games played" (G), "games started" (GS), "minutes played per game" (MPG), "points per game" (PPG), etc. for 413 former and current NBA players. We detected 4 interesting clusters each containing players of similar characteristics (cf. Table 2). In addition, several players were assigned to the noise set. Cluster 1 contains active and former superstars like Michael Jordan, Allen Iverson, Larry Bird, Dominique Wilkins, and LeBron James, etc. The second cluster features point and shooting guards. A third cluster contains only very few players that are not so well-known because they are usually reserves. The fourth cluster consists of small forwards. Let us note that we also applied the original ERiC algorithm (without the extensions) to the NBA data set but could not get any clear clusters. In summary, using our novel concepts, the algorithm ERiC is now able to detect some meaningful clusters on the NBA set.

In addition, we applied our novel concepts in combination with ERiC to the Metabolome data set of [14] consisting of the concentrations of 43 metabolites in 20,391 human newborns. The newborns were labelled according to some specific metabolic diseases. The data contains 19,730 healthy newborns ("control"), 306 newborns suffering from phenylketonuria ("PKU"), and 355 newborns suffering from any other diseases ("other"). The results are depicted in Table 3. As it can be seen, we could separate several of the newborns suffering from PKU from the other newborns. Again, the original version of ERiC could not find any comparatively good results.

7 Conclusion

Almost all correlation clustering algorithms suffer from an arbitrary selection of points in the local neighborhood of cluster members or cluster representatives from which the subspace of a cluster is determined by applying PCA. Choosing the local neighborhood, is a heuristics that is usually more meaningfull than random sampling. However, due to outliers in this selection, the process of finding the correct subspace is often misled because PCA is rather sensitive to outliers. In this paper, we discuss general concepts to enhance the robustness of PCA for finding the correct subspace in order to increase the effectiveness of any PCA-based correlation clustering algorithm. Thereby, we do not solve the problem of

[1] obtained from `http://www.nba.com`

making a more suitable selection rather than the local neighborhood but try to ease the influence of outliers in this local neighborhood by a two-step approach: First, a weighting function is applied to the points when computing the covariance matrix for PCA in order to weight points that are potential outliers lower than points that are potential cluster members. Second, a method for selecting a suitable number of neighbors for each cluster member or cluster representative separately is presented. We further discuss how our general method can be integrated into existing correlation clustering algorithms based on different cluster paradigms. Our experiments show that the quality of the corresponding results can be significantly enhanced when using our new methods. In addition, the experiments show that our approach remarkably simplifies the selection of critical parameters. In summary, our method considerably enhances the robustness and usability of correlation clustering algorithms.

References

1. Aggarwal, C.C., Yu, P.S.: Finding generalized projected clusters in high dimensional space. In: Proc. SIGMOD. (2000)
2. Böhm, C., Kailing, K., Kröger, P., Zimek, A.: Computing clusters of correlation connected objects. In: Proc. SIGMOD. (2004)
3. Tung, A.K.H., Xu, X., Ooi, C.B.: CURLER: Finding and visualizing nonlinear correlated clusters. In: Proc. SIGMOD. (2005)
4. Achtert, E., Böhm, C., Kröger, P., Zimek, A.: Mining hierarchies of correlation clusters. In: Proc. SSDBM. (2006)
5. Achtert, E., Böhm, C., Kriegel, H.P., Kröger, P., Zimek, A.: Robust, complete, and efficient correlation clustering. In: Proc. SDM. (2007)
6. Achtert, E., Böhm, C., Kriegel, H.P., Kröger, P., Zimek, A.: On exploring complex relationships of correlation clusters. In: Proc. SSDBM. (2007)
7. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful? In: Proc. ICDT. (1999)
8. Hinneburg, A., Aggarwal, C.C., Keim, D.A.: What is the nearest neighbor in high dimensional spaces? In: Proc. VLDB. (2000)
9. Aggarwal, C.C., Hinneburg, A., Keim, D.: On the surprising behavior of distance metrics in high dimensional space. In: Proc. ICDT. (2001)
10. Chakrabarti, K., Mehrotra, S.: Local dimensionality reduction: A new approach to indexing high dimensional spaces. In: Proc. VLDB. (2000)
11. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. KDD. (1996)
12. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: Ordering points to identify the clustering structure. In: Proc. SIGMOD. (1999)
13. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Mach. Learn. **56** (2004) 89–113
14. Liebl, B., Nennstiel-Ratzel, U., von Kries, R., Fingerhut, R., Olgemöller, B., Zapf, A., Roscher, A.A.: Very high compliance in an expanded MS-MS-based newborn screening program despite written parental consent. Preventive Medicine **34**(2) (2002) 127–131