to appear in:
Proceedings of the 12th IEEE International Conference on Data Mining (ICDM),
Brussels, Belgium, 2012.

Outlier Detection in Arbitrarily Oriented Subspaces

Hans-Peter Kriegel, Peer Kröger, Erich Schubert
Ludwig-Maximilians-Universität München
Munich, Germany
http://www.dbs.ifi.lmu.de
Email: {kriegel,kroeger,schube}@dbs.ifi.lmu.de

Arthur Zimek
Department of Computing Science
University of Alberta, Edmonton, AB, Canada
http://webdocs.cs.ualberta.ca/~zimek/
Email: zimek@ualberta.ca

***Abstract*—In this paper, we propose a novel outlier detection model to find outliers that deviate from the generating mechanisms of normal instances by considering combinations of different subsets of attributes, as they occur when there are local correlations in the data set. Our model enables to search for outliers in arbitrarily oriented subspaces of the original feature space. We show how in addition to an outlier score, our model also derives an explanation of the outlierness that is useful in investigating the results. Our experiments suggest that our novel method can find different outliers than existing work and can be seen as a complement of those approaches.**

I. Introduction

Outlier detection is important in many applications including e.g. the detection of credit card abuse in financial transactions data, the identification of measurement errors in scientific data, or the recognition of exceptional protagonists in athletic statistics, etc. Most of the recent work in outlier detection refer to Hawkins' definition of an outlier as "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism" [1]. Several outlier detection schemata have been proposed over decades differing widely in how an outlier is modelled and, thus, offer different features applicable to varying scenarios. Here, we focus on the unsupervised variant of the problem.

While in the early years of data mining, the features of data sets were carefully selected, nowadays we are entering the area of *Big Data*. There is a trend of measuring as much parameters as possible because modern capabilities of data generation produce data at low costs. Thus, a real world data set usually contains several groups of observations (i.e., data objects) that have been generated by different (typically unknown) mechanisms or statistical processes. These different generating mechanisms may show their effects in varying subsets of attributes, i.e. a varying subset of features is correlated differently for each generating mechanism defining a *local correlation* of features for the corresponding subset of data objects generated. Typically, a mechanism is supposed to have generated a minimum number of data objects in order to be considered as significant. Outliers are those objects that have not been generated by these mechanisms, i.e. those objects that do not fit into the corresponding local correlations. The subset of points that show a local correlation are located on a common δ-dimensional hyperplane, where $\delta < d$ and d is the dimensionality of the full-dimensional data space. As a consequence, outliers are those objects that are not located on those hyperplanes rather than objects that are in a less dense area in the full-dimensional data space. In addition, in such a scenario, the task of outlier detection is not only to identify possible outliers but also—in order to find outliers at all—to automatically determine the local correlations from which objects deviate. Almost all existing approaches for outlier detection implicitly rely on the assumption that all features are equally relevant for detecting outliers and do not account for local correlations nor for determining those local correlations.

This general idea and the difference of this idea to existing approaches is visualized in Figure 1(a). A sample 3D data set is shown that has been generated by three mechanisms each following a unique correlation of (a subset of) features. Existing approaches consider the full-dimensional space and would most likely find objects n_1 and n_2 as outliers because these objects deviate from all other objects significantly when considering vicinity in the 3D space. Objects o_1, o_2, and o_3 on the other hand would most likely not be detected because they do not deviate from the other objects conspicuously if the full-dimensional space is considered. However, objects n_1 and n_2 are located on one of the hyperplanes that represent the correlations of the corresponding generating mechanisms and, thus, should not be seen as outliers; these objects fit perfectly to the mechanisms that have generated the normal instances. Rather, objects o_1, o_2, and o_3 are outliers because they deviate considerably from any of the hyperplanes representing the generating mechanisms. We can find these outliers only when considering the subspaces that are perpendicular to the hyperplanes of the generating mechanisms. For example, in the subspace that is perpendicular to the hyperplane representing mechanism 1 all objects generated by mechanism 1 (including n_2) are dense and o_1 deviates considerably from those objects. Thus, in such a scenario, existing outlier detection approaches may incorrectly classify normal instances as outliers and may miss true outliers because these approaches do not take any local correlations into account.

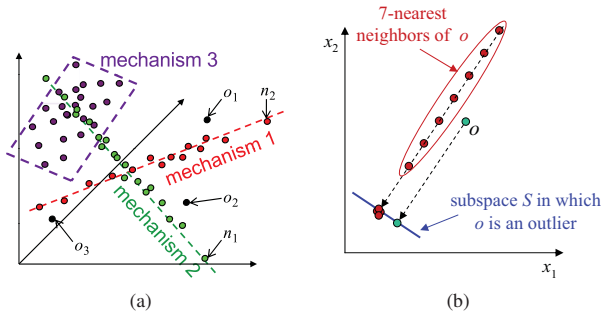


Figure 1. The general idea of finding outliers in subspaces of the original feature space by projecting the data to the orthogonal subspace S.

In this paper,[1] we introduce an outlier model that detects outliers as points that do not fit to any significant local correlation in the data. Orthogonal to existing methods, this model is the first approach to consider local correlations within the outlier detection process. We simultaneously identify outliers in arbitrarily oriented subspaces of the original feature space and determine the relevant correlation of attributes that needs to be considered to detect the corresponding outlier. This is obviously beneficial in many applications, e.g. in scientific domains where the relationship between causation and effect can only be exploited when considering correlations among attributes. Our new model should be seen as a complement of the set of existing approaches rather than as a strict rival because it computes a completely different type of outliers under completely different assumptions. The benefit of having quite different rationals and models to derive outliers has been demonstrated for the construction of outlier detection ensembles [4].

The remainder is organized as follows. We review related work in Section II. Our novel outlier model is described in Section III. An experimental evaluation of the proposed method is presented in Section IV. Section V provides final conclusions.

II. Related Work

Existing unsupervised outlier detection approaches can be classified as global or local. A global model is based on properties compared over the complete data set assuming one global generating mechanism underlying the normal instances. A local outlier approach considers a local selection of the data set which seems better suitable when multiple generating mechanisms exist.

Existing outlier detecting methods differ in the way they model and find the outliers and, thus, in the assumptions they rely on, implicitly or explicitly. In statistics, outlier detection is usually addressed by modelling the generating mechanism(s) of the normal data instances using a single or a mixture of multivariate Gaussian distribution(s) and measuring the Mahalanobis distance to the mean(s) of this (these) distribution(s). Barnett and Lewis [5] discuss numerous tests for different distributions in their classical textbook. As a rule of thumb, objects that have a distance of more than $3 \cdot \sigma$ to the mean of a given distribution (σ denotes the standard deviation of this distribution) are considered as outliers to the corresponding distribution. The data mining community developed many different approaches that have less statistically oriented but more spatially oriented notions to model outliers. Distance-based outlier models consider the number of nearby objects or the distances to nearby objects as an indication of the outlierness of an object [6]–[11]. Angle-based outlier scores like ABOD [12] assess the variance in angles between an outlier candidate and all other pairs of points. Density-based approaches compare the density of each object with the density of its neighbors [13]–[18]. However, all these approaches rely implicitly on the assumption that a globally fixed set of features (usually all available attributes) are equally relevant for the outlier detection process.

Some approaches try to account for a local feature relevance and search outliers in axis-parallel subspaces of the data space [19]–[27]. This is obviously a special case of finding outliers in arbitrarily oriented subspaces. The most recent of these approaches, HiCS [27], uses a Monte-Carlo sampling approach to detect interesting feature combinations, then runs an existing outlier detection method (in the article they experiment with LOF [13], but suggest that any other method will be usable) in each such subspace. The feature selection is in fact global, locality is introduced by the method used within the chosen projections. A recent survey [3] is discussing subspace outlier detection in a broader perspective.

So far, no outlier detection approach is considering *local* correlations of attributes for outlier detection and searches for outliers in arbitrarily oriented subspaces, analyzing the *locally* relevant feature combinations only.

Outlier detection is orthogonal to clustering [28] where the aim is to find a natural grouping of sets of similar data objects, i.e. the generating mechanisms of a data set. In fact, clustering algorithms have similar problems like outlier detection approaches in the above described scenarios. Thus, a plethora of specialized methods has been proposed for the detection of clusters in subspaces of the data space (see e.g. the surveys [29]–[31]), some taking into account local correlations, e.g., [32]–[34]. Although outliers can be seen as objects that do not fit well into any cluster, clustering algorithms can usually not be used for outlier detection because these algorithms search for clusters and their corresponding subspaces rather than outliers and their corresponding subspaces. Object that are not assigned to any subspace cluster need not necessarily be remarkable outliers in any of the subspaces in which the detected clusters exist.

[1]This paper presents an improved version of the method COP as discussed in [2, chap. 18]. See also the discussion in a recent survey [3].

III. Outlier Detection in Subspaces

A. General Idea

In the following, we assume $\mathcal{D} \subseteq \mathbb{R}^d$ to be a database of n feature vectors in a d-dimensional space generated by several mechanisms (i.e. statistical processes). Data objects from the same generating mechanism are assumed to show a similar correlation among some attributes such that they are located on a common δ-dimensional ($\delta \leq d$) hyperplane, hereafter also called *correlation hyperplane*. The basic idea of our approach is that a data point o is an outlier w.r.t. a set of "normal" reference objects $N \subset \mathcal{D}$ if o is not located on the hyperplane spanned by the points of N. If the objects in N and o itself are projected on the (arbitrarily oriented) subspace perpendicular to the hyperplane spanned by the objects in N, we can observe that the objects in N exhibit a high density in that subspace whereas o is considerably far apart from these objects. This idea is visualized in Figure 1(b). Thus, we consider a set of reference objects N for an object o in order to evaluate the outlier degree of o w.r.t. N similar to existing local outlier detection approaches. However, fundamentally different to existing approaches, we do not consider the density of o and the density of the neighbors of o in the full-dimensional space. Rather, we determine the correlation, i.e. the hyperplane, defined by the neighbors of o and evaluate the deviation of o to its neighbors in the subspace perpendicular to that hyperplane. This procedure measures how good o fits to this correlation. The motivating idea for this approach is the assumption of possible dependencies among different attributes [35]. Different mechanisms that have generated the data will then most likely exhibit also different sets of dependencies among attributes. Eventually, these dependencies are also interesting themselves in order to grasp possible underlying mechanisms, since those mechanisms are presumably unknown *a priori* and local.

In the following, we first introduce a concept to describe local correlation models (Section III-B) which are used to determine our outlier scores. In Section III-C we discuss how to obtain a useful outlier score from the model. We discuss the choice of the local reference set N in Section III-D and present a method to derive an explanation for the found outliers in Section III-E. Finally, a short description of the outlier detection algorithm and a discussion of its properties completes this Section (Section III-F).

B. Local Correlation Models

Correlations in 2 dimensions are commonly measured using Pearson correlation and more generally by using covariance. To abstract this to arbitrary dimensionality, it is common to use the *covariance matrix* $\boldsymbol{\Sigma}_N$ of N, where $\sigma_{ij} := \text{Cov}(X_i, X_j)$ for attributes X_i and X_j. Prime examples are Mahalanobis distance which is defined as:

$$d_M(x, \mu) := \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

and Principal Component Analysis (PCA), which decomposes the covariance matrix into a rotation matrix V containing the eigenvectors and a diagonal matrix Λ containing the associated eigenvalues in descending order such that

$$V \Lambda V^{-1} = \Sigma$$

Using this decomposition yields the following formulation of Mahalanobis distance of x from mean μ:

$$d_M(x, \mu) := \sqrt{(x-\mu)^T V \Lambda^{-1} V^{-1} (x-\mu)}$$

The eigenvalues λ in Λ correspond to the variances along the individual eigenvectors and sum up to the total variance of the original data, Var(N). If a diagnonal matrix Ω is defined using $\omega_i := \sqrt{1/\lambda_i} = \lambda_i^{-\frac{1}{2}}$, then $\Omega\Omega = \Lambda^{-1}$. Since V is a rotation matrix, $V^{-1} = V^T$, and since Ω is a diagonal matrix, $\Omega = \Omega^T$, therefore

$$\begin{aligned} d_M(x, \mu) &:= \sqrt{(x-\mu)^T V^{TT} \Omega^T \Omega V^{-1} (x-\mu)} \\ &= \sqrt{(\Omega V^T (x-\mu))^T \Omega V^T (x-\mu)} \\ &= L_2(\Omega V^T (x-\mu)) \end{aligned}$$

As we can see from this (well-known) decomposition, Mahalanobis distance is closely related to Euclidean norm L_2, weighted by the eigenvalues along the principal components.

The eigenvectors in V describe the primary axes of the data set, ordered with decreasing eigenvalues (variance). If there is a strong correlation in the data set, the first eigenvectors will be directed along this variance, while the remaining eigenvectors are orthogonal and can be seen as describing the deviation from the data set. For our local correlation model, we want to exploit this property. By replacing Ω with $\hat{\Omega}_\delta$ ($\delta \leq d$) using

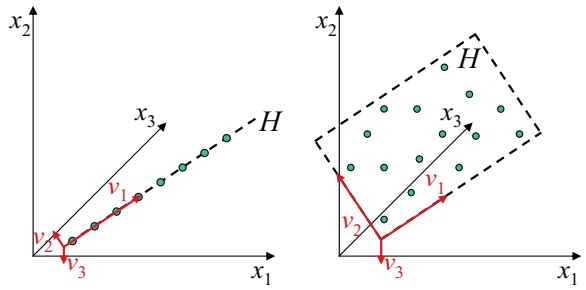
$$\hat{\omega}_{\delta,i} := \begin{cases} 0 & \text{iff } i \leq \delta \\ \omega_i = \lambda_i^{-\frac{1}{2}} & \text{iff } i > \delta \end{cases}$$

we obtain distances that do not take the first δ projected dimensions into account. Assuming that the local data is δ dimensional,

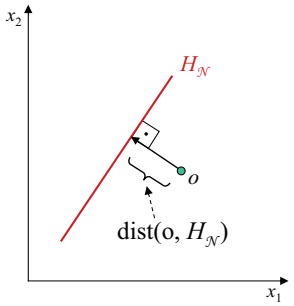
$$d_{\text{E},\delta} := L_2(\hat{\Omega}_\delta V^{-1} (x-\mu)) \tag{1}$$

is therefore a good measure of deviation from the local correlation. We call this δ the *correlation dimensionality* of N. The correlation dimensionality is closely related to the intrinsic dimensionality of the data distribution. If, for instance, the points in N are located near a common line, the correlation dimensionality of these points will be approximately 1. The difficult part is to determine this correlation dimensionality δ. In [36] the authors propose to use a threshold α, and consider those dimensions such that they explain the fraction α of the total variance:

$$\text{argmin}_\delta \sum_{i=1}^{\delta} \lambda_i \geq \alpha \text{Var}(N)$$



(a) One dimensional correlation (b) Two dimensional correlation



(c) Distance from hyperplane H_N

Figure 2. Illustration of the correlation dimension and the distance of an object o to a hyperplane H_N.

where $\alpha = 85\%$ showed good performance. However, this is a rather crude heuristic. In particular, there may be more than one correlation (of different dimensionality) present in the same set. A weaker one-dimensional (linear) correlation can be embedded within a higher dimensional manifold. Therefore, using a threshold may not necessarily be adequate. Additionally, experiments have shown that even in uncorrelated data, in particular when the sample size is low and the dimensionality is high, there will be a significant difference in the eigenvalues that may result in an incorrect dimensionality. In an extreme case, when $|N| < d$ there can obviously be at most $|N| - 1$ non-zero eigenvalues.

These ideas are illustrated in Figure 2. Figure 2(a) shows a set of points N that span a correlation hyperplane H of correlation dimensionality 1 corresponding to a (perfect) line. One eigenvector (v_1) already explains the total variance of N. The projection to the first eigenvector describes the position of the object within the subspace, while the projections on the other eigenvectors describe the deviation from the subspace. Figure 2(b) shows a set of points N that span a correlation hyperplane H of correlation dimensionality 2 corresponding to a (perfect) plane. Here, two eigenvectors are needed to represent the position of the object, while the third would determine the deviation.

Let us note that in the displayed examples the correlations are perfect, i.e. there is no deviation from the hyperplane but all points within the set perfectly fit to the hyperplane. In real-world data sets, this is obviously a rather unrealistic scenario. The points will most likely deviate from the (idealized) hyperplane. However, for a known correlation dimensionality δ we can measure the deviation from the correlation using Equation 1. It may seem a bit counter intuitive to use the last vectors of the PCA result. The reason is that the first (high variance) vectors will describe the extend of the correlation hyperplane, while the orthogonal (low variance) vectors describe the deviation, which we are interested in here. This is visualized in Figure 2(c), where the distance of an object o to a hyperplane H_N is clearly to be measured orthogonally to the hyperplane. Note that we do not explicitly compute or use the hyperplane H_N in the following, but it is implicitly encoded in the distance function.

In the first components (those spanning the subspace) the data may be arbitrarily distributed. The remaining components, however, seem to contain errors, which we can intuitively assume to follow approximately a normal distribution if they are indeed not relevant.

The projection onto the last $d - \delta$ eigenvectors also provides a very useful tool for evaluating the outliers found: it is the difference vector between the true object location and the idealized position the object was expected to be at, which can easily be used as explanation for the outlierness of an object, which we will use in Section III-E.

C. Correlation Outlier Probability

Even if we know the appropriate correlation dimensionality δ, the raw distances are not well suited for outlier detection. A key improvement of LOF [13] over preceding outlier detection methods such as the method of Knorr and Ng [6] was to compare the distances associated with one object to the distances of neighbor objects.

LoOP [18] is a variation of LOF that uses a local statistical density estimation to become less sensitive to the choice of the size of the neighborhood. By using normalization and regularization, the outlier scores become also interpretable as probabilities. A general framework for regularization of arbitrary methods is discussed in [37].

Here, it is not reasonable to use the distance for density estimation like in LOF, but instead we can assume that the error distances correspond to a $d - \delta$ dimensional normal distribution. After rescaling the data with $\hat{\Omega}_\delta$, we can even assume the individual dimensions to be approximately independent and identically distributed (i.i.d.), i.e. $\forall_{i>\delta} \omega_i v_i^T (x - \mu) \sim \mathcal{N}(0,1)$. Then the deviation $d_{\mathrm{E},\delta}$ however is $\chi(d - \delta)$ distributed, as it is:

$$d_{\mathrm{E},\delta}(x,\mu) = \sqrt{\sum_{i=\delta+1}^{d} (\omega_i v_i^T (x-\mu))^2}$$
$$\sim \sqrt{\sum_{i=\delta+1}^{d} \mathcal{N}(0,1)^2}$$
$$\sim \chi(d-\delta)$$

For convenience and simpler computations, we can also look at the squared distances, which then are $\chi^2(d-\delta)$ distributed (a special case of the Γ distribution), i.e.

$$d_{\mathrm{E},\delta}(x,\mu)^2 \sim \chi^2(d-\delta) \tag{2}$$
$$\sim \Gamma\left(\frac{d-\delta}{2}, 2\right) \tag{3}$$

To improve accuracy, we can also try to fit a Gamma distribution to the observed squared distances, instead of relying on the data to be exactly χ^2 distributed. We used the maximum likelihood estimation by [38], but only applied it to the 85% closest points, to avoid outliers from influencing the parameter estimation.

For these distributions, the cumulative density function (CDF) measures how many objects are expected to be closer to the mean than the given distance. This is a very intuitive value, ranging from 0 to 1, representing the probability, that a random generated object has a smaller distance than the observed instance.

At this point, we can now define the *Correlation Outlier Score* by taking the maximal unlikely deviation, and this way implicitly choosing the correlation dimensionality δ.

Definition 1 (Correlation Outlier Score):
Formally, the *Correlation Outlier Score* is defined as

$$\mathrm{COS}(o) := \max_{\delta} \mathrm{cdf}_{\Gamma}(d_{\mathrm{E},\delta}(x-\mu)) \tag{4}$$

Note that the exact shape of the Γ distribution depends on the dimensionality $\delta - d$, and for improved results can be estimated from the observed distances (for each δ).

While this value can already be seen as an "outlier probability" as suggested by [37], it is not yet entirely intuitive. Obviously, 10% of objects are expected to have a probability higher than 90% and 1% higher than 99%. This linear behaviour of the score does not align well with the intuitive use, where we would expect objects with a score of more than 50% to be more likely outliers than inliers, and this should be a much more rare occurrence than 50%.

To obtain this more intuitive probability, we adjust the score in the style of hypothesis testing and compare the hypothesis that the object is *normally distributed* to the alternate hypothesis of the object coming from an arbitrary different distribution. Note that this is not a proper statistical test, but only a best effort to make the score easy to interpret for humans. Assuming a true outlier rate of φ, an object o with $\mathrm{COS}(o) = 1 - \varphi$ should have intuitively an outlier probability of 50%, as there are expected to be as many outliers as normal distributed objects with an equal or higher distance. For normalization we propose to use the formula:

$$\mathrm{norm}(p,\varphi) := \frac{\varphi \cdot (1-p)}{\varphi + p}$$

which rescales the score to $\mathrm{norm}(0,\varphi) = 1$, $\mathrm{norm}(1,\varphi) = 0$. As normalization constant we used $\varphi = 0.1\%$, but for large data sets it may be appropriate to lower this value to decrease sensitivity.

Definition 2 (Correlation Outlier Probability):
Let N denote a local set of reference objects. The *correlation outlier probability* of $o \in \mathcal{D}$ w.r.t. N and an assumed outlier rate φ, denoted by $\mathrm{COP}_N(o,\varphi)$, is defined as

$$\mathrm{COP}_N(o,\varphi) := \mathrm{norm}(1 - \mathrm{COS}(o), \varphi)$$

The role of φ is to make the scores more intuitive and usable.

As a final remark, note that we did not exclude $\delta = 0$. In this case, no correlation was detected and all dimensions are treated as noise dimensions. Our method then degenerates to a full-dimensional method, and (as we will see in the experimental section) is well able to handle this.

D. Choosing a Reference Set

So far, we have not yet discussed how to choose the reference set N w.r.t. which the correlation outlier probability $\mathrm{COP}(o)$ of an object $o \in \mathcal{D}$ is determined. As discussed above, we assume an unknown number of different generating mechanisms for the normal instances, thus, we argue to use a local rather than a global approach. In fact, we use the k-nearest neighbors (kNNs) of o for some input parameter k as a reference set, i.e. those k (or more in case of ties) objects in $\mathcal{D}$ having the smallest distance to o. Intuitively, k determines a threshold for the minimum number of points necessary to determine a *significant* mechanism. In addition, k should be large enough to span a δ dimensional hyperplane, e.g. $k > 3 \cdot d$ as suggested in [39]. On the other hand, k should not be chosen too high, because then it is likely that N contains points that are generated by different mechanisms themselves and PCA cannot detect a meaningful correlation hyperplane.

In order to obtain the principal components in a more stable and robust way (as we expect the presence of outliers), we experimented with two methods. The first is Robust PCA [39] (RPCA) using the suggested weight function $1 - erf$ to compute a weighted covariance matrix. Secondly, we applied a RANSAC [40] variation during computing the covariance matrix, with a threshold of $d_M^2 \leq \mathrm{quantile}_{\chi(d)}(0.9)$ to obtain a consensus covariance matrix. RPCA comes at little extra cost, as the weights are computed from the already known distances. RANSAC PCA scales much more badly with dimensionality, as it performs many iterations of PCA, which is $O(d^3)$ in complexity and contributes significantly to the overall complexity.

```
algorithm computeCOP
for each o ∈ 𝒟 do:
    compute N_k(o) the k-nearest neighbors of o;
    determine Σ_{N_k(o)}, V, Λ using robust PCA;
    for each δ ≤ d do:
        compute deviation from δ-dim hyperplane;
        estimate deviation distribution via Γ or χ²;
        compute COS_{N_k(o)}(o) according to Definition 1;
        if deviation > maximum deviation then:
            update maximum deviation score;
            store associated error vector;
        end if
    end for
    compute COP_{N_k(o)}(o) according to Definition 2;
end for
```

Figure 3. Computing the COP.

E. Explaining and Interpreting Outliers

Obviously, it would also be interesting for the user not only to retrieve outliers but also to obtain an interpretation and explanation *why* objects are considered outliers. As indicated above, we can utilize our modelling of correlation hyperplanes not only to derive outliers, but also to derive a quantitative model that explains the correlation w.r.t. which an outlier has been identified as an outlier. The *outlier explanation* generated by COP consists of two main components: the error vector which indicates the error estimated for the object, pointing to the expected position of the object, and the actual outlier score, which indicates how much more likely the outlier is to come from a different (i.e., outlier) mechanism as opposed to being just a rare object from the same mechanism that generated N.

F. Algorithm

With the concepts described in the previous subsections, we are now able to evaluate outliers by considering local correlations in the data and derive a model for each outlier o that explains why o is considered an outlier quantitatively by means of an equation system. An efficient algorithm for computing the outlier probability of all objects $o \in \mathcal{D}$ is given in Figure 3. The only input parameter is k, the number of nearest neighbors that are included into $N_k(o)$, which has already been discussed above. The algorithm computes the k-nearest neighbors $N_k(o)$ of each object o which requires $O(n^2)$ time using a sequential scan but can be supported by any well-established index structure reducing the runtime to $O(n \cdot \log n)$ on average. In addition, for each object o the local correlation is computed using PCA, which requires $O(k \cdot d^2 + d^3)$ time. The inner loop includes matrix multiplications, but is $O(d^3)$ overall. In general, $k \in O(d)$ and $k \ll n$, so the overall runtime complexity is $O(n^2 \cdot d^3)$ without index and $O(n \log n \cdot d^3)$ when using a spatial index for nearest neighbor search. With RANSAC-PCA, the number of iterations is significant and increases the runtime linearly by a factor of $O(i)$ where $i \gg d$.

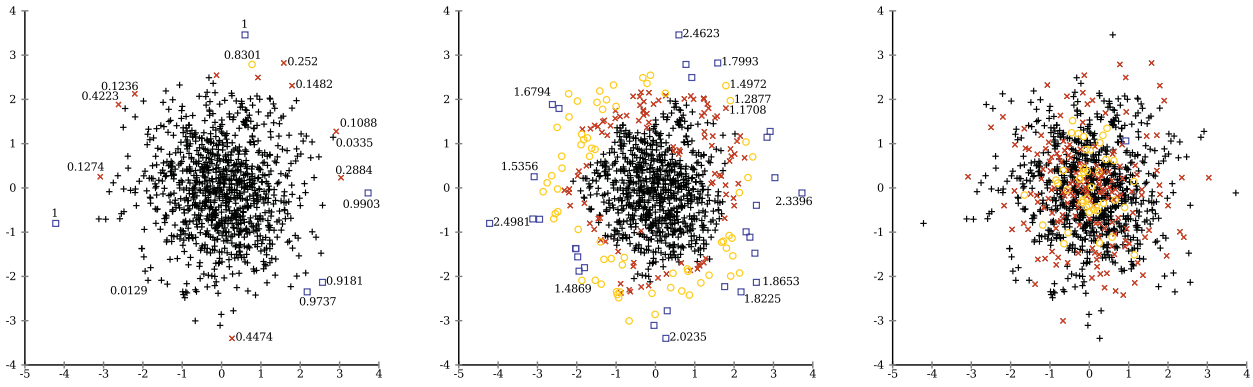
IV. Experiments

We compared COP to the local outlier factor (LOF) [13], the local correlation integral (LOCI) [16], and local outlier probabilities (LoOP) [18]. These methods consider full-dimensional densities around points and their neighbors. We have extended LOCI slightly to turn it into a ranking outlier detection scheme by using the k_σ value at which the point would turn into an outlier as rank. LOF and LOCI are probably among the most prominent and well-known outlier detection algorithms. All competitors have been implemented within in the unified framework ELKI [41], where also our new method COP is available.

A. Accuracy

As a baseline experiment, we generated 1000 objects that are standard normal distributed in 2 dimensions. There are no correlations in this data set and no true outliers, only rare objects from the normal distribution. Figure 4 visualizes the result for $k = 20$, with the colors assigned by the object score. As we can see, only a few objects score higher than 0.1. A few objects that deviate more than 3 standard deviations were given a high score over 0.5, while the remaining objects scored close to 0. Despite the method being designed to detect correlations, it works very well with this baseline approach. LOF is detecting similar outliers, but none reaches the suggested threshold of 3, and there is an almost linear progression with the outlier scores on the outside of the distribution. The reason is that LOF was designed with uniform density in mind. LOCI has big problems with this data set, due to the fluctuations inside the distribution. This can be seen as an issue of overfitting: for many objects it will find a radius where the object has a low local density compared to neighbor objects. We first expected an implementation error, but if you inspect the data closely, each of these objects can indeed be seen as a local outlier at the particular radius. The results of the approximate variant aLOCI [16] are slightly better, albeit it still detects only outliers in sparse areas at the center of the normal distribution. Similar to LOF, LOCI assumes that the inliers form a uniformly dense region, which does not hold for this data set. So even on this basic and uncorrelated test set, COP outperforms LOF and LOCI with respect to quality and usability of the scores.

As second toy example, we generated points along a sinus curve (with low variance normal distributed error) along with a few obvious outliers. Again we show the top scores for COP and LOF in Figure 5. As expected, LOF performs very well at detecting the obvious outliers, but fails to detect outliers close to the sinus curve. If the threshold is set too low, it starts detecting outliers inside the curve. The main motivation for this example is to see how well COP works in the presence of non-linear correlations. It handles this situation surprisingly well: Many non-linear correlations behave just like linear correlations in a small



(a) COP red $>$ 0.1, yellow $>$ 0.5, blue $>$ 0.9

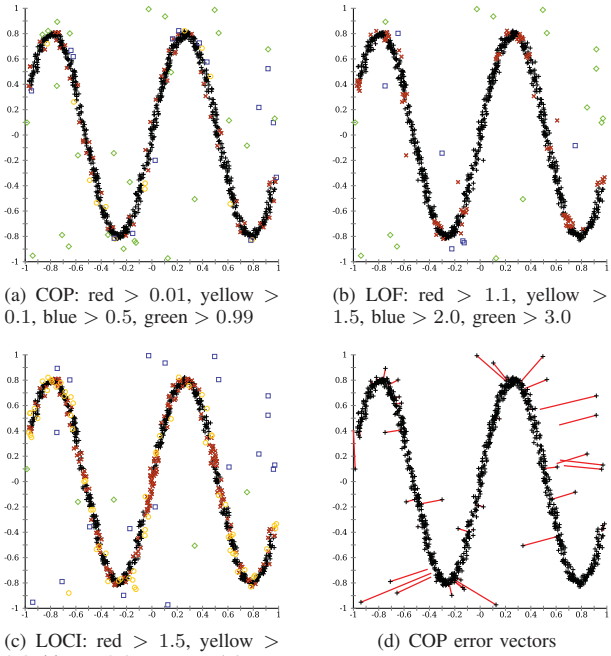
(b) LOF red $>$ 1.1, yellow $>$ 1.2, blue $>$ 1.5,

(c) LOCI red $>$ 1.5, yellow $>$ 2, blue $>$ 3,

Figure 4. Scores on a standard normal distribution

enough neighborhood and the error by modelling them with a linear approximation is not significant for outlier detection. The scores produced by COP are very reasonable – objects with a score larger than 0.5 (colored blue and green in Figure 5(a)) can safely be considered outliers. Objects with a score of $0.1-0.5$ may still be worth further investigation, and even objects with a score of just 0.01 are on the fringe of the actual data distribution. Similar to LOF, LOCI (Figure 5(c)) was able to detect the obvious outliers, and even some of the outliers nearby the sinus curve, but LOCI reacts much more sensitive to fluctuations within the cluster, similar to the observations in the previous example. Additionally, in Figure 5(d) we visualize the error vectors produced by COP, weighted with their outlier probability. Clearly, these vectors fit the intuitition of measuring the divergence from a local correlation as they point to a nearby location on the curve. Only for the outlier at approximately 0.9, 0.5 the vector is suboptimal. This object has 25% outliers in its 20 nearest neighbors, so the robust PCA did not perform that well.

However, COP also has its limitations. It uses a well defined outlier model, based on the deviation of a local trend. Classic density-based outliers will not necessarily have such a trend in their vincinity. Figure 6 visualizes such a situation. While COP does very well in recognizing outliers near the dominant correlation, the object marked as O (a typical density-based outlier) is not detected, because the 20 nearest neighbors (indicated with a green background) do not show a strong correlation, and as such the object is by definition not a correlation outlier. Other algorithms such as LOF might also have problems detecting some of these outliers, as the direct neighbors do also have a similar density. Only a method based on global density will be able to detect this outlier. This example shows the need to use multiple types of outlier detection models and methods: global density, local density and correlation outlier detection are *different kinds of outliers*, best detected by different approaches.



(a) COP: red $>$ 0.01, yellow $>$ 0.1, blue $>$ 0.5, green $>$ 0.99

(b) LOF: red $>$ 1.1, yellow $>$ 1.5, blue $>$ 2.0, green $>$ 3.0

(c) LOCI: red $>$ 1.5, yellow $>$ 2.0, blue $>$ 3.0, green $>$ 4.0

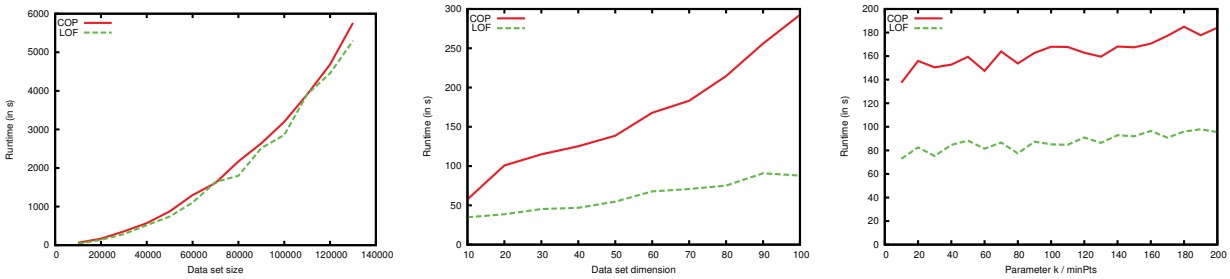
(d) COP error vectors

Figure 5. Scores on a sinus distribution with low noise and a few strong outliers

B. Scalability

We compared the scalability of COP with LOF, without index acceleration. We did not include LOCI in the benchmarks, since we were using the exact version for the other comparisons which is known to scale very poorly. A comparison with the approximate version aLOCI would have been possible. However, this would not have produced novel insights since aLOCI scales similar as LOF.

The runtime required to compute the COP value for



(a) Scalability in data set size n. (b) Scalability in dimensionality d. (c) Scalability in parameter k.

Figure 7. Scalability of COP compare to LOF.

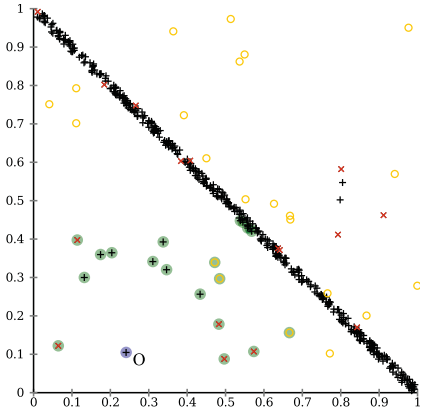


Figure 6. Strengths and limitations of COP: finds outliers close to the correlation, but does not recognize density outliers.
red > 0.1, yellow > 0.99. Green indicates the 20 nearest neighbors of O

all database objects w.r.t. increasing database size n in comparison to LOF is shown in Figure 7(a). Both approaches scale super-linear as expected. Let us note that we used materialized neighborhoods for LOF so that each neighborhood needs to be computed only once similar to COP. However, this implies a considerably higher storage overhead of LOF over COP. We also examined the scalability of our novel outlier detection approach compared to LOF w.r.t. the dimensionality d of the data points. The results shown in Figure 7(b) suggest a linear scalability for LOF and a super-linear growth for COP as expected. The impact of the parameter k ($minPts$ in LOF) on the runtime of the algorithm is not very importat, as both methods scale almost constant w.r.t. k, as it can be seen in Figure 7(c), supporting our analysis that the inner loop of COP is $O(k \cdot d^2 + d^3)$. As such, the methods are linear in k, but the iteration over the k neighbors is comparably cheap as opposed to the matrix inversion and neighbor search.

Unlike the pure scalability suggests, COP will not work with arbitrarily high-dimensional data, because PCA will not work very well with high-dimensional data. Not only does it scale with $O(d^3)$, but it also requires a significantly larger neighborhood with increasing dimensionality, which at some point will no longer be truly local. Furthermore, it will likely be affected by the curse of dimensionality. It is however very reasonable to combine COP with methods that produce subspace candidates to test, such as HiCS [27].

C. Results on Real-world Data

We used COP to find outliers in a data set containing 15 statistical measures for 413 former and current NBA players obtained from the NBA website[2]. We removed players with only a few games played, retaining 357 players.

Table I(a) depicts the top outliers according to COP, along with a trend indication of the error vector. The actual 15-dimensional error vectors are not as easy to interpret as the general trend indication. The data set is quite diverse with various types of outliers. Seven players scored the maximum score of 1.0 is this model, and a large number of records scored just slightly less. The key benefit here actually is to have this error model that indicates how a particular record is unusual. We will discuss the errors found by COP for some of the top players, because there are some interesting situations here.

Note that the error vectors point to the expected value. Eddy Curry for example, was unusually successfull at regular and 3 point throws and overall points, but not in free throws. Indeed in the data set he is given a 100% success rate for 3 point throws, but only 64.8% on free throws. Bruce Bowen played 661 games and 28.2 minutes on average, but scores only on 56.8% of free throws. Dennis Rodman is an exceptional rebounder, his blocks per game score is not low, unless compared to his number of rebounds. Antoine Carr is his very opposite, scoring low on rebounds and high on blocks and fouls. He played a lot and is a good shooter. Steve Kerr is an exceptional fair player and also extremely good at shooting. He played 910 games, but only 30 in the starting team. Steve Scheffler only played 5.3 minutes on average, despite having not bad scoring values. The most

[2]http://www.nba.com

Table I
RESULTS ON REAL-WORLD DATA

(a) Top outliers found by COP in NBA data set.

Player	COP	Error vector trend
Eddy Curry	1.0	-3PSuccess -Success -PointsPG FTSuccess -FoulsPG OffRebounds -DefRebounds TurnoverPG AssistPG StealsPG
Bruce Bowen	1.0	FTSuccess -StartTeam -MinutesPG TurnoverPG OffRebounds
Dennis Rodman	1.0	-OffRebounds -SumRebounds -DefRebounds FTSuccess BlocksPG PointsPG -Success -GamesPlayed
Antoine Carr	1.0	-GamesPlayed -FoulsPG -BlocksPG SumRebounds OffRebounds -FTSuccess DefRebounds -Success
Steve Kerr	1.0	FoulsPG -GamesPlayed -Success -3PSuccess TurnoverPG -FTSuccess SumRebounds OffRebounds DefRebounds
Steve Scheffler	1.0	MinutesPG -Success FoulsPG TurnoverPG SumRebounds DefRebounds PointsPG StealsPG BlocksPG
Danny Manning	1.0	GamesPlayed -FoulsPG -Success -StealsPG -TurnoverPG
Allen Iverson	0.999997	-TurnoverPG -StealsPG -PointsPG -MinutesPG -AssistPG FoulsPG -StartTeam
John Stockton	0.999989	-AssistPG -StartTeam -GamesPlayed -StealsPG DefRebounds -TurnoverPG SumRebounds -Success PointsPG BlocksPG
Andrei Kirilenko	0.999891	-BlocksPG -StealsPG FoulsPG GamesPlayed StartTeam -AssistPG
Avery Johnson	0.999789	3PSuccess -AssistPG -StartTeam -GamesPlayed FTSuccess FoulsPG -Success SumRebounds DefRebounds OffRebounds
Dirk Nowitzki	0.999678	-DefRebounds -SumRebounds -PointsPG -MinutesPG -FTSuccess OffRebounds -BlocksPG -3PSuccess
Charlie Bell	0.999010	TurnoverPG -MinutesPG -StealsPG
Jason Kidd	0.998700	-AssistPG -TurnoverPG -StealsPG -SumRebounds -OffRebounds 3PSuccess -DefRebounds Success -StartTeam FoulsPG -GamesPlayed
Shaquille O'Neal	0.995704	-BlocksPG -PointsPG -SumRebounds -OffRebounds FTSuccess -Success -DefRebounds -TurnoverPG -StartTeam 3PSuccess -MinutesPG -FoulsPG -GamesPlayed

(b) ALOI results.

Method	COP	LOF	LoOP	aLOCI
ROC AUC Score	0.82186	0.73131	0.77408	0.7112

interesting outlier however is Danny Manning. While all his individual statistics are well within the range of the data set, COP clearly indicated this player should have played more games. Upon closer investigation we noticed that while he is credited for playing 83 games total, he is also credited for being in the starting team in 398 games – obviously an error occurred during data entry of this data set.

We also applied LOF on that data set. Even when trying to optimize the parameter $minPts$, LOF could not find significant outliers. In all cases, the top outlier achieved a LOF value of below 1.8. This indicates that the objects in that data set exhibit a rather uniform density and outliers like the measurement error can only be detected when considering correlations as implemented by COP. The top outliers found by LOCI were players that had played next

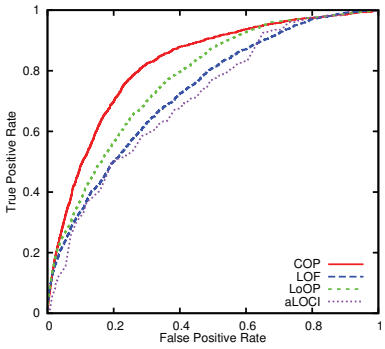


Figure 8. ROC curves for ALOI data set

to no games. The highest score achieved was 2.1, which would not have been considered an outlier by LOCI (the suggested significance value is 3). This difference to the results published in [16] is probably due to our NBA data set covering more seasons and players than their version.

As second real-world data set, we used the Amsterdam Library of Object Images (ALOI) [42], a collection of 110250 images from 1000 different objects in various light conditions. We downsampled some objects to obtain rare classes, so that 50000 images remain, 1508 of which are from rare objects. We extracted Haralick texture features [43] from the images, obtaining a 14 dimensional vector space. This is a rather challenging data set, and none of the methods evaluated performed excellent on this data set.

In Figure 8 we visualize the ROC curves for different algorithms: the new method COP along with the well-known local density-based outlier detection method LOF, the variation LoOP, and aLOCI. COP clearly offers an improved detection performance compare to all three methods. The area under curve (AUC) values are given in Table I(b).

V. CONCLUSIONS

We proposed a local model that takes correlations among varying subsets of attributes into account in order to find outliers in arbitrarily oriented subspaces. Our algorithm assigns to a point a score of being an outlier w.r.t. a set of reference points in the local neighborhood. An important contribution of COP is to not only produce an outlier score, but also to generate an explanation of how the outlier diverts from the norm. This significantly helps analysis of the outliers by a domain expert. The different abilities of COP compared to LOF and LOCI as prototypes of classical local outlier detection algorithms have been demonstrated on synthetic and real-world data. COP provides a new paradigm for outlier detection that exhibits rather different characteristics orthogonal to established methods. As a consequence, COP does not necessarily compete with existing outlier detection methods but rather complements them, and is best used in parallel with classic density-based outlier detection methods, as they detect different types of outliers.

REFERENCES

[1] D. Hawkins, *Identification of Outliers*. Chapman and Hall, 1980.

[2] A. Zimek, "Correlation clustering," Ph.D. dissertation, LMU, Germany, 2008.

[3] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Stat. Anal. Data Min.*, vol. 5, no. 5, pp. 363–387, 2012.

[4] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel, "On evaluation of outlier rankings and outlier scores," in *Proc. SDM*, 2012, pp. 1047–1058.

[5] V. Barnett and T. Lewis, *Outliers in Statistical Data*, 3rd ed. John Wiley&Sons, 1994.

[6] E. M. Knorr and R. T. Ng, "Algorithms for mining distance-based outliers in large datasets," in *Proc. VLDB*, 1998.

[7] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proc. SIGMOD*, 2000, pp. 427–438.

[8] S. D. Bay and M. Schwabacher, "Mining distance-based outliers in near linear time with randomization and a simple pruning rule," in *Proc. KDD*, 2003, pp. 29–38.

[9] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchthold, "Efficient biased sampling for approximate clustering and outlier detection in large datasets," *IEEE TKDE*, vol. 15, no. 5, pp. 1170–1187, 2003.

[10] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," in *Proc. PKDD*, 2002, pp. 15–26.

[11] Y. Pei, O. Zaïane, and Y. Gao, "An efficient reference-based approach to outlier detection in large datasets," in *Proc. ICDM*, 2006, pp. 478–487.

[12] H.-P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proc. KDD*, 2008, pp. 444–452.

[13] M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. SIGMOD*, 2000, pp. 93–104.

[14] W. Jin, A. Tung, and J. Han, "Mining top-n local outliers in large databases," in *Proc. KDD*, 2001, pp. 293–298.

[15] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Proc. PAKDD*, 2002, pp. 535–548.

[16] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," in *Proc. ICDE*, 2003, pp. 315–326.

[17] W. Jin, A. K. H. Tung, J. Han, and W. Wang, "Ranking outliers using symmetric neighborhood relationship," in *Proc. PAKDD*, 2006, pp. 577–593.

[18] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: local outlier probabilities," in *Proc. CIKM*, 2009.

[19] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," in *Proc. SIGMOD*, 2001, pp. 37–46.

[20] J. Zhang, M. Lou, T. W. Ling, and H. Wang, "HOS-miner: A system for detecting outlying subspaces of high-dimensional data," in *Proc. VLDB*, 2004, pp. 1265–1268.

[21] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Outlier detection in axis-parallel subspaces of high dimensional data," in *Proc. PAKDD*, 2009, pp. 831–838.

[22] E. Müller, I. Assent, U. Steinhausen, and T. Seidl, "OutRank: ranking outliers in high dimensional data," in *Proc. ICDE Workshop DBRank*, 2008, pp. 600–603.

[23] E. Müller, M. Schiffer, P. Gerwert, M. Hannen, T. Jansen, and T. Seidl, "SOREX: Subspace outlier ranking exploration toolkit," in *Proc. ECML PKDD*, 2010, pp. 607–610.

[24] E. Müller, M. Schiffer, and T. Seidl, "Adaptive outlierness for subspace outlier ranking," in *Proc. CIKM*, 2010.

[25] ——, "Statistical selection of relevant subspace projections for outlier ranking," in *Proc. ICDE*, 2011, pp. 434–445.

[26] H. V. Nguyen, V. Gopalkrishnan, and I. Assent, "An unbiased distance-based outlier detection approach for high-dimensional data," in *Proc. DASFAA*, 2011, pp. 138–152.

[27] F. Keller, E. Müller, and K. Böhm, "HiCS: high contrast subspaces for density-based outlier ranking," in *Proc. ICDE*, 2012.

[28] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2011.

[29] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM TKDD*, vol. 3, no. 1, pp. 1–58, 2009.

[30] K. Sim, V. Gopalkrishnan, A. Zimek, and G. Cong, "A survey on enhanced subspace clustering," *Data Min. Knowl. Disc.*, 2012.

[31] H.-P. Kriegel, P. Kröger, and A. Zimek, "Subspace clustering," *WIREs DMKD*, vol. 2, no. 4, pp. 351–364, 2012.

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

[33] E. Achtert, C. Böhm, J. David, P. Kröger, and A. Zimek, "Global correlation clustering based on the Hough transform," *Stat. Anal. Data Min.*, vol. 1, no. 3, pp. 111–127, 2008.

[34] S. Günnemann, I. Färber, K. Virochsiri, and T. Seidl, "Subspace correlation clustering: finding locally correlated dimensions in subspace projections of the data," in *Proc. KDD*, 2012, pp. 352–360.

[35] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, and A. Zimek, "Deriving quantitative models for correlation clusters," in *Proc. KDD*, 2006, pp. 4–13.

[36] ——, "On exploring complex relationships of correlation clusters," in *Proc. SSDBM*, 2007, pp. 7–16.

[37] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Interpreting and unifying outlier scores," in *Proc. SDM*, 2011.

[38] S. C. Choi and R. Wette, "Maximum likelihood estimation of the parameters of the gamma distribution and their bias," *Technometrics*, pp. 683–690, 1969.

[39] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "A general framework for increasing the robustness of PCA-based correlation clustering algorithms," in *Proc. SSDBM*, 2008, pp. 418–435.

[40] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[41] E. Achtert, S. Goldhofer, H.-P. Kriegel, E. Schubert, and A. Zimek, "Evaluation of clusterings – metrics and visual support," in *Proc. ICDE*, 2012, pp. 1285–1288.

[42] J. M. Geusebroek, G. J. Burghouts, and A. Smeulders, "The Amsterdam Library of Object Images," *Int. J. Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.

[43] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE TSAP*, vol. 3, no. 6, pp. 610–623, 1973.