HiCS: High Contrast Subspaces for Density-Based Outlier Ranking

Fabian Keller, Emmanuel Müller, Klemens Böhm

Institute for Program Structures and Data Organization
Karlsruhe Institute of Technology (KIT), Germany
{fabian.keller, emmanuel.mueller, klemens.boehm}@kit.edu

***Abstract*—Outlier mining is a major task in data analysis. Outliers are objects that highly deviate from regular objects in their local neighborhood. Density-based outlier ranking methods score each object based on its degree of deviation. In many applications, these ranking methods degenerate to random listings due to low contrast between outliers and regular objects. Outliers do not show up in the scattered full space, they are hidden in multiple high contrast subspace projections of the data. Measuring the contrast of such subspaces for outlier rankings is an open research challenge.**

In this work, we propose a novel subspace search method that selects high contrast subspaces for density-based outlier ranking. It is designed as pre-processing step to outlier ranking algorithms. It searches for high contrast subspaces with a significant amount of conditional dependence among the subspace dimensions. With our approach, we propose a first measure for the contrast of subspaces. Thus, we enhance the quality of traditional outlier rankings by computing outlier scores in high contrast projections only. The evaluation on real and synthetic data shows that our approach outperforms traditional dimensionality reduction techniques, naive random projections as well as state-of-the-art subspace search techniques and provides enhanced quality for outlier ranking.

I. Introduction

Outlier mining is an important task in the field of knowledge discovery. In applications such as fraud detection, gene-expression analysis or environmental surveillance, one is interested in rare, suspicious, and unexpected objects. Outlier analysis searches for such highly deviating objects in contrast to regular objects. An outlier has highly deviating attribute values compared to its local neighborhood. For example, in environmental surveillance (cf. Fig. 1) a sensor node might be an outlier as it shows an abnormally high deviation w.r.t. *air pollution index* and *noise level*. For instance, $outlier_1$ shows a high deviation in this specific subset of attributes only. Another sensor node ($outlier_2$) shows high deviation w.r.t. *humidity* and *temperature*, independent of its *air pollution index* and its *noise level*. Thus, a sensor node might be an outlier in one of these attribute combinations and a regular object in all other attributes. In general, these multiple roles (outlying vs. regular behavior) of objects can be observed in other domains as well: Suspicious customers show fraud activity only w.r.t. some financial transactions, and genes show unexpected expression only under specific medical conditions.

Traditional outlier mining [26], [16], [5], [13], [7], [25] is unable to detect such outliers hidden in subsets of all given attributes. Most outlier mining techniques search for outliers w.r.t. all given attributes. Considering object distances in the full data space, these methods fall prey to randomly distributed attribute combinations. In our example, *humidity* and *noise level* in combination show no clear outlier objects and hinder outlier detection. Furthermore, due to the increasing number of attributes in today's databases, distances between objects grow more and more alike [6]. Outlier ranking techniques score each object based on the degree of deviation, e.g., by computing its density in the full data space [7]. Thus, for high dimensional data, outlier rankings degenerate to random listings, as outliers do not show up in the full space. Other common statistical techniques try to detect outliers in single attributes [26]. However, by ignoring the dependencies between several attributes, these techniques miss outliers that appear only due to correlations in multi-dimensional spaces. We focus on such outliers, which are neither visible in the full space nor in a single attribute.

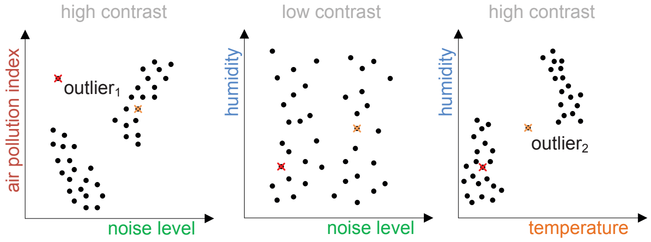


Fig. 1. Environmental surveillance example: suspicious sensor readings

Subspace mining has been proposed as a novel data mining paradigm to tackle this challenge. It detects highly deviating objects in any possible attribute combination (low dimensional projection). While dimensionality reduction techniques aim at such lower dimensional projections, they are not designed as pre-processing step for outlier ranking. General measures, such as the variance of the data in PCA [14], are not appropriate objective functions for outlier ranking. Novel quality criteria and processing schemes are required for subspace outlier mining. In particular, we search for high contrast subspaces. Such subspaces have the defining characteristic that outliers can be clearly distinguished from regular objects within the subspace context. Our general aim is a two-step processing:

(1) **Subspace search:** measuring the contrast of subspaces
(2) **Outlier ranking:** score objects in high contrast subspaces

We consider the decoupling of these two steps to be an open research issue. Current subspace outlier mining techniques [1], [18], [23], [21] focus on interleaved algorithms only, which select subspaces during outlier mining. We propose to consider subspace outlier mining as a decoupled process, divided into "subspace search" and "outlier ranking". By treating these two steps as independent problems, one can design and combine the respective algorithms in a modular fashion. It also allows both research fields to evolve independently. In conclusion, any improvement in either of these steps will lead to an improvement in the overall outlier detection quality. Thus, future research in outlier mining may benefit from the proposed decoupling.

In this work, we focus on the first step and propose a novel subspace search method that selects high contrast subspaces for density-based outlier ranking. As outlier score for the ranking we rely on the commonly used local outlier factor (*LOF*) [7]. However, any other outlier score could be used as instantiation of the second step. Our subspace search technique is based on a novel selection of high contrast subspaces (HiCS). It provides three main contributions:

- The decoupling of *subspace search* as generalized preprocessing step for outlier ranking
- A *contrast measure* based on the conditional dependence of dimensions in the selected subspaces
- Two *statistical instantiations* of our contrast measure ensuring a robust parametrization of our technique

Our contrast measure is based on statistical tests and enables a high quality outlier ranking of outliers hidden in arbitrary subspace projections. Our approach searches for high contrast subspaces with a significant amount of conditional dependence among the selected dimensions. Thus, we enhance the quality of traditional outlier rankings by computing outlier scores in high contrast projections only. The evaluation on real and synthetic data shows that our approach outperforms traditional dimensionality reduction techniques [14], naive random projections [20] as well as state-of-the-art subspace search techniques [8], [15] and provides enhanced quality for outlier rankings.

II. Related Work

In this section, we review existing techniques in the areas of outlier discovery and subspace mining. In particular, we explain the differences of existing paradigms compared to our novel subspace search approach.

a) Traditional Outlier Ranking: There have been different outlier detection paradigms proposed in the literature, ranging from deviation-based methods [26], distance-based methods [16], [5], [13] to density-based methods [7], [25]. We focus on the density-based outlier ranking paradigm, which computes a score for each object by measuring its degree of deviation w.r.t. a local neighborhood. Thus, one is able to detect local density variations between low density outliers and their high density (clustered) neighborhood. However, all of those traditional outlier mining approaches have one drawback. They cannot detect outliers in subspaces, as their degree of deviation considers only the full data space.

b) Subspace Outlier Ranking: Outlier detection in subspaces has first been proposed by [1]. Recent approaches have enhanced subspace outlier mining by ranking objects based on any possible subspace projection [11], [20], [18], [23], [21]. These techniques differ in their choice of subspaces. The majority of approaches uses specialized heuristics for subspace selection that are integrated into the outlier ranking [11], [18], [23], [21]. In general, all of these techniques use an integrated processing of subspaces and outliers. This implies that scoring functions and subspace selection are tightly coupled such that none of these techniques would benefit from a novel scoring function or a novel subspace selection technique.

The only approach with a decoupled processing is considered as a baseline for our technique. It selects several subspace projections randomly [20]. Obviously, this random selection does not guarantee high quality results. Selection of arbitrary projections will result in random rankings just as in the full data space. With our work we aim at a decoupled processing with two steps as proposed in [20]. In contrast to a naive random selection of subspaces, we aim at an enhanced contrast measure based on sound statistical foundations.

c) Subspace Search: Based on the general idea of subspace mining in arbitrary projections of the data, several preprocessing techniques for the selection of subspaces have been proposed [8], [15], [24], [4]. All of these techniques focus on the related domain of subspace clustering. They try to decouple the detection of clusters and the selection of individual subspaces for each cluster. However, each of the four subspace search models depends on a specific cluster definition.

First, the *Enclus* approach proposes a selection based on the entropy measure [8]. Its quality measure for subspaces highly depends on the subspace clustering algorithm *CLIQUE* [2]. It partitions the data space in equally sized grid cells. A subspace is selected if it has low entropy, i.e., if it shows a large variation in the densities of the grid cells. With our approach we follow this basic idea of contrast, however, we do not rely on fixed grid cells. This is because they induce several drawbacks for density estimation in high dimensional spaces.

Other techniques, i.e., RIS [15] and SURFING [4], have been proposed for the detection of density-based subspace clusters based on the DBSCAN paradigm [10]. For instance, RIS counts the core objects in a subspace projection and uses them as a measure for its subspace selection criterion. Recently, a subspace search method has been proposed for spectral clustering as well [24].

In general, all of the proposed subspace search methods focus on specific clustering tasks. Their selection highly depends on the underlying clustering model. In contrast to this, our technique is based on a more general analysis of conditional dependence. Furthermore, we propose an instantiation of our objective function that aims at high contrast w.r.t. density-based outlier ranking, and thus, is tailored to detect low density regions as required for many outlier models.

III. High Contrast Subspaces (HiCS)

The main idea of our HiCS approach is the statistical selection of high contrast subspaces. We propose a processing based on a series of statistical tests. Each test compares the data distribution in a local subspace region to its marginal distribution. Dependencies between attributes highlight the high contrast of a subspace. Based on these statistical tests and the detected dependence between attributes we derive our contrast measure. It provides the means for high quality outlier ranking in a selection of high contrast subspaces.

Overall, HiCS establishes a first statistical subspace search technique for density-based outlier ranking. In the following, we will introduce the necessary notation in Section III-A, and define the general objective for our high contrast subspaces in Section III-B. We will introduce the notion of subspace slices that specify local subspace regions in Section III-C, and define the contrast measure in Section III-D. In Section III-E we will show how different statistical tests can be used to instantiate our contrast definition.

A. Notation

Let DB be a database containing N objects, each described by a D-dimensional real-valued data vector $\vec{x} = (x_1, \ldots, x_D)$. The set $\mathcal{A} = \{1, \ldots, D\}$ denotes the full data space of all given attributes. Any attribute subset $S = \{s_1, \ldots, s_d\} \subseteq \mathcal{A}$ will be called a d-dimensional subspace projection. We denote the distance between objects $\vec{x}$ and $\vec{y}$ as $dist_{\mathcal{A}}(\vec{x}, \vec{y})$, which can be instantiated for instance by the widely used Euclidean Distance $dist_{\mathcal{A}}(\vec{x}, \vec{y}) = \sqrt{\sum_{s \in \mathcal{A}} (x_s - y_s)^2}$.

As general property of any outlier ranking method we have to consider the underlying scoring function. It measures the outlierness of an object. Traditionally, each object is sorted according to a single outlier score $score(\vec{x})$ measuring the degree of deviation in all given attributes $\mathcal{A}$. Traditional density-based outlier scores measure the density $p(\vec{x})$ of an object and compare it to the density in the local neighborhood of $\vec{x}$. Local outlier ranking based on density deviation in local neighborhoods has first been proposed by LOF [7]. In recent years, this outlier mining paradigm has been extended by enhanced scoring functions and efficient outlier ranking algorithms [25], [5], [13], [19], [17], [23], [9].

The problem with all of these full space approaches is introduced by the curse of dimensionality. As pointed out in [6], the definition of a local neighborhood becomes meaningless for a large number of attributes. Furthermore distances between objects grow more and more alike, thus

$$\lim_{|\mathcal{A}| \to \infty} \max_{\vec{z} \in DB} dist_{\mathcal{A}}(\vec{z}, \vec{x}) \; - \min_{\vec{z} \in DB} dist_{\mathcal{A}}(\vec{z}, \vec{x}) = 0$$

Since local outlier ranking calculates the density based on the object distances, we observe the same effect for the minimal and maximal value of $score(\vec{x})$. As a result, all mentioned outlier score functions will suffer from a loss of contrast, i.e.:

$$score(\vec{x}) \approx score(\vec{y}) \quad \forall \; \vec{x}, \vec{y} \in DB$$

Any outlier ranking obtained for a sufficiently high dimensional database will degenerate into a random ranking with very similar scores for all objects.

Subspace outlier rankings address this problem by evaluating the score function in lower dimensional subspace projections. They simply restrict the distance computation to a selected subspace S, i.e., compute $dist_S$. Thus, any outlier ranking with $score(\vec{x})$ can be extended to a subspace score $score_S(\vec{x})$. The idea is to aggregate these $score_S(\vec{x})$ values over several subspaces. Each score provides some insights about the deviation of $\vec{x}$ in a lower dimensional projection S. The final ranking is derived from the aggregation of these scores:

Definition 1: ***Outlier Score***

$$score(\vec{x}) = \frac{1}{|RS|} \sum_{S \in RS} score_S(\vec{x})$$

In the most basic approach [20], RS is a selection of random subspaces that contribute to the overall ranking. A major drawback of this approach is that irrelevant subspaces in RS might blur the overall order of objects. To tackle this challenge, we propose a novel method to select high contrast subspaces only. Our subspace search technique excludes low contrast subspaces, which inhibit a clear distinction between outliers and regular objects.

For our experiments, we instantiate $score_S(\vec{x})$ with the commonly used local outlier factor [7]. It has been used for the subspace extension based on random projections [20] as well. However, our technique is not restricted to LOF only. Any other density-based scoring function could be used for $score_S(\vec{x})$. This flexibility w.r.t. the score function is a main advantage of our method. We only consider the contrast of subspaces and their selection as pre-processing step. Any improvement in the area of outlier scoring can be applied directly to our approach as well. In recent years several extensions of LOF have addressed specific challenges for this local outlier ranking [25], [19], [23], [17]. While each of these publications proposes an individual score function, they all have an assumption in common: **An outlier has low density compared to its local neighborhood.** Our technique relies only on this general assumption.

To derive our criterion for subspace contrast, we treat the attributes in DB as random variables. We use the notion of probability density functions (pdf) to derive the formal background of our contrast criterion. We will adapt the notation for subspaces as follows. For a given subspace $S = \{s_1, \ldots, s_d\}$, we refer to the projected data vectors as $\vec{x}_S = (x_{s_1}, \ldots, x_{s_d})$.

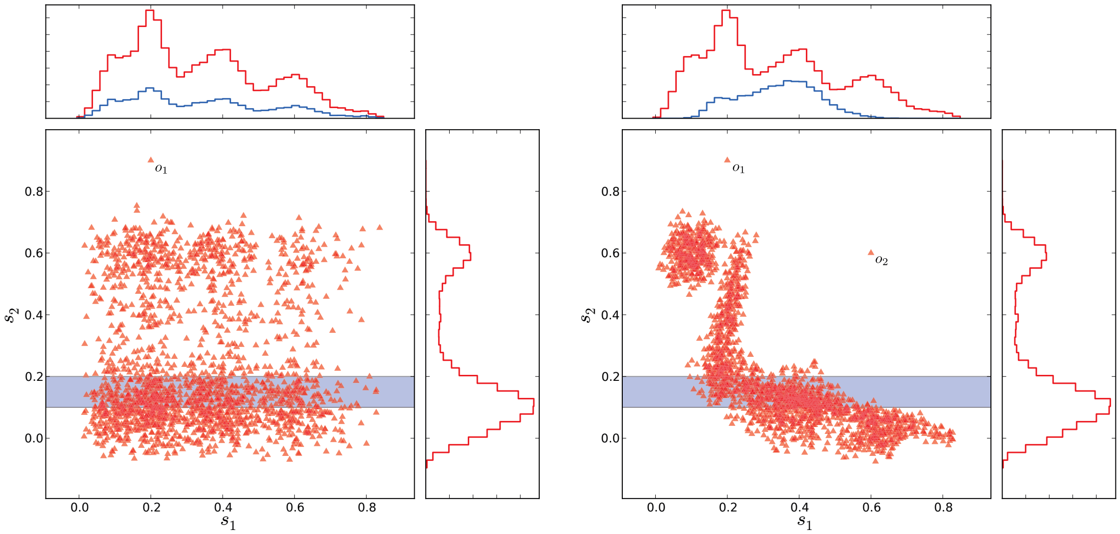
*Notation 1: The subspace data vector $\vec{x}_S$ is distributed by an unknown **joint pdf** of S:*

$$p_{s_1,\ldots,s_d}(x_{s_1}, \ldots, x_{s_d})$$

By integration over all attributes $s \in \mathcal{A} \setminus s_i$ we obtain:

*Notation 2: The **marginal pdf** of attribute s_i:*

$$p_{s_i}(x_{s_i})$$



(a) Dataset A – example of an uncorrelated joint pdf

(b) Dataset B – example of a correlated joint pdf

Fig. 2. high vs. low contrast and the effects on outlier ranking

Please note that the marginal densities are simply one-dimensional projections, independent from any subspace. Furthermore, we can require a condition on the attributes $s \in S \setminus s_i$, which leads to the following notion.

Notation 3: *The* ***conditional pdf*** *of attribute* s_i*:*

$$p_{s_i \,|\, s \in S \setminus s_i}\left(x_{s_i} \,\middle|\, \{x_s \;:\; s \in S \setminus s_i\}\right)$$

Thus, we express the probability density function of s_i w.r.t. $|S| - 1$ conditions on all other attributes in the subspace.

B. High Contrast Improves Outlier Ranking

Given the notion of probability density in any subspace S, we measure the contrast by comparing conditional probability densities to the corresponding marginal densities for all attributes $s_i \in S$. This idea is based on the following key hypothesis: the detection of *non-trivial outliers* is only possible in a subspace S that shows high dependence between all attributes $s_i \in S$. The notion of *non-trivial* outliers is a new concept and we will postpone the formal definition for a moment. Intuitively, a non-trivial outlier is an outlier in subspace S, but it is not visible as outlier in any one-dimensional projection of S, i.e., all its one-dimensional attribute values are located in regions of high density. Based on the one-dimensional projections, a non-trivial outlier appears to be a clustered object.

1) ***Motivation Example****:* We illustrate the relationship between correlated subspaces and non-trivial outliers by a toy example (cf. Figure 2). It consists of two two-dimensional datasets. Both datasets were generated from the same marginal distributions. In dataset A, s_1 and s_2 are completely uncorrelated. As a result, this two-dimensional subspace is filled by a random scattering of objects in consistency with the marginal distribution. Nevertheless the dataset contains an outlier object o_1. By considering the one-dimensional projections of this subspace, the existence of o_1 is not a surprise: o_1 could trivially be detected by the examination of the one-dimensional distribution of attribute s_2. We call such an object a *trivial outlier*. In summary, the evaluation of the two-dimensional subspace does not reveal any new information for this dataset.

The other dataset features marginal distributions identical to the ones of dataset A. The difference is that dataset B shows a significant correlation. The correlation allows the data objects to form regions of varying or unexpected densities over the total possible area that would be consistent with the marginal distribution. We observe (a) cluster-like dense agglomerations of objects and (b) sparse or even empty regions. Besides a trivial outlier o_1, the subspace also features an other outlier o_2. This time the outlier is hidden in all one-dimensional subspace projections, where it even appears to be a clustered object. We will call this type of objects non-trivial outliers. For dataset B the evaluation of the two-dimensional subspace was worthwhile and reveals significant insight regarding the data structure. Accordingly, we have found an example for a high contrast subspace in this case.

Once we have found such a high contrast subspace we can apply any density-based outlier ranking algorithm: for instance in dataset B, o_1 and o_2 both exhibit a much lower density compared to the local neighborhood. Thus, determining the outlierness in the two-dimensional subspace of dataset B would result in a detection of o_1 and o_2, i.e., $score_S(o_{1/2}) \gg score_S(o_i)$ for all other objects o_i in the database.

We can also explain the essential idea of our approach to identify high contrast subspaces using this toy example. Depicted on top of each plot in Figure 2, we show two different histograms for the s_1 axis of both datasets. The first one (red) represents the full data sample, i.e., corresponds to the marginal probability distribution $p_{s_1}(x_{s_1})$. The blue one shows the conditional probability distribution that is generated by the sample according to the selection range w.r.t. the s_2 axis (blue area). The comparison of the blue vs. the red histograms for both datasets show a basic property of correlation: Whereas the histograms for dataset A are in good agreement, we see a significant discrepancy between the two histograms for the high contrast subspace B. The proposed HiCS algorithm is based on the evaluation of this discrepancy.

Please note that we design our contrast measure as a conservative subspace selection criterion. The set of selected subspaces is a proper superset of the subspaces containing non-trivial outliers. We will later show that high contrast is a necessary condition for non-trivial outliers. Still, the result may contain subspaces without any outliers.

In the following we will focus on non-trivial outliers only. The reason is simple: A user might already know about the existence of one-dimensional outliers; one can detect these outliers by existing methods [26] without difficulty. Moreover, our subspace search can detect trivial outliers as a by-product of the search for non-trivial outliers. For instance in dataset B, we will always detect o_1 as outlier as soon as attribute s_2 is part of any high contrast subspace. In any case, the detection of non-trivial outliers will provide a much higher information gain to the user. Therefore, we focus on the detection of correlated subspaces containing such non-trivial outliers.

*2) **Contrast based on correlation of dimensions**:*
In probability theory, two events A and B are called independent and uncorrelated, if and only if the probability of the combined event is given by the product of the individual probabilities, i.e.:

$$p(A \cap B) = p(A) \cdot p(B) \tag{1}$$

By putting the notion of correlation in the context of subspaces, we obtain:

*Definition 2: A subspace S is called an **uncorrelated subspace** if and only if:*

$$p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d}) = \prod_{i=1}^{d} p_{s_i}(x_{s_i}) \tag{2}$$

Please note that the formal distinction between statistical dependence and correlation is not important for our purpose. Strictly speaking, the term *set of independent attributes* would be the appropriate expression. Instead we prefer to use the more concise term *uncorrelated subspace* to express the statistical independence within a subspace.

To support the observations regarding Figure 2, we want to examine the characteristics of outlier mining in uncorrelated subspaces more formally. The observation of a high value of $score_S(\vec{x})$ implies that the object $\vec{x}$ is located in a region with a low value of the joint pdf $p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d})$. On the other hand, we can evaluate the expected density for $\vec{x}$ under the assumption of an uncorrelated subspace:

$$p_{expected}(x_{s_1},\ldots,x_{s_d}) \equiv \prod_{i=1}^{d} p_{s_i}(x_{s_i}) \tag{3}$$

We define the notion of trivial outliers over the comparison of the expected density with the joint density:

*Definition 3: We call an object $\vec{x}_S$ a **non-trivial outlier** w.r.t. subspace S if*

$$p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d}) \ll p_{expected}(x_{s_1},\ldots,x_{s_d}) \tag{4}$$

Comparing the definition of an uncorrelated subspace (Eq. 2) with the definition of non-trivial outliers leads to:

Theorem 1: An uncorrelated subspace S does not contain any non-trivial outlier.
For an uncorrelated subspace, the joint probability density function $p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d})$ is by definition equal to the product of the marginal pdfs and thus, will never fulfill Eq. 4. On the other hand, a correlated subspace allows significantly smaller values of $p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d})$ compared to the expected density. Thus, we define subspace correlation as objective function for the subspace contrast.

*3) **Measuring Correlation**:*
We propose to quantify the subspace contrast by a comparison of different probability density functions. To simplify the notation, we will express all following conditional probability densities only for s_1 without loss of generality. In the case of an uncorrelated subspace, Eq. 2 simplifies the definition of all conditional probability densities within the subspace, i.e.:

$$\begin{aligned} p_{s_1}(x_{s_1}|x_{s_2},\ldots,x_{s_d}) &= \frac{p_{s_1,\ldots,s_d}(x_{s_1},\ldots,x_{s_d})}{p_{s_2,\ldots,s_d}(x_{s_2},\ldots,x_{s_d})} \\ &= p_{s_1}(x_{s_1}) \end{aligned} \tag{5}$$

This allows to measure the contrast of a subspace by determining the degree of violation of Eq. 5. In other words, we have to compare a conditional pdf of s_1 to the corresponding marginal pdf, and we assign a high contrast to a subspace if we observe a significant deviation between the two pdfs. Please note that the correlation analysis within subspaces goes beyond classical correlation analysis approaches, since we may be faced with high contrast subspaces with more than two dimensions. In contrast to, say, the Pearson or Spearman correlation coefficient [28], the proposed approach is not limited in the subspace dimensionality. Furthermore, it is possible to detect any kind of non-linear correlation. Above all, our approach does not require an evaluation of a high dimensional joint pdf, but is based on one-dimensional densities only. Hence, it does not fall prey to the curse of dimensionality.

In the following sections we will discuss (1) how to empirically analyze the the conditional pdf by introducing the notion of *subspace slices*, (2) how to compare the conditional pdf to the marginal pdf by means of statistical tests, and (3) how to instantiate these statistical tests in our contrast measure.

C. Evaluation of conditional densities

The main challenge for the proposed calculation of the subspace contrast is the empirical analysis of the conditional probability densities $p_{s1|\ldots} \equiv p_{s_1|s_2,\ldots,s_d}(x_{s_1}|x_{s_2},\ldots,x_{s_d})$. Since we do not require any knowledge of the underlying density functions, our goal is to obtain a sample of $p_{s1|\ldots}$ for a specific set of conditions.

Definition 4: A set of $|S|-1$ lower and upper conditions $[l_i, r_i]$ is called a ***subspace slice*** *w.r.t. subspace S:*

$$C = \{x_{s_2} \in [l_2, r_2], \ldots, x_{s_d} \in [l_d, r_d]\} \tag{6}$$

The selection of objects that satisfy a subspace slice condition leads to a subsample of DB with a sample size N'. The advantage of these subspace slices over any grid-based density estimation is that we can construct the subspace slices in a way that does not suffer from the curse of dimensionality. The goal is to choose the intervals in the subspace slice C in such a way that the expectation value for the selection sample size N' is fixed. We derive the construction of the intervals as follows: Each condition in C can be associated with a certain selection of objects. Starting with the full sample of $|DB|$ objects, each selection removes a certain fraction of objects from the current sample. We denote the fraction of objects that will remain in the sample by $\alpha_1 \in (0,1)$. The suffix emphasizes that α_1 is the selection probability for a single condition. By assuming an uncorrelated subspace, the selections are independent from each other. In this case the probability for a single object to be selected after $|C|$ equally sized selection steps is $\alpha_1^{|C|}$. Thus, the expectation value of the remaining sample size N' after $|C|$ selections is given by:

$$E[N'] \ = \ N \cdot \alpha_1^{|C|} \tag{7}$$

We can utilize this step-wise selection in the algorithm to generate subspace slices that automatically adapt the selection intervals $[l_i, r_i]$ to provide a desired target statistic size N', independent of the dimensionality of the subspace. The implementation details are given in Section IV-A.

D. Quality criterion for the subspace contrast

As mentioned before, our subspace contrast definition is based on the degree of violation of Eq. 5. Since we do not require density functions explicitly given, we introduce the following notation to emphasize that we refer to estimated density distributions from a data sample:

- $\hat{p}_s$ refers to the marginal density of some attribute $s \in S$ w.r.t. the full dataset.
- $\hat{p}_{s|C}$ refers to the density of x_s w.r.t. the remaining dataset that fulfills a certain condition set C.

We are now looking for a function $deviation\left(\hat{p}_s,\ \hat{p}_{s|C}\right)$ that compares $\hat{p}_s$ to $\hat{p}_{s|C}$, measures the discrepancy between the two distributions and outputs a value that is proportional to the deviation. There are many ways to define such a function. With HiCS we focus on two different statistical tests, namely Welch's t-test and the Kolmogorov-Smirnov test, which will be described in Section III-E. We will call the two resulting variants HiCS$_{\text{WT}}$ and HiCS$_{\text{KS}}$.

In terms of statistical testing, we define the null hypothesis as: *Both samples originate from the same underlying pdf.* In other words, the null hypothesis states that the differences between $\hat{p}_s$ and $\hat{p}_{s|C}$ are within the limits of statistical fluctuations. Due to these fluctuations, the significance of a single statistical test is very limited. In order to achieve a high statistical precision, the HiCS algorithm performs a large number M of different tests. Thus, the definition of our quality criterion of the subspace contrast is given by:

Definition 5: ***Subspace contrast***

$$contrast(S) \ \equiv \frac{1}{M} \sum_{i}^{M} deviation\left(\hat{p}_{s_i},\ \hat{p}_{s_i|C_i}\right) \tag{8}$$

HiCS computes the subspace contrast with a Monte Carlo approach. The algorithm performs M iterations. For each iteration, we randomly pick an attribute $s_i \in S$ and generate a random subspace slice C_i. The respective samples are passed to the $deviation$ function, i.e., a function that performs the statistical test. We calculate the final result of the subspace contrast by averaging the deviations of all M statistical tests.

E. Statistical tests

Regarding the implementation of the $deviation(\hat{p}_A, \hat{p}_B)$ function, we have employed and examined two different statistical tests.

The first approach uses Welch's t-test, which is a variation of a Student's t-test. The idea of this solution is to first extract estimations of statistical moments from both samples, and then perform a comparison based on these characteristics. The difference between Welch's t-test over the classical Student's t-test is that it utilizes more statistical moments: While the test statistic for Student's t-test only requires the sample means, Welch's t-test also uses information from the estimated sample variances. The test variable is defined as:

$$t = \frac{\hat{\mu}_{s_i} - \hat{\mu}'_{s_i}}{\sqrt{\frac{\hat{\sigma}^2_{s_i}}{N} + \frac{\hat{\sigma}'^2_{s_i}}{N'}}} \tag{9}$$

Intuitively, the test variable t will have small absolute values if both samples are taken from the same distribution, i.e. the sample moments are similar. Strong discrepancies between both samples will result in large values for $|t|$. In principle, we could use this test statistic directly as measurement for our deviation, but it has turned out to be preferable to convert the t value into a probability p_t as a means of normalization. This can be achieved by considering the distribution of the t values for a fulfilled null hypothesis. If the null hypothesis is true, i.e., if both samples originate from the same probability density, the test statistic t follows a t-distribution with a degree of freedom which can be obtained by the Welch-Satterthwaite equation [27]. Based on the t-distribution, we can calculate the probability p_t by integration of the t-distribution.

Thus, the detailed steps to calculate the value of the $deviation$ function are:

- First, determine the required statistical moments for both samples: $\hat{\mu}_A, \hat{\sigma}^2_A, \hat{\mu}_B, \hat{\sigma}^2_B$.
- Calculate the test statistic t using Equation 9.
- Determine the degree of freedom of the underlying t-distribution $f_t(x)$. The problem of finding the degree of freedom is solved by the Welch-Satterthwaite equation.

- Calculate p_t by evaluating the area of the two-tail integral over $f_t(x)$ for $|x| > t$. This means that p_t is the probability to observe a larger absolute value than $|t|$ by chance if the null hypothesis is fulfilled.
- Finally, we set $deviation(\hat{\mu}_A, \hat{\sigma}_A^2, \hat{\mu}_B, \hat{\sigma}_B^2) = 1 - p_t$.

The second approach uses a two-sample Kolmogorov-Smirnov test to compare the distributions [29]. This test operates on the data samples themselves and does not rely on statistical moments. To calculate the deviation, we first have to build the empirical cumulated distribution functions for both samples. The empirical cumulated distribution function of a sample of x_{s_i} consisting of N objects is defined by:

$$F(x_{s_i}) = \frac{1}{N} \sum_{\vec{y} \in DB} I[y_{s_i} < x_{s_i}] \qquad (10)$$

where $I[cond]$ is the indicator function, equal to 1 if the condition $[cond]$ is fulfilled and equal to 0 otherwise. In other words, the value of F at a certain point x_{s_i} is the percentage of objects in the sample that have a value less than x_{s_i}. After the construction of F_A and F_B for the two samples, we can calculate the deviation as:

$$deviation(\hat{p}_A, \hat{p}_B) = \sup_{x_{s_i}} |F_A(x_{s_i}) - F_B(x_{s_i})| \qquad (11)$$

Thus, the deviation value is defined by the maximal difference of the two empirical cumulated distribution functions.

Comparing the two approaches for the statistical test, the Kolmogorov-Smirnov test features two favorable properties from a theoretical point of view. First, it uses the full information of the data samples and does not rely on the indirect calculation of statistical moments. The other problem with all types of t-tests is that the formal derivation requires Gaussian distributed samples. On the other hand, the Kolmogorov-Smirnov test does not make any assumptions on the sample distributions. Nevertheless, our evaluation in Section V shows that both approaches can achieve good results, even for datasets that differ significantly from a Gaussian distribution.

IV. HiCS Algorithm

Our algorithm consists of three logically independent parts:

- The calculation of the subspace contrast takes a specific subspace as input, and the output is its contrast.
- The subspace framework is responsible for the generation of subspace candidates that should be evaluated. All results are collected and will be filtered and sorted in a post-processing.
- The application of an outlier ranking on the list of high contrast subspaces.

A. *Contrast calculation*

The algorithm operates according to the sampling formalism in III-D. Besides the set of attributes that belong to the specific subspace, the algorithm requires two parameters:

- The number of Monte Carlo iterations M, i.e., the number of statistical tests to perform.
- The desired average size of the test statistic. In our implementation we specified the size by a ratio $\alpha \in (0, 1)$ that determines the sample size dynamically in relation to the total size of the database.

The idea of the adaptive subspace slices is implemented as follows: instead of defining the condition intervals $[l_i, r_i]$ directly in the domain of the underlying variables x_{s_i}, we precalculate one-dimensional index structures for all attributes of the database. This allows to perform the selection over the sorted indices. Thus, the adaptive selection of the subspace slice can by implemented by selecting a block of index entries with a certain size $\alpha_1 \cdot N$. The value of α_1 is determined by the parameter α and the dimensionality of the subspace $|S|$:

$$\alpha_1 = \sqrt[|S|]{\alpha}$$

The result of multiple selections can be obtained by a conjunctive boolean combination of the selection blocks. The adaptive random selection process is followed by the comparison between the marginal and the conditional distributions to obtain a deviation value.

In summary, the algorithm consists of these two steps: (1) generate a random subspace slice and (2) determine the respective deviation value using a statistical test. Finally, all deviation results will be combined to obtain a single contrast value for the subspace. The procedure is shown in Alg. 1.

Algorithm 1 calculation of subspace contrast

```
Input: $S$, $M$, $\alpha$
Output: $contrast(S)$
  for $i = 1 \rightarrow M$ do
    Permute list of subspace attributes $s \in S$
    Initialize boolean vector selected_objects for all objects: true
    for $i = 1 \rightarrow |S| - 1$ do
      Select random index block of attribute $s_i$ with a size of
      $N \cdot \sqrt[|S|]{\alpha}$
      Mask index block with selected_objects
    end for
    Compare distributions: $deviation\left(\hat{p}_{s_i},\ \hat{p}_{s_i|selected\_objects}\right)$ for the
    remaining attribute with $i = |S|$.
  end for
  Combine the results of all statistical test (cf. Definition 8).
```

B. *Subspace framework*

The subspace generation for HiCS works as follows: in each step we evaluate the contrast of the current d-dimensional subspaces. The subspaces that have a contrast above a certain threshold will be used for the generation of $(d+1)$-dimensional subspace candidates. This step-wise generation of higher dimensional subspace candidates resembles the principle of the well-known Apriori algorithm [3]. In contrast to Apriori, the HiCS starts with two-dimensional instead of one-dimensional subspaces, since the definition of a one-dimensional subspace contrast would not be reasonable (no notion of correlation). Another difference to Apriori is that it is not possible to formally derive a monotonicity criterion for the correlation of subspaces. To see this, we can construct a simple counterexample, such as the dataset shown in Figure 3. Each box corresponds to a cluster and all four clusters have the same density. In this example, the three-dimensional joint pdf is not given as the product of the three marginal distributions, i.e., the space is correlated. On the other hand, all two-dimensional subspace projections are equally distributed and, therefore, show no correlation at all. But this example also demonstrates that the construction of such a case requires an extremely specific setup. correlation is very likely to be visible in lower dimensional projections. Thus, one can combine lower dimensional subspaces to find correlations in higher dimensional spaces. Based on this heuristic, we can apply the Apriori-like subspace generation to the search of correlated subspaces.

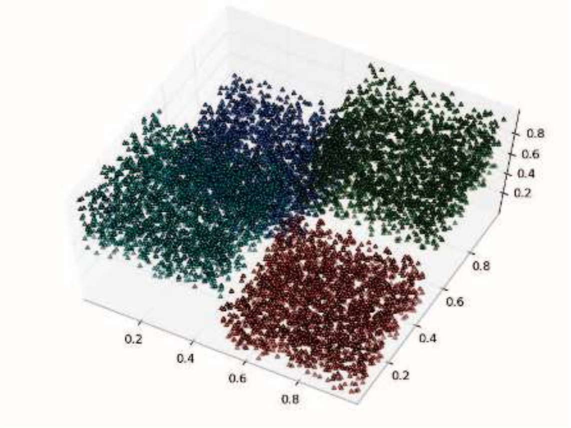


Fig. 3. High dimensional correlation

Like with other Apriori algorithms, the threshold for the candidate generation – in our case a lower bound on the contrast value – has a considerable impact on the results. Setting the value too high will result in a very restrictive subspace search, with only low dimensional subspaces or possibly even an empty list of subspaces. In contrast, if the value is much too low, the algorithm will consider almost all possible attribute combinations, resulting in an exponential runtime w.r.t. the total number of attributes.

Since our goal has been to design the algorithm in a way that allows a direct application to unknown datasets, we have solved this problem by means of an adaptive threshold. In contrast to conventional Apriori-like approaches, we postpone the decision whether to keep a candidate or not to the point when the contrast of all d-dimensional candidates is available. This allows to sort all current candidates and to keep only a certain number. We use the number of maximally retained candidates as parameter. Setting this *candidate_cutoff* parameter allows a much more precise prediction of the runtime than specifying a reasonable minimum contrast threshold for a specific dataset.

The subspace generation process terminates when the Apriori merge step produces an empty list for the $(d+1)$-dimensional subspace candidates. In the HiCS algorithm, the subspace generation is followed by a pruning step. The idea is to remove redundant subspaces from the output to ensure that the final subspace ranking contains only important subspaces [22]. We remove a redundant d-dimensional subspace T if the subspace list contains a $(d+1)$-dimensional subspace S that has a higher contrast score than T.

C. Subspace outlier ranking

As final step, HiCS has to apply an external outlier ranking algorithm to the list of detected subspaces and aggregate all results. For our evaluation we use LOF as outlier score [7]. As aggregation functions we considered maximum and average. In practice maximum is very sensitive to fluctuations of the outlierness and will lead to poor results especially if the number of detected subspaces is large. Therefore we have used the average of the outlier ranking values throughout our experiments (cf. Definition 1). This also ensures that the outlierness is cumulative: If an object deviates in several subspaces, its total outlierness will increase compared to objects that only appear as outlier in a single subspace.

V. Experiments

To evaluate the quality of our HiCS approach we perform experiments on synthetic and real world datasets. We treat the problem of outlier ranking independently from the selection of high contrast subspaces. Thus, we evaluate HiCS against a series of other subspace search algorithms as pre-processing to a common outlier ranking algorithm. We focus on LOF [7] as a widely used reference algorithm for full-space outlier mining. We abstract from any enhancements by recent or future techniques [25], [19], [23], [17], which can be used as instantiations of the outlier ranking as well. We compare HiCS against the following competitors:

- the full-space LOF outlier ranking [7]
- dimensionality reduction PCA [14] + LOF [7]
- the baseline approach using random subspaces [20]
- state-of-the-art subspace search: Enclus [8] and RIS [15]

For all subspace methods, we adapted LOF to measure object distances only w.r.t. the given subspace, as proposed by [20]. To ensure comparability, we applied the same LOF outlier model with identical parameter settings (i.e., the *MinPts* value) for all competitors. We use only the best 100 subspaces from the results of all subspace search methods, to enforce a concise subspace selection.

We quantify the quality of the obtained outlier rankings by calculating the *area under curve* (AUC) of the ROC curve. To ensure comparability for runtime evaluation, we implemented all algorithms in C++ and performed all experiments on an Intel® i3-550 Processor with 4 GB RAM. In addition, we provide all datasets and parameter settings online[1], to ensure repeatability of our experiments.

A. Experiments on synthetic data

For scalability experiments, we have generated synthetic datasets of different size and dimensionality. We randomly selected 2-5 dimensional subspaces out of the full data space and generated high density clusters in these subspaces. In each subspace we picked 5 objects and modified them to

[1] http://www.ipd.kit.edu/~muellere/HiCS/

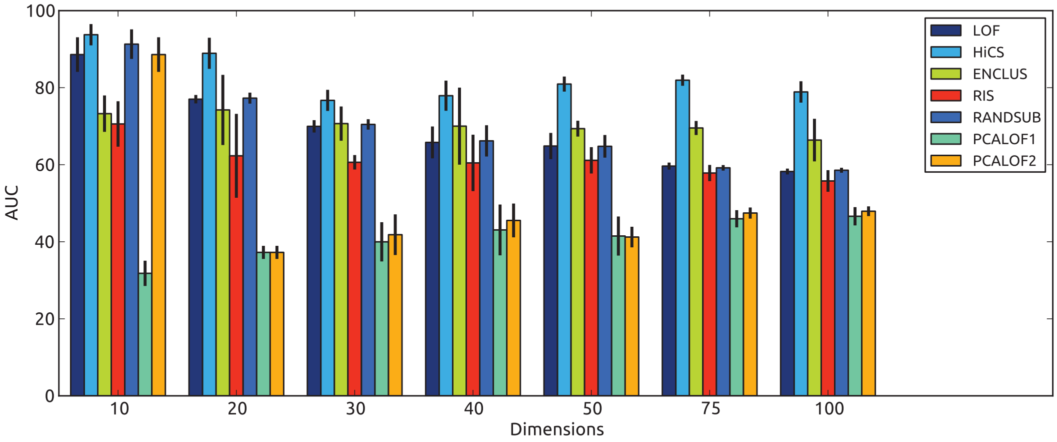


Fig. 4. Quality (AUC) of outlier rankings w.r.t. increasing dimensionality

deviate from all clusters in the selected subspace. To ensure the challenge of non-trivial outlier detection, this deviation was done in a way that the object will not be visible as outlier in any lower dimensional projection. Please note that this generation allows an object to be an outlier in multiple subspaces independently. This fulfills the real world observation of outliers hidden in multiple subspace projections (cf. Section I).

1) Quality evaluation: To evaluate the quality of HiCS we compare it with the competing algorithms in a series of experiments based on AUC. We focus on the scalability of all competitors w.r.t. the dimensionality of the data space. In Fig. 4 we depict the average AUC and its standard deviation for each algorithm (derived out of three randomly generated databases). HiCS outperforms the competing approaches. In particular, it scales with increasing dimensionality and shows high quality results even for high dimensional databases. Only Enclus shows similar scalability but with lower overall quality. However, a detailed examination of the subspaces selected by Enclus shows that it mainly found all two and some of the three-dimensional subspaces. This is expected because the grid based entropy measure is likely to fail for higher dimensional subspaces. In contrast, HiCS is able to detect even a high contrast in most of the five-dimensional subspaces. On the other hand, full-space runs of LOF show a degradation with increasing dimensionality, due to the curse of dimensionality. Traditional dimensionality reduction techniques such as PCA, should cope with the curse of dimensionality. However, as shown, PCA fails as pre-processing technique for outlier ranking. Please note that we have evaluated two strategies for dimensionality reduction: PCALOF1 (reduction to 50% of the total dimensionality) and PCALOF2 (constant reduction to 10 PCA-attributes). For the 10-dimensional datasets, the second strategy does not reduce the dimensionality, hence it shows the same quality as LOF. For all other cases PCA shows the worst performance (with AUC values close to 50%). This means that the resulting outlier ranking is equivalent to random guessing. We exclude PCA from further consideration, as preliminary experiments had indicated similar bad results for the following experiments as well.

2) Runtime evaluation: In addition to the quality evaluation, we depict the runtime w.r.t. increasing dimensionality in Fig. 5. All experiment runs are identical to the previous experiment on quality evaluation, but we consider only the competitors that are based on subspace rankings. We always specify the total processing time, i.e., the time for both the subspace search and the outlier detection. Overall, the results show the scalable processing of HiCS. In particular we observe almost no increase in runtime for more than 30 dimensions. This results in a runtime comparable to the simple grid-based processing of Enclus, which is the fastest algorithm in this test but with drawbacks in terms of quality. This scalability effect of HiCS is due to our *candidate_cutoff* parameter in the subspace generation framework. It is set to 400 in this experiment. For the experiments with a dimensionality below 30, HiCS never generated more than 400 candidates. Thus, the runtime increases with more dimensions and more possible combinations of attributes. When we reach 40 dimensions, the cutoff is applied for the first time. It ensures both high quality, by maintaining the top-400 highest contrast subspaces, and low runtime, by pruning low contrast subspaces.

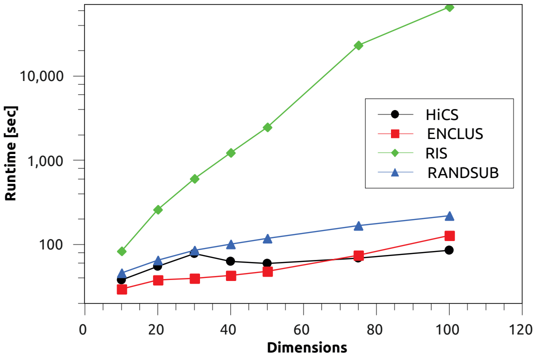


Fig. 5. Runtime w.r.t. dimensionality D, with fixed DB-size 1000

Besides the scalability w.r.t. data dimensionality, we have been interested in scalability w.r.t. the database size. The experimental results are shown in Fig. 6. The minimum runtime of all competitors is determined by the runtime of LOF and the number of selected subspaces. The latter one is fixed for all algorithms to the 100 most promising subspaces. Due to the quadratic complexity of the LOF algorithm, we expect at least a quadratic total processing time for all competitors. For RIS we observe a cubic complexity w.r.t. the database size, and accordingly this technique does not scale very well. For HiCS and Enclus, the overhead for the subspaces detection is almost negligible if the database is sufficiently large. If we compare these two subspace search algorithms to the naive random selection, we observe that RANDSUB actually consumes more time. This is because it generates much larger subspaces on average. This seems to have a bigger impact on the runtime than the execution of a subspace search algorithm.

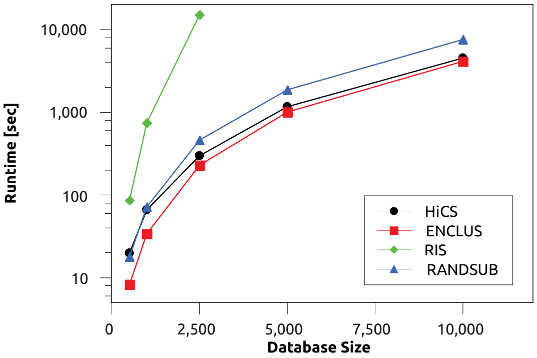


Fig. 6. Runtime w.r.t the DB-size, with fixed dimensionality 25

3) Parameters: In our comprehensive quality experiment (cf. Fig. 4), we have noticed a high sensitivity w.r.t. parametrization for our competitors. For RIS and Enclus in particular, we have observed that finding good parameter settings is difficult. Therefore we had run the whole experiment with a large number of configurations for these two algorithms. We have shown only the best values in the previous graphs. To evaluate the robustness of our parameter settings, we describe more detailed experiments in the following. We evaluate both variants of our statistical instantiation HiCS_{WT} and HiCS_{KS} as defined in Section III-E, and we used HiCS_{WT} as default setting in all other experiments.

The first parameter is the number of statistical tests M that are performed for each subspace or, in other words, the number of iterations of the Monte Carlo algorithm. This tradeoff between runtime and the influence of statistical fluctuations does not have a critical impact on the results. Fig. 7 shows the AUC quality measure contingent on the number of statistical tests. We recommend to use 50 as a default value for this parameter, as used in all other experiments.

Furthermore, we evaluated the influence of the test statistic size α as depicted in Fig. 8. The experiment shows that the resulting quality is fairly robust w.r.t. the parameter α. For very low values ($\alpha < 5\%$, i.e., less than 50 objects in this experiment) we noticed a slightly increased fluctuation of the quality. This effect becomes more important when we also reduce the number of statistical tests. Thus, having more statistical tests helps to decrease the influence of α. For larger α values, the statistical tests are less sensitive, resulting in a minor quality reduction.

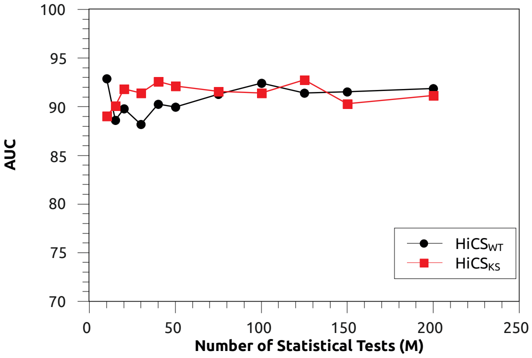


Fig. 7. Dependence on the number of statistical tests (M)

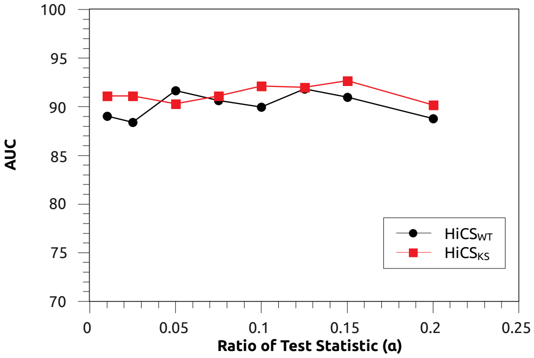
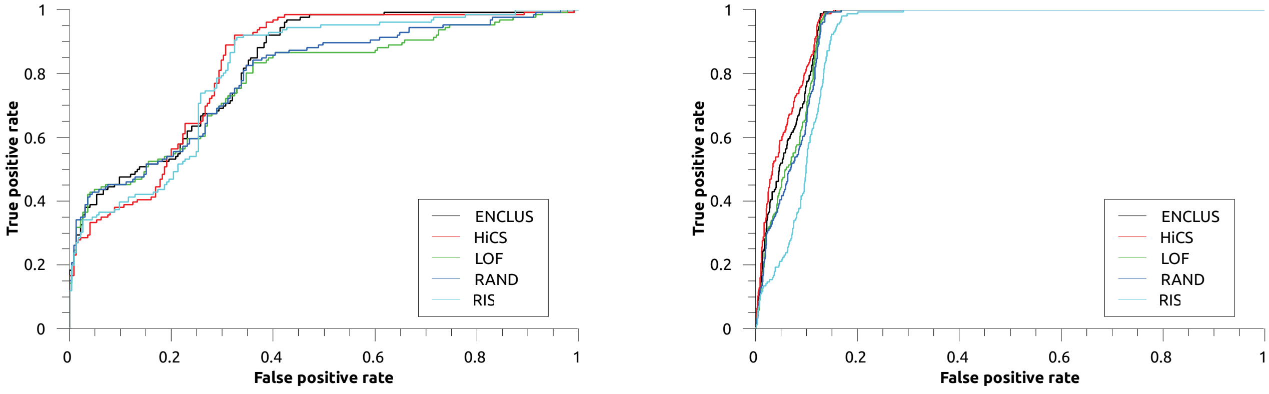


Fig. 8. Dependence on the size of the test statistic (α)

The last parameter *candidate_cutoff* limits the number of candidates in the bottom-up subspace processing. Thus, it influences the total runtime and the maximal dimensionality in the subspace ranking. To avoid any dataset dependence of this parameter, we have evaluated the qualities on several synthetic datasets. The following graphs always show average values. In Fig. 9 we can see a peak in the quality at around 500. For lower values, the quality is reduced, since the cutoff may remove some good candidates from the subspace list. The reason for this quality decrease can be found by analyzing the resulting subspace ranking: We observed that the selection starts to contain many redundant subspaces. This redundancy leads to a slight quality loss in the resulting outlier score. Please note that the fluctuations introduced by this parameter still are relatively small if we compare them to the results in Fig. 4. In addition to the quality evaluation we depict the influence of the cutoff parameter on the runtime in the lower part of Fig. 9. We see that the *candidate_cutoff* parameter allows to control the total runtime precisely. In combination with the previous quality experiments we conclude that not all candidates are required and can be pruned without a significant quality loss.

B. Experiments on real world data

To evaluate HiCS in a real life situation, we chose eight real world benchmark datasets from the UCI ML Repository [12]: Thyroid (ANN version), Arrhythmia, Breast Cancer, Breast Cancer Wisconsin (Diagnostic), Diabetes, Glass, Ionosphere



(a) Ionosphere

(b) Pendigits

Fig. 10. ROC plots for two real world experiments

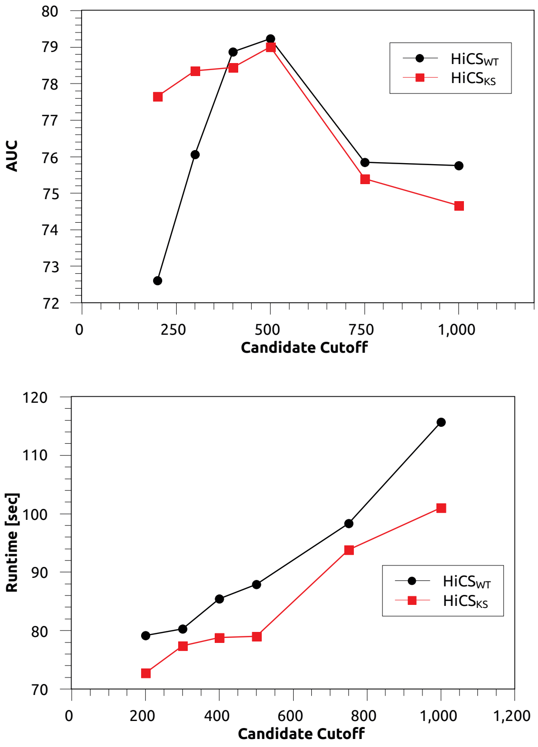


Fig. 9. Quality and Runtime w.r.t. candidate cutoff parameter

and Pendigits. Since outlier mining is conceptually similar to detecting objects that belong to a rare class, we focused on datasets where the class definitions featured a clear minority class. We assume this class to contain the outliers in these datasets. For the Pendigits dataset, all classes have equal frequencies. In this case we reduced the number of objects for one class (corresponding to the digit "0") by a factor of 10%.

The results of all real world experiments are shown in Fig. 11. The best AUC values are highlighted in bold, and high quality results that are within 1% of the best are not grayed out. The results demonstrate that HiCS achieves a very good overall performance. It is the best algorithm for three datasets and is close to the best result in four other experiments. Other approaches achieve high quality only for a small subset of the datasets and show a higher quality variation depending on the dataset used. HiCS is the only algorithm with high quality on most of the datasets. Considering runtime, HiCS is among the fastest subspace search algorithms. Only Enclus shows similar runtimes.

In addition, we show two ROC curves for the Ionosphere and Pendigits datasets in Fig. 10. It is interesting to note that the HiCS algorithm shows a tendency to reach the maximal true positive rate earlier than other methods. Thus, it is perfect for applications that require a high recall of outliers with best precision of the outlier ranking. On the other hand, we observe a minor weakness of HiCS if one is interested in very low false positive rates: In the Ionosphere dataset for example, the outlier ranking seems to miss some full space outliers. This results in a reduced steepness of the ROC curve for low false positive values. The reason for this might be the focus on multi-dimensional subspaces. After all, we did not remove any outliers that are trivially visible in one-dimensional projections. Therefore it might be possible to improve the quality of HiCS even further by applying a pre-processing step that takes care of the detection of trivial outliers. This would result in even higher quality, while the overall results of all AUC values show that we already obtain very good quality without such a pre-processing. Overall, HiCS shows excellent results on a broad variety of datasets, with robust and easy-to-use parameters, and a scalable processing w.r.t. the dimensionality of databases.

VI. Conclusion and Future Work

In this work, we developed an approach that is able to detect subspaces for outlier mining in high dimensional databases. We proposed the first subspace search method that selects high contrast subspaces for density-based outlier ranking. We focus

Experiment	**AUC [%]**					**Runtime [sec.]**				
	LOF	HiCS	Enclus	RIS	RANDSUB	LOF	HiCS	Enclus	RIS	RANDSUB
Ann-Thyroid	86.16	95.11	94.32	**95.16**	93.32	7.1	37.2	68.1	574.0	674.0
Arrhythmia	62.92	62.29	62.11	**63.61**	63.52	0.5	26.4	7.9	2216.1	48.2
Breast	56.42	59.31	**59.55**	-	56.98	0.1	2.4	1.5	-	3.5
Breast (diagnostic)	86.94	**94.23**	94.19	90.77	87.07	0.3	15.8	11.8	14.3	28.2
Diabetes	70.98	**72.47**	71.15	71.63	71.70	0.3	3.3	5.9	4.0	26.2
Glass	76.86	80.05	79.73	**80.65**	78.48	0.0	0.2	0.3	0.1	1.7
Ionosphere	77.97	82.34	**82.37**	80.93	79.02	0.1	6.1	4.2	668.2	11.0
Pendigits	93.54	**95.04**	94.29	90.74	93.22	34.1	1194.5	2195.6	11282.7	3326.2

Fig. 11. Results on real-world datasets

on the detection of outliers that are neither visible in the full space nor in a single attribute. These non-trivial outliers show up in high contrast subspace with a strong correlation in the selected dimensions. In our two-step approach, we measure the contrast of subspaces and select the most promising ones for outlier ranking. In this decoupled processing, we propose a first contrast measure based on correlation analysis. It uses the difference between marginal and conditional pdf of a subspace as a criterion for high contrast. The extensive set of experiments shows that our HiCS approach outperforms existing subspace search techniques, both on synthetic and on real world datasets.

For future work, we aim at further evaluations with other outlier scores such as ORCA [5] or OUTRES [23]. Both seem very promising extensions of LOF with enhanced outlier scoring. ORCA would improve the efficiency from a quadratic to a linear runtime in the outlier ranking step. OUTRES might improve the quality of our outlier ranking due to its adaptive density scoring in subspace projections. Due to the decoupled processing, our subspace search can be applied directly to these or other outlier scores.

Furthermore, we would like to extend the research on subspace selection and enhance our subspace search based on other outlier ranking paradigms. Although HiCS would be applicable to these paradigms, transferring some specific properties out of the underlying definition into the subspace search might result in further quality improvements. Overall, the flexible two-step processing opens a wide range of research challenges in the domain of subspace outlier mining.

Acknowledgment: This work was supported by the German Research Foundation (DFG) within the Research Training Group GRK 1194 “Self-organizing Sensor-Actuator-Networks”.

REFERENCES

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

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, “Automatic subspace clustering of high dimensional data for data mining applications,” in *SIGMOD*, 1998, pp. 94–105.

[3] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in *VLDB*, 1994, pp. 487–499.

[4] C. Baumgartner, C. Plant, K. Kailing, H.-P. Kriegel, and P. Kröger, “Subspace selection for clustering high-dimensional data,” in *ICDM*, 2004, pp. 11–18.

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

[6] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, “When is nearest neighbors meaningful,” in *IDBT*, 1999, pp. 217–235.

[7] M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander, “LOF: identifying density-based local outliers,” in *SIGMOD*, 2000, pp. 93–104.

[8] C.-H. Cheng, A. W. Fu, and Y. Zhang, “Entropy-based subspace clustering for mining numerical data,” in *KDD*, 1999, pp. 84–93.

[9] T. de Vries, S. Chawla, and M. E. Houle, “Finding local anomalies in very high dimensional space,” in *ICDM*, 2010, pp. 128–137.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases,” in *KDD*, 1996, pp. 226–231.

[11] P. Filzmoser, R. Maronna, and M. Werner, “Outlier identification in high dimensions,” *Comp. Stat. Data Anal.*, vol. 52, no. 3, pp. 1694–1711, 2008.

[12] A. Frank and A. Asuncion, “UCI machine learning repository,” 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[13] A. Ghoting, S. Parthasarathy, and M. Otey, “Fast mining of distance-based outliers in high-dimensional datasets,” *Data Mining and Knowledge Discovery*, vol. 16, pp. 349–364, 2008.

[14] I. Joliffe, *Principal Component Analysis*. Springer, New York, 1986.

[15] K. Kailing, H.-P. Kriegel, P. Kröger, and S. Wanka, “Ranking interesting subspaces for clustering high dimensional data,” in *PKDD*, 2003, pp. 241–252.

[16] E. Knorr and R. Ng, “Algorithms for Mining Distance-Based Outliers in Large Datasets,” in *VLDB*, 1998, pp. 392–403.

[17] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, “Interpreting and unifying outlier scores,” in *SDM*, 2011, pp. 13–24.

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

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

[20] A. Lazarevic and V. Kumar, “Feature bagging for outlier detection,” in *KDD*, 2005, pp. 157–166.

[21] E. Müller, M. Schiffer, and T. Seidl, “Statistical selection of relevant subspace projections for outlier ranking,” in *ICDE*, 2011, pp. 434–445.

[22] E. Müller, I. Assent, S. Günnemann, R. Krieger, and T. Seidl, “Relevant subspace clustering: Mining the most interesting non-redundant concepts in high dimensional data,” in *ICDM*, 2009, pp. 377–386.

[23] E. Müller, M. Schiffer, and T. Seidl, “Adaptive outlierness for subspace outlier ranking,” in *CIKM*, 2010, pp. 1629–1632.

[24] D. Niu, J. G. Dy, and M. I. Jordan, “Multiple non-redundant spectral clustering views,” in *ICML*, 2010, pp. 831–838.

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

[26] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*. Wiley, 1987.

[27] F. E. Satterthwaite, “An approximate distribution of estimates of variance components,” *Biometrics Bulletin*, vol. 2, no. 6, pp. 110–114, 1946.

[28] C. Spearman, “The proof and measurement of association between two things,” *American J. of Psych.*, vol. 15, no. 1, pp. 72–101, 1987.

[29] M. Stephens, “Use of the Kolmogorov-Smirnov, Cramer-von Mises and related statistics without extensive tables,” *J. of the Royal Stat. Society*, pp. 115–122, 1970.