Flexible and Adaptive Subspace Search for Outlier Analysis

Fabian Keller° Emmanuel Müller°• Andreas Wixler° Klemens Böhm°

°Karlsruhe Institute of Technology (KIT), Germany
{fabian.keller, emmanuel.mueller, klemens.boehm}@kit.edu
andreas.wixler@student.kit.edu

•University of Antwerp, Belgium
emmanuel.mueller@ua.ac.be

ABSTRACT

There exists a variety of traditional outlier models, which measure the deviation of outliers with respect to the full attribute space. However, these techniques fail to detect outliers that deviate only w.r.t. an attribute subset. To address this problem, recent techniques focus on a selection of subspaces that allow: (1) A clear distinction between clustered objects and outliers; (2) a description of outlier reasons by the selected subspaces. However, depending on the outlier model used, different objects in different subspaces have the highest deviation. It is an open research issue to make subspace selection adaptive to the outlier score of each object and flexible w.r.t. the use of different outlier models.

In this work we propose such a flexible and adaptive subspace selection scheme. Our generic processing allows instantiations with different outlier models. We utilize the differences of outlier scores in random subspaces to perform a combinatorial refinement of relevant subspaces. Our refinement allows an individual selection of subspaces for each outlier, which is tailored to the underlying outlier model. In the experiments we show the flexibility of our subspace search w.r.t. various outlier models such as distance-based, angle-based, and local-density-based outlier detection.

Categories and Subject Descriptors: H.2.8 Database Management: Database Applications [Data mining]

Keywords: data mining; subspace search; high dimensional data; outlier detection; outlier description

1. INTRODUCTION

Outlier analysis is a widely used data mining task. It aims at the detection and the description of exceptional objects in a database. For instance, applications such as sensor networks, gene expression analysis, or health surveillance systems use outlier analysis to identify irregular, suspicious, or unexpected measurements.

However, in many of today's applications objects are described by a large number of attributes, and outliers are likely to deviate only w.r.t. a subset of attributes (a so-called *subspace*). Traditional full-space outlier mining approaches [2] fail to detect these subspace outliers since the detection process considers a large number of irrelevant attributes. Recent outlier mining techniques tackle this problem by searching for relevant subspaces. The objective is to find a subset of attributes with a significant deviation between an outlier and regular objects. For example in Figure 1, o_1 deviates only w.r.t. $S_1 = \{$*Voltage Magnitude, Harmonic Content*$\}$. Using a deviation measure in this subspace allows to clearly detect the object as an outlier. In other subspaces, for instance in $S_3 = \{$*Harmonic Content, Transient Voltage*$\}$, o_1 is regular. Examining the attributes of S_1 together with irrelevant ones such as *Transient Voltage* tends to miss that anomaly. Hence, it is an important issue to select relevant attributes in order to detect outliers. The selection is also important for outlier description: Detecting the relevant subspace, for instance S_1 for o_1, provides a valuable explanation why this object is anomalous. The subspace serves as a description of the anomalous properties and assists manual outlier verification.

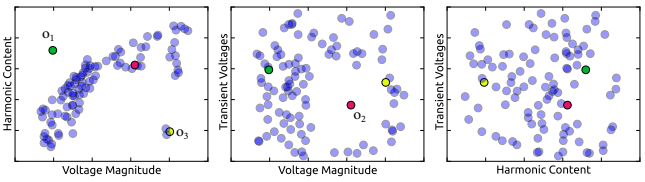

Figure 1: Example of different outliers in subspaces

However, the characteristics of an anomalous object can vary largely depending on the application domain. To tackle this problem, many different outlier models have been proposed. Each one considers different outlier properties. For instance, some models are sensitive to distance deviations [14]; others capture deviation in the local density [7]; yet other models prefer angle-based [19] or statistical deviation [24]. Examples of three different notions are given in Figure 1 with o_1 as a local density-based outlier, o_2 as distance-based outlier, and o_3 detected either as outlier or as part of a micro-cluster depending on parameter settings. As each outlier model is meaningful for different application domains, we do not want to discuss the pros/cons or even parametrization of different models in this work. We focus on the orthogonal problem of *subspace search* and provide a general solution that can use any of these outlier models for enhanced subspace search.

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505560.

Due to the exponentially large number of subspaces, it is an open challenge to select the relevant subspaces for each outlier. Existing techniques from the field of *subspace outlier mining* perform such a combinatorial search [3, 18, 22, 17]. However, they all rely on a fixed outlier model, which cannot be exchanged depending on the application domain. On the other hand, *subspace search* techniques [8, 12, 13, 23] are agnostic w.r.t. the outlier model, which only is applied as a post-processing step. They ignore the underlying outlier definition and focus on global data characteristics such as entropy, density, and comparison of distributions. Depending on the outlier model however, different objects in different subspaces have the highest deviation. Hence, subspace search results should be tailored to the outlier characteristics of individual outliers. To solve this problem, the approach envisioned must be both *flexible* (the method allows to exchange the outlier model at all) and *adaptive* (the method performs the search tailored to the outlier model). A subspace search scheme with these properties is applicable to a broad range of application domains. Flexibility also ensures that the approach directly benefits from any research progress on traditional outlier models. Adaptiveness allows to search for relevant subspaces individually for each outlier and, hence, enables to describe each outlier by its specific outlier properties.

As the main contribution of this paper we propose RefOut, a flexible and adaptive subspace search framework for outlier mining. It finds relevant subspaces by a refinement process that adapts to the given outlier model. The key idea is based on the observation that traditional outlier detection methods (applied to subspaces) do capture at least small deviations of an outlier even though some irrelevant attributes are included. In the distance-based outlier model for instance, o_2 is a clear outlier in subspace $S_2 = \{$*Voltage Magnitude*, *Transient Voltage*$\}$. In a high dimensional database it is hard to detect this subspace directly. But when considering random subspaces T with $|T| \geq |S_2|$, some of these random spaces will contain S_2. When applying the distance-based model to evaluate o_2 in such a space $T \supseteq S_2$ the model will report a relatively high outlier score. In contrast to this, we measure relatively low outlier scores in all other spaces $T \not\supseteq S_2$ in which o_2 shows no irregular behavior w.r.t. the distance-based model. Our main idea is to detect these *score discrepancies* of high outlier scores in $T \supseteq S_2$ over low scores in $T \not\supseteq S_2$ for individual objects. We extract information hidden in outlier scores to make a conclusion which subspace induces a high outlier score for the given outlier model. We use this information to refine a pool of random subspaces according to the discrepancies in the outlier scores. This means that if we for instance perform RefOut with an angle-based outlier model in our example, it would ignore o_2 and S_2 and instead focus on angle-based outliers and their respective subspaces.
With RefOut, we make the following contributions:

- We formalize outlier characteristics in different subspaces as profiles and use these in our adaptive search.
- We derive the *score discrepancy problem*, which provides a new theoretical perspective on subspace search.
- We propose the first subspace search approach based on the *score discrepancy problem* providing outlier descriptions for individual objects.

To the best of our knowledge RefOut is the first subspace search technique that is both flexible and adaptive w.r.t. different outlier models. In our experiments we show that this adaptivity leads to an enhanced quality for various outlier models.

2. RELATED WORK

A number of different outlier paradigms have been proposed in the literature. We review the main directions and highlight the difference to our approach.

Traditional Outlier Mining: We use the term traditional outlier mining to refer to any outlier scoring technique that operates on a fixed attribute set. There are various full-space outlier models ranging from deviation-based methods [24], distance-based methods [14], local density-based methods [7] right up to angle-based methods [19] or hashing-based scoring [25]. We abstract from these individual models and propose to use an abstract outlier score in our framework. Our research is orthogonal to the development of novel outlier scores, i.e., RefOut benefits from any future improvements of traditional outlier mining w.r.t. quality, efficiency, or novel outlier definitions.

Mining Descriptions (for given outliers): There are several approaches that identify subspaces as so-called outlier descriptions [15, 5, 21]. These methods extract a subspace for a given outlier, assuming that outlier detection has taken place in advance. Obviously this results in a *chicken and egg dilemma*: $(1 \rightarrow 2)$ Traditional outlier detectors require a prior subspace selection to detect outliers hidden in subspaces. $(2 \rightarrow 1)$ Outlier descriptions would provide such a subspace selection, but they require the outliers to be detected in advance. With the proposed RefOut approach, we break this cyclic dependency by solving these two problems simultaneously. Our search process applies to both outliers and the corresponding subspaces.

Subspace Outlier Mining: Subspace outlier mining was first specified by [3], and recent approaches have extended this idea to subspace outlierness scores [18, 9, 22, 17]. They propose an interleaved detection of both outliers and subspaces. This means that each of these techniques relies on some specific outlier criterion that is tailored to its subspace processing. They are restricted to this outlier criterion, and thus, are not flexible w.r.t. instantiations with different outlier models. In contrast to these methods, RefOut not only allows to exchange the outlier model, it also adapts its subspace search to the actual outlier score detected by the model.

Subspace Search: Approaches from the field of subspace search [8, 12, 13, 23] in turn focus on the selection of subspaces. They can be used as a pre-processing step to any traditional outlier mining algorithm. Thus, subspace search allows to exchange the outlier model. While flexibility is fulfilled, adaptivity is not: The outlier model is ignored, and the subspace search does not take the specific characteristics of the given outlier model into account.

3. BASIC NOTIONS

Let DB be a database consisting of 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. For calculations in specific subspaces we constrain the vectors to the respective attributes, i.e., $\vec{x}_S = (x_{s_1}, \ldots, x_{s_d})$. This allows to deploy notions such as distance, density, and outlierness directly at the subspace level.

To define an adaptive outlier detection framework, we formalize the notion of an outlier model:

DEFINITION 1. *An **outlier model** is a function that maps every object of the database to a real-valued **outlier score** w.r.t. a given subspace S:*

$$score(\vec{x}_S) \in \mathbb{R} \quad \forall \vec{x} \in DB$$

3.1 Pre-processing Outlier Scores

Since our framework evaluates individual objects in different subspaces, the only necessary requirement is that the outlier scores are comparable among different subspaces. Most outlier models do not immanently provide this comparability among subspaces. However, comparability can always be enforced by applying a normalization scheme. We assume that the normalization ensures that the outlierness distribution of the majority of regular objects has *(1)* a mean of $default_{out}$ and *(2)* a variance of 1 independent of S. For examples of such normalization schemes for arbitrary outlier models we refer to unification techniques [16]. For the outlier models used in this work we obtain the required properties by applying the following transformation:

$$\overline{score}_S = \frac{1}{N} \sum_{\vec{x} \in DB} score(\vec{x}_S) \quad (1)$$

$$Var(score_S) = \frac{1}{N-1} \sum_{\vec{x} \in DB} (score(\vec{x}_S) - \overline{score}_S)^2 \quad (2)$$

$$score'(\vec{x}_S) = (score(\vec{x}_S) - \overline{score}_S)/\sqrt{Var(score_S)} \quad (3)$$

In the remainder of this work, we apply this transformation to all outlier models utilized. For the sake of presentation, we also assume an increasing sort order of $score(\vec{x}_S)$, i.e., higher values correspond to stronger outlier characteristics. Finding alternative normalization schemes is orthogonal to our work. We focus on the selection of subspaces only and use this existing pre-processing scheme.

3.2 Formalization of Outlier in Subspaces

In the following we focus on one individual object $\vec{x}$ and formalize the outlier score properties evaluated over different subspaces by keeping one subspace S fixed for comparison.

DEFINITION 2. *The **outlierness profile** of an individual object $\vec{x}$ w.r.t. subspace S is a function over random subspaces T with $|T| = d'$ defined as*

$$profile_{\vec{x},S}(d') = \begin{cases} E\left[score(\vec{x}_T)\right] & \text{with } T \subset S, \text{ for } d' < |S| \\ score(\vec{x}_S) & \text{, for } d' = |S| \\ E\left[score(\vec{x}_T)\right] & \text{with } T \supset S, \text{ for } d' > |S| \end{cases}$$

Based on this outlier profile, we are able to compare the outlier score of $\vec{x}$ in subspace S with all of its super- and sub-spaces T. Considering various spaces T with different dimensionality d' we derive the definition of a true subspace outlier as follows:

DEFINITION 3. *An object $\vec{x}$ is a **true subspace outlier** with respect to subspace S iff*

$$profile_{\vec{x},S}(|S|) = \max_{d' \in 1 \ldots D} profile_{\vec{x},S}(d') \gg default_{out}$$

*We call this maximum value the **peak** of $\vec{x}$ in subspace S and we further require:*

$$\begin{array}{ll} profile_{\vec{x},S}(d') \ll peak & \forall d' < |S| \\ profile_{\vec{x},S}(d') > default_{out} & \forall d' > |S| \\ profile_{\vec{x},S}(d') < profile_{\vec{x},S}(d'-1) & \forall d' > |S| \end{array}$$

*We will also refer to **true subspace outliers** as **d-dimensional outlier** with $d = |S|$, the dimensionality of the subspace.*

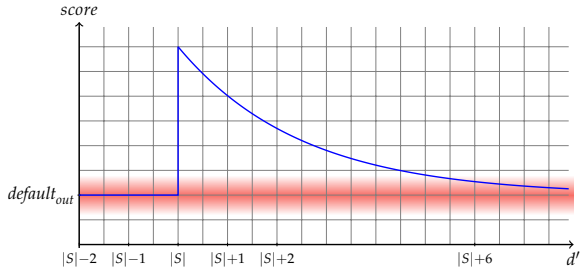

Figure 2: Ideal profile of a true subspace outlier

Figure 2 illustrates these definitions. The plot shows an idealized outlierness profile of an individual object (blue line) that fulfills the true subspace outlier conditions. The red area shows the distribution of regular objects (unit variance as a result of normalization). At $d' = |S|$, we can see the clear outlier score peak, deviating by several standard deviations.

When considering random superspaces of S ($T \supset S$), the expectation value of the outlier score decreases monotonically. It is precisely this manifestation of the curse of dimensionality [6] that is commonly observed in reality: Adding irrelevant attributes hampers the outlier detection. Thus, the measured outlier score decreases with increasing dimensionality since all objects become more and more alike. Comparing the blue curve of an individual outlier with the red distribution of regular objects shows that at some point the deviation of our true subspace outlier is comparable with the average deviation of regular objects. Thus, it is no longer possible to detect the true subspace outlier.

For lower dimensional subspaces $d' < |S|$ the object is projected in random subspaces of S. The defining property $profile_{\vec{x},S}(d') \ll peak$ for these spaces means that the true subspace outlier is projected into regions of regular densities in these subspace projections. This effect is also very common in reality. Think of o_1 from our example in Figure 1. This object clearly has a $peak \equiv profile_{\vec{o}_1,S_1}(2)$. Projecting the two dimensional subspace $S_1 = \{$*Voltage Magnitude*, *Harmonic Content*$\}$ to its one-dimensional subspaces will project o_1 into regions of high density. In none of these subspaces the object shows an exceptional outlier score, thus, $profile_{\vec{o}_1,S_1}(1) \ll peak$. By assuming that no other attribute contributes to the deviation of o_1, all properties are fulfilled and o_1 is a true subspace outlier in S_1.

Regarding higher dimensional true subspace outliers (i.e. large $|S|$), the condition $profile_{\vec{x},S}(d') \ll peak \; \forall d' > |S|$ implies that the object is not exceptional in *all* lower dimensional projections. For instance, a true subspace outlier in a 4-dimensional subspace S appears to be regular in all 3-, 2-, and 1-dimensional projections of S. Only the joint consideration of all attributes makes the object exceptional, and no single attribute of S is responsible for the anomalousness alone. This property of true subspace outliers makes

their detection exceptionally hard. Note that, if an object deviates in for instance two attributes s_1 and s_2, this object is not a true subspace outlier in $S = \{s_1, s_2\}$ since it suffices to clearly detect the outlier by considering the attributes separately. Thus, we would consider this object to be a true (1-dimensional) subspace outlier in both $S_1 = \{s_1\}$ and $S_2 = \{s_2\}$.

Please note that our definition of true subspace outliers is not a binary definition. For our detection framework we output the size of the peak as final outlier score for each object. Thus, we provide an *outlier ranking* with the most prominent true subspaces outliers ranked first.

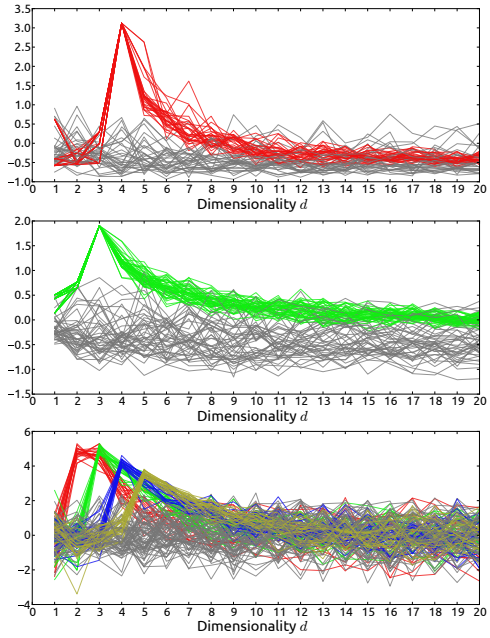

Figure 3: Examples of outlierness profiles

To corroborate our model of outlierness profiles and of true subspace outliers, Figure 3 shows examples of real outlierness profiles. For the sake of illustrating outlierness profiles we introduce profile instantiations: We draw a single line corresponding to one specific sequence of random subspaces T over the dimensionality range (each point corresponds to the outlier score in a random subset/superset of S; $T = S$ at the peak). This allows to visualize outlier score distributions and expectation values by plotting a large number of these instantiations. The first figure shows a real world outlier from the Breast dataset, detected as 4-dimensional true subspace outlier in our evaluation. The outlierness profile was generated based on the local density outlier model. As a reference we show profiles of regular objects in gray. The second figure shows the outlierness profile of a different object with a 3-dimensional peak, this time evaluated with a distance-based model. Overall, the observed outlierness profiles are in good agreement with our model. In fact, such observations of true subspace outliers on real world data were the primary motivation for the development of RefOut. We also generate our synthetic data according to these observations (cf. Sec. 5) and include hidden outliers of different subspace dimensionalities. The third figure shows examples from our synthetic data; this time evaluated with an angle-based model. Note that the three examples are generated based on outlier models with vast differences in their raw outlier score distributions, but the general shape of outlier profiles is preserved after normalization.

All these examples illustrate the need for subspace selection: Outliers can be clearly detected in the peaking subspace. In addition, this subspace is a valuable description of the individual outlier characteristics.

4. REFOUT TECHNIQUE

Our RefOut approach consists of two building blocks. The first one is the definition of a general framework for an adaptive subspace analysis based on traditional outlier scores. The underlying idea is based on the transformation of the subspace search problem into a score discrepancy analysis problem (Sec. 4.1 and Sec. 4.2). The second building block of RefOut deals with the question of how to solve this novel score discrepancy problem. We will propose our solution in Sec. 4.3.

4.1 The Score Discrepancy Problem

Identifying outlier in subspaces is computationally expensive. In principle, an exhaustive search for true subspace outliers requires scanning through all possible subspaces $2^{\mathcal{A}}$ for each object in the database. Due to the exponential number of subspaces, this would only be feasible for very low dimensional databases. To achieve a scalable subspace outlier detection it is necessary to drastically reduce the search space. To this end, we follow the idea of random subspace sampling [20] as a basis for our adaptive subspace search.

In order to take a new perspective on the subspace search problem, we look at the effects of applying a given outlier model in subspaces selected randomly. In the following, we focus on a single object $\vec{x}$ that is a true subspace outlier in subspace S under the outlier model. To simplify the analysis, we further assume that the object $\vec{x}$ is not a true subspace outlier in any other subspace. We denote the set of irrelevant attributes as $I = \mathcal{A} \setminus S$.

Let T be a random variable of subspaces, i.e., T is drawn uniformly from $2^{\mathcal{A}}$. We refer to the sample over these random subspaces as *subspace pool* $\mathcal{P} = \{T \mid T \text{ drawn iid from } 2^{\mathcal{A}}\}$. By applying the given outlier model to the random subspaces T, we obtain a sample of outlier scores:

$$\mathcal{O} = \{score(\vec{x}_T) \mid T \in \mathcal{P}\}$$

The subspace S of $\vec{x}$ plays an important role in the random sampling process: It partitions both the subspace pool $\mathcal{P}$ and the outlier scores $\mathcal{O}$ depending on whether the random subspace T is a superset of S or not. We denote the split of the subspace pool $\mathcal{P}$ as

$$\mathcal{P}_S^+ = \{T \mid T \supset S \wedge T \in \mathcal{P}\} \quad \mathcal{P}_S^- = \{T \mid T \not\supset S \wedge T \in \mathcal{P}\}$$

and the partition of the outlier scores $\mathcal{O}$ as:

$$\mathcal{O}_S^+ = \{score(\vec{x}_T) \mid T \supset S \wedge T \in \mathcal{P}\}$$

$$\mathcal{O}_S^- = \{score(\vec{x}_T) \mid T \not\supset S \wedge T \in \mathcal{P}\}$$

We now examine the two outlier score populations $\mathcal{O}_S^+$ and $\mathcal{O}_S^-$ by considering our observations w.r.t. the outlierness profiles. We know that for the spaces in $\mathcal{P}_S^+$, the outlier score is described by the outlierness profile (cf. Fig. 2), since they are supersets of the true subspace S. This means that for score $o \in \mathcal{O}_S^+$ we have $E[o] > default_{out}$, i.e., the expectation value of the score is increased over $default_{out}$. Note that this observation only applies for the expectation value of

the score; in reality one can obtain an $o < default_{out}$ by chance.

For the spaces $T \in \mathcal{P}_S^-$ the true subspace S is never completely covered. We have to consider two cases when analyzing the population $\mathcal{O}_S^-$. The first case is that T partially covers S, i.e., T includes some but not all attributes of S. This means that we obtain a subspace which projects the true subspace outlier into a region of regular density. Regarding the outlierness profile, this corresponds to the left side of the peak. Thus, in this case we have $E[o] \approx default_{out}$ for $o \in \mathcal{O}_S^-$. The second case is that the random subspace T and true subspace S are completely disjunct. Thus $T \subseteq I$, i.e., T exclusively consists of attributes that are irrelevant for this true subspace outlier. In these attributes $\vec{x}$ is completely regular, thus, $E[o] \approx default_{out}$.

Combining these observations implies that we observe a discrepancy between the expectation values of the outlier score populations $\mathcal{O}_S^+$ and $\mathcal{O}_S^-$, namely:

$$E\left[\mathcal{O}_S^+\right] \;>\; E\left[\mathcal{O}_S^-\right] \tag{4}$$

The main idea behind our framework is to exploit this discrepancy.

Effects of random sampling: Before we reformulate the problem statement, we analyze how the random sampling of T influences this discrepancy. The general goal is to keep the total number of analyzed subspaces $|\mathcal{P}|$ low to ensure a feasible processing, i.e., $|\mathcal{P}| \ll 2^{\mathcal{A}}$. This means that in practice we have to deal with the limited size of the populations $\mathcal{O}_S^+$ and $\mathcal{O}_S^-$. It is reflected in the statistical uncertainty when comparing $\mathcal{O}_S^+$ and $\mathcal{O}_S^-$ as in Eq. 4. This statistical uncertainty is influenced by the dimensionality $|T|$ of the subspaces $T \in \mathcal{P}$. We have to consider the effects of both high and low dimensional T:

Low $|T|$: Considering the dimensionality dependence of the outlierness profile (cf. Fig. 2), it is obvious that the observed outlierness difference becomes statistically more significant when the subspace T is more similar to S, i.e., when the superset T contains only a small number of additional irrelevant attributes. In Fig. 2, this corresponds to subspaces with a dimensionality close to the outlierness peak. This means that we can maximize the discrepancy in Eq. 4 by reducing the dimensionality of the subspaces in $\mathcal{P}$ to a dimensionality that is only slightly larger than $|S|$.

High $|T|$: On the other hand, we have to consider the underlying combinatorial problem: What is the probability that a random subspace T is a superset of S? Since the subspaces are drawn independently, we can use the hypergeometric distribution to quantify the probability that a space $T \in \mathcal{P}$ is a superset of subspace S. For a database consisting of D attributes, we obtain the ***coverage probability***:

$$P(T \supseteq S) \;=\; \frac{\binom{D-|S|}{|T|-|S|}}{\binom{D}{|T|}}$$

Intuitively, the coverage probability increases if either $|T|$ is large (large covering subspace) or $|S|$ is small (small subspace to cover). For instance, in a database with $D = 100$ attributes and $|T| = 25$ the coverage probability is 6.06% for a two-dimensional subspace and 0.07% for a five-dimensional one. Increasing the size of the sampled subspaces to $|T| = 75$ increases these probabilities to 56.1% and 22.9% respectively. As we can see, if the subspaces in $\mathcal{P}$ are low-dimensional, it becomes more and more likely that $\mathcal{P}$ does not contain any supersets of S. For a limited subspace pool sample $\mathcal{P}$, the superset samples $\mathcal{P}_S^+$ and $\mathcal{O}_S^+$ become very small or even empty. This means that the comparison $\mathcal{O}_S^+$ and $\mathcal{O}_S^-$ is affected by a high statistical uncertainty. Thus, we require high dimensional subspaces T to ensure that the superset populations $\mathcal{P}_S^+$ and $\mathcal{O}_S^+$ are large enough to allow a statistical inference with a high significance level.

Problem Statement: To finally transform the problem of searching for relevant subspaces into a new formulation of the problem statement, we reverse the interpretation of Eq. 4 in the following. So far, we have assumed a given true subspace S and analyzed its influence on $\mathcal{P}$ and $\mathcal{O}$. We now turn to the question of searching for an S' given a subspace pool $\mathcal{P}$ and outlier scores $\mathcal{O}$. We have found that for a true subspace outlier the corresponding true subspace S causes a partition of subspaces and outlier scores. For this partition we observe the discrepancy of $E\left[\mathcal{O}_S^+\right]$ and $E\left[\mathcal{O}_S^-\right]$. The reversal yields our problem statement: *Given a subspace pool $\mathcal{P}$ and outlier scores $\mathcal{O}$, which refinement S' causes a partitioning that maximizes the discrepancy of the outlier score populations $\mathcal{O}_{S'}^+$ and $\mathcal{O}_{S'}^-$?* For the given object, this S' is the best possible approximation of the underlying true subspace S given the limited sample size of $\mathcal{P}$ and $\mathcal{O}$. For the construction of our adaptive framework, we consider this to be a stand-alone problem and only require a subspace refinement function of the form:

$$\texttt{Refine}\left(\mathcal{P},\ \mathcal{O},\ d'\right) \;\rightarrow\; S'$$

This function takes a subspace pool $\mathcal{P}$ and outlierness scores $\mathcal{O}$ of the considered object as input. The third parameter d' determines the dimensionality of the output candidate, i.e., $|S'| = d'$. The output S' is the refined subspace candidate. Formally, this refined candidate is the subspace maximizing the discrepancy, i.e.:

$$\arg\max_{S'} \left(E\left[\mathcal{O}_{S'}^+\right] - E\left[\mathcal{O}_{S'}^-\right]\right)$$

Intuitively, this S' is the best possible d'-dimensional subspace that lets the given object appear anomalous. In other words, we can use `Refine` to get the best lower dimensional attribute explanation why the considered object is an outlier for the given outlier model. The `Refine` function is the key component of our adaptive framework and is used to refine the subspaces adaptively to the outlier score of an individual object. We postpone the discussion of an instantiation of the `Refine` function to Section 4.3 and continue with the overview of our framework in the following.

4.2 Adaptive Subspace Search Framework

At a glance, the RefOut framework consists of three steps: (1) perform outlier mining on the subspaces of an *initial subspace pool* $\mathcal{P}_1$ consisting of random subspaces; (2) refine $\mathcal{P}_1$ resulting in a *refined subspace pool* $\mathcal{P}_2$ that contains subspaces tailored to the given outlier model; (3) perform outlier mining on $\mathcal{P}_2$ to obtain the final output. The first step of the framework can be considered a modified version of the random feature bagging approach proposed in [20]. However, our approach goes beyond this random guessing by performing an adaptive refinement in the second step.

Step 1: The objective of the first step is to collect as much information about objects and subspaces as possible. We randomly draw subspaces of dimensionality d_1 without replacement and add them to $\mathcal{P}_1$ until $|\mathcal{P}_1|$ reaches a threshold

psize. Note that this allows RefOut to perform an exhaustive search on dimensionality level d_1 for very low dimensional databases or large *psize*, but in general $\binom{D}{d_1} \gg psize$. The dimensionality parameter d_1 controls the trade-off between a good subspace coverage probability (large d_1) or a less severe curse of dimensionality (low d_1). The framework then applies the given traditional outlier model to all subspaces $T \in \mathcal{P}_1$. To ensure the desired property of comparable outlier scores amongst different subspaces, we apply the normalization (Eqs. 1-3) to the outlierness distribution in every subspace. The framework stores these normalized outlier scores for every object in every subspace.

Step 2: The goal of the second step is to exploit the information collected in Step 1 by refining the subspaces adaptively to the outlier scores resulting in the refined subspace pool $\mathcal{P}_2$. Note that the subspace refinement operates per object, i.e., every object has an individually refined subspace. In principle it would be possible to produce a refined subspace for every object in the database, resulting in $|\mathcal{P}_2| = N$. However, if an object does not show anomalous behavior in any of the subspace projections of $\mathcal{P}_1$, it is very likely that this object simply is regular. Thus, to speed up the processing, the framework excludes these inliers for subspace refinement. Instead of processing all objects, the framework ranks all objects according to their maximum outlier score over all subspaces in $\mathcal{P}_1$. A parameter *opct* controls the number of objects (expressed as ratio of the database size) that are considered for subspace refinement, i.e., we consider the top $\lfloor opct \cdot N \rfloor$ objects from this ranking. Since each subspace refinement adds one subspace to the refined pool, this also determines the size $|\mathcal{P}_2|$. The target dimensionality of the subspace refinement is given by parameter d_2, i.e., $|T| = d_2 \quad \forall T \in \mathcal{P}_2$.

Step 3: The third step applies the outlier model again – this time to the refined pool $\mathcal{P}_2$. As in Step 1, we normalize the outlier scores of each subspace to ensure comparability. The final outlier score of an object is the maximal normalized outlier score observed over all subspaces in $|\mathcal{P}_2|$. Algorithm 1 summarizes the steps of the RefOut framework.

Algorithm 1 Adaptive Subspace Search

Input: DB, outlier model *score*(.), d_1, d_2, *psize*, *opct*
Output: score and best subspace description for each object
  $\mathcal{P}_1$ = random subspaces of dimensionality d_1
  Apply *score*(.) to all $T \in \mathcal{P}_1$ and normalize outlier scores
  Rank objects according to maximal outlier score
  for $\vec{x} \in \lfloor opct \cdot N \rfloor$ top ranked objects **do**
    Extract $\mathcal{O}$ for individual object $\vec{x}$
    $S' =$ `Refine`$(\mathcal{P}_1, \mathcal{O}, d_2)$
    Insert S' in $\mathcal{P}_2$
  end for
  Apply *score*(.) to all $T \in \mathcal{P}_2$ and normalize outlier scores
  Output maximum score and subspace for each object

To analyze the complexity of this algorithm, we look at the search space processed. A naive algorithm would check all $2^{\mathcal{A}}$ subspaces, which clearly does not scale. In contrast, we only look at a limited set of subspaces. The search space is limited by the parameters *psize* and *opct*. Furthermore, the subspace candidate refinement requires only a small number of subspaces considered in the pool. The total number of subspaces processed is $(psize + \lfloor opct \cdot N \rfloor)$. Thus, the complexity of the framework itself is $\mathcal{O}(N)$. In terms of the underlying outlier model to check these subspaces, we depend on the complexity of the detection algorithm, which range from $\mathcal{O}(D \cdot N)$ for efficient distance-based [11], $\mathcal{O}(D \cdot N^2)$ for density-based methods [7], up to $\mathcal{O}(D \cdot N^3)$ for the basic version of angle-based methods [19].

4.3 Instantiation of the Refinement Function

The goal of the refinement function `Refine` is to obtain the d'-dimensional subspace S' that maximizes the discrepancy of the populations $\mathcal{O}^+_{S'}$ and $\mathcal{O}^-_{S'}$. The input of `Refine` is the set of subspaces $\mathcal{P}$ and the corresponding outlier scores $\mathcal{O}$ of an individual object $\vec{x}$. To simplify the notation we treat both input sets $\mathcal{P}$ and $\mathcal{O}$ as sequences with an arbitrary but fixed order. Since there is an outlierness value for every subspace $T \in \mathcal{P}$, we define the order $\mathcal{P} \equiv (T_1, T_2, \ldots, T_M)$ and $\mathcal{O} \equiv (o_1, o_2, \ldots, o_M)$ such that $o_i = score(\vec{x}_{T_i})$. We will use the notation (T_i, o_i) to refer to a pair of subspace and corresponding outlier score.

To illustrate the problem to solve, we introduce a running example in Figure 4. The table shows the measured outlier scores of an outlier with true subspace $S = \{1, 2, 3, 4\}$ evaluated in random subspaces of dimensionality 9 within a database of dimensionality 12. A green box indicates that an attribute is included in the random subspace. To ease presentation, we have ordered the (T_i, o_i) tuples according to the outlier score of the object in the respective subspaces. If we partition the rows according to $T \supset S$ vs $T \not\supset S$, we obtain the rows with the ranks 1, 2, 3, 4, and 7 as population $\mathcal{P}^+_S$. Considering the corresponding outlier score populations $\mathcal{O}^+_S$ and $\mathcal{O}^-_S$ clearly shows that $\mathcal{O}^+_S$ is stochastically greater than $\mathcal{O}^-_S$. Ideally, for any $d' \geq 4$ the goal of the `Refine` function is to detect this discrepancy and return a refined subspace $S' \supseteq S$.

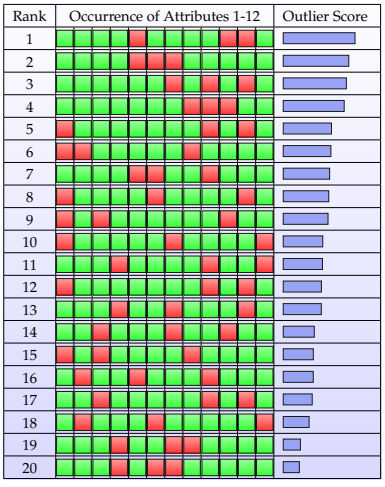

Figure 4: Score discrepancy for $S = \{1, 2, 3, 4\}$

In the following we point to the three major challenges of the refinement problem and explain how we deal with them in our proposed solution.

Uncertainty of populations: This challenge refers to the general problem of comparing populations. For instance, the example demonstrates that the two populations are not strictly separable in general due to statistical fluctuations: We observe that the subspace on rank 7, which is a superset of the true subspace, is ranked below two irrelevant subspaces that coincidentally show a high outlierness for the object. Hence, any solution of the refinement problem must

handle uncertainty in outlier score distributions. Another issue is that for a high dimensional S', the partition may yield a very small sample $\mathcal{P}^+_{S'}$ due to the low coverage probability of high dimensional subspaces. In this case the size of the outlier score populations becomes unbalanced, i.e., $\mathcal{O}^+_{S'}$ is much smaller than $\mathcal{O}^-_{S'}$. For instance, if we consider an S' that corresponds exactly to the top ranked subspace in the example, the statistical significance of comparing $\mathcal{O}^+_{S'}$ and $\mathcal{O}^-_{S'}$ is low since $|\mathcal{O}^+_{S'}| = 1$. Therefore, we propose to rely on statistical tests that are designed for comparing populations and properly handle uncertainty. To quantify the separation power for a given candidate C, our approach requires an instantiation of the following function:

$$\begin{aligned} \mathit{discrepancy}\left(\mathcal{O}^+_C, \mathcal{O}^-_C\right) \;\equiv\; & \text{p-value of a statistical test} \\ & \text{that is sensitive to} \\ & E\left[\mathcal{O}^+_C\right] \;>\; E\left[\mathcal{O}^-_C\right] \end{aligned}$$

By using the p-value we leave the question of the statistical significance to the underlying test: In case of a very small population $\mathcal{O}^+_C$, any reasonable test will report lower p-values, since it is not possible to reject the null-hypothesis of identical populations with a high certainty. There are many possibilities to instantiate the statistical test. For instance, we can use the one-sided versions of the Mann-Whitney-Wilcoxon test or the Student's t-test. We evaluated several instantiations in our experiments. Although we observed only minor differences, we obtained the overall best results with Welch's t-test (a Student's t-test without assuming equal variances of the samples). The reason could be that a t-test is more sensitive to outliers compared to the Mann-Whitney-Wilcoxon test, which only considers the ranks of the populations. While the t-test's sensitivity to outliers is an issue in other domains, it actually is useful in our case: For a high dimensional true subspace S the coverage probability is low. Thus, we might only have a few matching subspaces in the subspace pool. Fortunately, the t-test captures this discrepancy well compared to a rank test. According to our experiments, this property seems to outweigh the fact that the Gaussian assumption of a t-test does not necessarily apply to the outlier score distributions.

Joint occurrence property: We know from the outlierness profiles that only the joint occurrence of the attributes S causes an increased outlier score of a true subspace outlier. In projections of S, the object falls in regions of regular density. In the given example, we observe that the individual occurrences of attributes $\{1, 2, 3, 4\}$ below Rank 7 are completely random and independent from each other since the complete set is never included in these subspaces. Detecting joint occurrences highlights the set-like property of the problem and its exponential characteristic: An exhaustive search to find the exact d'-dimensional subspace S' that maximizes the discrepancy of $\mathcal{O}^+_{S'}$ and $\mathcal{O}^-_{S'}$ would require to evaluate the *discrepancy* of all possible $\binom{D}{d'}$ partitions. Thus, it is not feasible to search for an exact solution. Instead we propose a heuristic search for a subspace S' that approximately maximizes the discrepancy. We define the quality of a candidate subspace C according to the discrepancy of the corresponding partition:

$$\mathit{quality}(C) = \mathit{discrepancy}\left(\mathcal{O}^+_C, \mathcal{O}^-_C\right)$$

Based on this quality function we perform a beam search of the candidates in a bottom-up processing. A parameter *beamSize* determines the number of candidates that we keep on each dimensionality level. We start with all possible one-dimensional candidates. In each iteration we calculate the quality $\mathit{quality}(C)$ of all candidates C. We rank the candidates depending on their quality and discard all candidates that have low quality, i.e., we only keep the top-*beamSize* ones. These top candidates are used to construct higher dimensional candidates. This construction is similar to constructing higher dimensional candidates in frequent itemset mining [4]: We form a $(d+1)$-dimensional candidate in case our candidate set contains all its d-dimensional projections. If it is not possible to construct a higher dimensional candidate, the processing stops.

To highlight the rationale of such a processing we discuss the question whether there is some kind of monotonicity in the candidate generation. In frequent itemset mining, monotonicity refers to the fact that when the quality criterion of a candidate C (in this case the itemset support) is above a certain threshold, so it is for all subsets of S. In our score discrepancy problem, we are faced with a quality criterion which is more complex than a simple count of items, and monotonicity does not hold. However, we observe that our problem has a property which we would call *per-level-monotonicity*. On a fixed dimensionality level d, we have

$$\mathit{quality}(C_{true}) > \mathit{quality}(C_{rand}) \tag{5}$$

where C_{true} are d-dimensional subsets of S and C_{rand} are random d-dimensional candidates which do not share attributes with S. We can see this by noting that $\mathcal{O}^+_{C_{true}} \supseteq \mathcal{O}^+_S$. Thus, the population $\mathcal{O}^+_{C_{true}}$ contains all increased scores of the true population $\mathcal{O}^+_S$ plus a random sample of $\mathcal{O}^-_S$. When taking expectation values, we still have:

$$E\left[\mathcal{O}^+_{C_{true}}\right] > E\left[\mathcal{O}^-_{C_{true}}\right]$$

For random candidates C_{rand} the expectation values of the samples $\mathcal{O}^+_{C_{rand}}$ and $\mathcal{O}^-_{C_{rand}}$ are the same, and thus, Eq. 5 holds. This per-level-monotonicity ensures that by keeping the top candidates on each level in the beam search, we maximize the likelihood of finding the correct S in each step.

To finally obtain the refined d'-dimensional output subspace, we proceed as follows: During the bottom-up beam search we keep a record of all candidate qualities ever evaluated. We rank all candidates according to their $\mathit{quality}(C)$, i.e., their p-values expressing how well they separate the outlier score populations. To collect exactly d' attributes for the output candidate, we iterate over this list, starting with the top ranked candidates. We add the attributes of the candidates in the ranking to the output candidate S' until $|S'| = d'$. In case of adding a candidate C completely would yield $|S'| > d'$, we rank the attributes $a \in C$ according to their one-dimensional qualities $\mathit{quality}(\{a\})$ and only add the best attributes until $|S'| = d'$.

Limited size of subspace pool: Another challenge is introduced by the limited size of the subspace pool. If this number is low, combinatorial interferences are likely to occur. For instance, the last attribute in Figure 4 is not part of the relevant subspace. But since it was never excluded from the top ranked subspaces, there is no way to detect that it is an irrelevant attribute for the given object. Due to the limited number of combinations, the attribute must be added to the set of relevant attributes as a false positive. In order to completely avoid false positives, it would be necessary to evaluate all $\binom{D}{d}$ possible d-dimensional subspaces

on each level. Clearly this is not feasible. However, we can reduce the issue of false positives by relaxing the general goal of the subspace refinement. After all, any reduction of irrelevant attributes already improves outlier detection. Thus, detecting the true S precisely is unlikely unless we construct a huge subspace pool. Instead, the framework increases outlier detection quality by refining the subspace to a dimensionality level d_2. This allows the refinement step to output an $S' \supseteq S$ which may include some false positive attributes. From the framework's point of view, the main goal is achieved: It has been possible to remove $(d_1 - d_2)$ irrelevant attributes, adaptively on the underlying outlier model, allowing enhanced outlier detecting by scoring an object in its individually best subspace S'.

We conclude this section with a brief summary of our solution: The proposed `Refine` function extracts a refined subspace individually for each object based on the outlier scores according to the underlying outlier model. These properties, per-object processing and adaptiveness, distinguish our approach from existing subspace search techniques [8, 12, 13, 23]. The refined subspace is obtained by maximizing the discrepancy in outlier score distributions. Our algorithm performs a beam search that exploits the per-level-monotonicity. Exploiting this special property of our problem distinguishes our approach from approaches e.g. in subgroup detection [26], where such a property does not hold. Furthermore, we have proposed a construction of the output subspace which allows $S' \supseteq S$, and thus, is tailored to the idea of refining subspaces within the enclosing RefOut framework.

5. EXPERIMENTS

Our experiments focus on the interplay of traditional outlier models with subspace search approaches. From the field of outlier models we chose three representative techniques: (1) Local Outlier Factor (LOF) [7], (2) distance-based outlier detection (DB) [14], and (3) angle-based outlier mining (ABOD) [19]. Our general evaluation scheme is to combine these three models with the following subspace selection schemes: (1) random subspace selection (RS) and (2) the full attribute space (FS) as two baselines; (3) HiCS [13] as representative of subspace search techniques; (4) RefOut. For HiCS and RS we always use the maximum outlier score of all subspaces. To ensure repeatability, we provide details on our experiments online.[1]

Our main focus is to analyze outlier detection quality on real world data. We use the area under the ROC curve (AUC) as quality criterion. To perform scalability experiments and to evaluate all RefOut parameters, we utilize synthetic data. Our synthetic data generator injects true subspace outliers in a database as follows: We partition the attributes of the database of dimensionality D in subspace components of dimensionality d randomly between 2 and 8 with equal probability. To create a structure of regular objects in each subspace component, we draw random values satisfying $x_{s_1} + \ldots + x_{s_d} = 1$. We inject a true subspace outlier by deviating one object slightly from this hyperplane, satisfying that all its lower dimensional projections are in a region of regular density. This special type of true subspace outlier can be detected clearly by all three outlier models in the subspace components.

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

Dataset (size x dim)	Ground Truth	Peaks in Dim				
		1	2	3	4	5
Breast (198 x 31)	ABOD	0	139	40	16	3
	DB	58	81	44	15	0
	LOF	36	67	52	29	14
Breast Diagnostic (569 x 30)	ABOD	0	284	187	98	-
	DB	101	268	155	45	-
	LOF	94	177	177	121	-
Electricity Meter (1205 x 23)	ABOD	6	217	405	577	-
	DB	99	537	393	176	-
	LOF	197	374	413	221	-

Table 1: Datasets and dimensionality of peaks

5.1 Adaptiveness on Real World Data

As already illustrated in our toy example in the introduction, it is clear that a LOF outlier is not necessarily an ABOD outlier. Since the true subspace outliers are individual to each model, it would be desirable to have a ground truth of true subspace outliers of each type. To this end, we introduce a novel evaluation approach for detection quality of true subspace outliers in dependence on the outlier model. We propose to perform an *exhaustive search* to obtain a ground truth of true subspace outliers for each model. That is, we scan all subspaces of a dataset exhaustively with each model up to an upper dimensionality level. This is obviously a very time-consuming operation. Therefore, we have to focus on datasets of moderate size and dimensionality to reach a reasonable upper dimensionality level. We chose the datasets Breast, Breast Diagnostic [10] and a larger Electricity Meter dataset from a collaboration partner. Note that we had to drop two discrete attributes from the Breast dataset to ensure a well defined local outlier factor. We further normalized all attributes to a unit interval. We scanned up to a dimensionality of 4 for Breast Diagnostics (31,930 subspaces for each model) and Electricity Meter (5,488 subspaces), and up to level 5 for Breast (206,367 subspaces). The overall scanning took several days, mainly spent on running ABOD (using the FastABOD version [19]).

Since in Sec. 3 we defined the target function to quantify true subspace outliers to be the height of the peak, we store the maximal peak for each object and the corresponding subspace during our exhaustive scan. A first insight is that the three models show very different distributions regarding the dimensionality in which each object showed its maximal subspace outlierness. These results are given in Table 1. For instance, we can see that for Breast and Breast Diagnostic LOF tends to see more high dimensional peaks, while for Electricity Meter ABOD detects more high dimensional peaks. Note that for ABOD the outlierness rarely peaks in 1-dimensional subspaces, since the ABOD score degenerates to a (still meaningful) variance over reciprocal distance products in one dimension.

For the following experiments we rank the peaks (for each model and dataset) and extract three different true subspace outlier ground truths for each model corresponding to the top 2%, 5%, and 10% of the peaks. This allows us to investigate interesting cross evaluations and analyze questions like how well does LOF detect ABOD outliers, or which one of the true subspace models is the hardest to detect in the full space? To this end, we evaluate all 12 combinations of {FS (full-space), RS (random-subspaces), HiCS, RefOut} $\times$ {ABOD, DB, LOF} on all ground truths. The average AUC values of these experiments are shown in

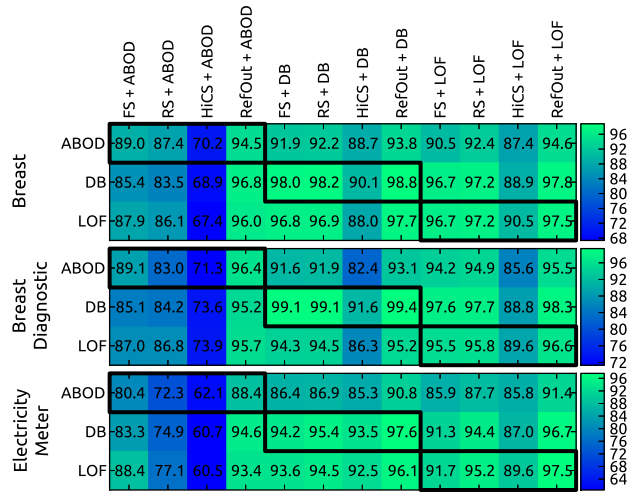

Figure 5: True subspace outlier detection quality (AUC) on real world data

Fig. 5. Each row corresponds to a certain ground truth model ABOD/DB/LOF. We highlight the blocks where a subspace approach uses the same outlier model as the ground truth, and intuitively we expect the best results in this case. We can see that this for instance is strongly pronounced for Breast Diagnostic with the DB model. On the other hand, we were surprised to find that the ABOD ground truth is sometimes better detected using DB/LOF instead of ABOD itself as detection model.

Regarding the adaptiveness of the subspace search models, we can see that the static selection scheme of HiCS does not perform well in general, especially in combination with ABOD. Using random subspaces shows better overall adaptation simply by making no assumption for the selection at all. In most cases RS improves over a full-space detection, but not when combined with ABOD. Regarding RefOut, we can see that its adaptive design clearly improves the subspace selection for all models. We observe the most pronounced improvement over the other subspace techniques in combinations with ABOD. The systematic quality improvement of RefOut comes along with a slightly increased runtime: The average runtimes over all models and datasets were: 41.6 sec for RS, 49.0 sec for HiCS, and 76.2 sec for RefOut, which is still several orders of magnitudes below the runtime for exhaustive searching and is worth to be invested when looking at the improved detection and description of individual outliers.

5.2 Scalability with Dimensionality

To analyze the dependence of the detection quality with the database dimensionality we performed experiments on different dimensionality levels. We generated 5 random datasets on each dimensionality level 25, 50, 75, and 100 with subspace outliers of a random dimensionality up to 8. For this experiment we focus on a single outlier model to keep the number of results manageable. We chose the LOF outlier model due to its high popularity. We kept the LOF parameter $MinPts = 10$ constant for all approaches. For the random subspace detection we chose the same dimensionality level as the dimensionality of the initial pool of RefOut (75% of D) to highlight the improvement due to subspace refinement. We keep the total number of evaluated subspaces equal for RS, HiCS, and RefOut. Fig. 6 shows the results. Regarding quality, we can see that even the random subspace approach consistently outperforms a fullspace subspace detection. Regarding HiCS we can see that it can improve over random subspaces on average. But we also see the effect of its non-adaptiveness: Sometimes the subspaces detected by HiCS match quite well (on the 50 dimensional level); other times HiCS outputs subspaces that are of no use to the outlier model (on $D = 75$). For RefOut we observe a very good scalability with respect to the dimensionality: The subspace selection consistently outperforms the other subspace approaches. The price for the increased quality is a slightly increased runtime. However, we can see that the increase over the runtime baseline defined by RS is rather low: This means that the majority of the runtime is spent on applying the outlier model itself and not on the subspace refinement framework. Overall RefOut shows a linear scalability w.r.t. the number of dimensions, making it capable of handling high dimensional databases.

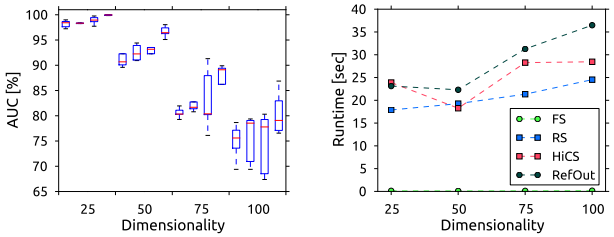

Figure 6: Scalability w.r.t. increasing dimensionality on synthetic data (from left to right in each group: FS, RS, HiCS, RefOut)

5.3 Parameter Evaluation

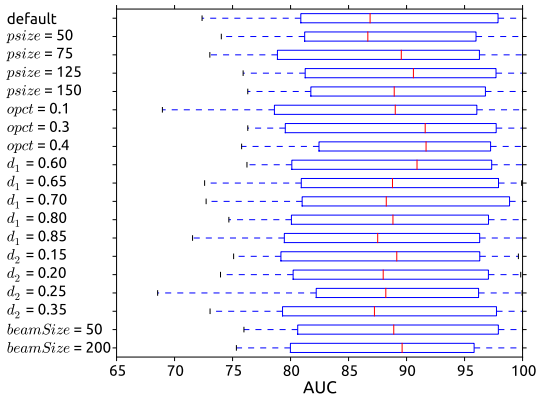

Figure 7: Parameter evaluation

We performed a thorough analysis of all parameters in RefOut, again based on the LOF model. We evaluated each parameter configuration on the pool of 20 datasets for Sec. 5.2. This means that the dataset pool contains both difficult and more easier datasets. In our opinion this is important to ensure that we do not analyze the influence of a parameter for a single database dimensionality. In order not to use absolute values for d_1 and d_2, we set these parameters as percentage of D. Our default parameters were $psize$=100, $opct$=20%, d_1=75%, d_2=30%, and a $beamSize$=100. Starting from this configuration we performed a sensitivity analysis by varying each parameter individually. The results are shown in Fig. 7. We can see that in general the parameters

are robust and slight variations of a parameter do not harm the results significantly. Note that the main fluctuations in the results are caused by the broad spectrum in difficulty of the datasets. As expected, increasing the pool size has a positive influence on the results, although we did not observe further improvements above a pool size of 125. The *opct* parameter that controls how many objects are considered for subspace refinement is also straightforward to set up: Higher values produce better results since the detection quality of the high dimensional subspace scan is less relevant. Our primary choice of 75% for d_1 was motivated by the idea that we wanted both good subspace coverage while keeping the number of irrelevant attributes low. The results show that this choice was still a bit too high: Checking subspaces of a dimensionality of 60% gave slightly better results. This indicates that RefOut works well with a low subspace coverage; the influence of irrelevant attributes is the bigger issue. We did not observe a significant influence of the *beamSize* in our bottom-up subspace refinement on the results, which shows that even low values in the beam search can find reasonably good refinement candidates.

6. CONCLUSIONS AND FUTURE WORK

In this work, we present a flexible and adaptive subspace search technique for outlier mining. It refines a pool of random subspaces by exploiting the score discrepancy in different subspaces. Based on the statistical comparison of outlier scores, we achieve an adaptive search tailored to the underlying outlier model. This allows us to inherit the properties (quality, performance, etc) of various well-established outlier definitions for the subspace search. This results in an improved outlier detection but also in individual outlier descriptions for each object.

Regarding future work, we aim at utilizing adaptive subspace search for outlier ensembles. So far we detect subspaces for each outlier according to a single model. Combining multiple outlier models is a promising extension of RefOut in order to find outliers that deviate w.r.t. different outlier models in different subspaces. This has not been addressed so far, and hence, RefOut might impact future development of outlier ensembles [1].

Acknowledgments

This work is supported by the German Research Foundation (DFG) within GRK 1194, by the Young Investigator Group program of KIT as part of the German Excellence Initiative, and by a Post-Doctoral Fellowship of the Research Foundation – Flanders (FWO).

7. REFERENCES

[1] C. C. Aggarwal. Outlier ensembles: Position paper. *SIGKDD Explorations*, 14(2):49–58, 2012.

[2] C. C. Aggarwal. *Outlier Analysis*. Springer, 2013.

[3] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. In *SIGMOD*, 2001.

[4] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *VLDB*, pages 487–499, 1994.

[5] F. Angiulli, F. Fassetti, and L. Palopoli. Detecting outlying properties of exceptional objects. *ACM Trans. Database Syst.*, 34(1):1–62, 2009.

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

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

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

[9] P. Filzmoser, R. Maronna, and M. Werner. Outlier identification in high dimensions. *Comp. Stat. Data Anal.*, 52(3):1694–1711, 2008.

[10] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

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

[12] K. Kailing, H.-P. Kriegel, P. Kröger, and S. Wanka. Ranking interesting subspaces for clustering high dimensional data. In *PKDD*, pages 241–252, 2003.

[13] F. Keller, E. Müller, and K. Böhm. HiCS: High contrast subspaces for density-based outlier ranking. In *ICDE*, 2012.

[14] E. Knorr and R. Ng. Algorithms for Mining Distance-Based Outliers in Large Datasets. In *VLDB*, pages 392–403, 1998.

[15] E. M. Knorr and R. T. Ng. Finding intensional knowledge of distance-based outliers. In *VLDB*, pages 211–222, 1999.

[16] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Interpreting and unifying outlier scores. In *SDM*, pages 13–24, 2011.

[17] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Outlier detection in arbitrarily oriented subspaces. In *ICDM*, pages 379–388, 2012.

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

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

[20] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *KDD*, pages 157–166, 2005.

[21] E. Loekito and J. Bailey. Mining influential attributes that capture class and group contrast behaviour. In *CIKM*, pages 971–980, 2008.

[22] E. Müller, M. Schiffer, and T. Seidl. Statistical selection of relevant subspace projections for outlier ranking. In *ICDE*, pages 434–445, 2011.

[23] H. V. Nguyen, E. Müller, J. Vreeken, F. Keller, and K. Böhm. CMI: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection. In *SDM*, pages 198–206, 2013.

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

[25] Y. Wang, S. Parthasarathy, and S. Tatikonda. Locality sensitive outlier detection: A ranking driven approach. In *ICDE*, pages 410–421, 2011.

[26] S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *PKDD*, pages 78–87, 1997.