Local Context Selection for Outlier Ranking in Graphs with Multiple Numeric Node Attributes

Patricia Iglesias Sánchez[○] Emmanuel Müller[○•] Oretta Irmler[○] Klemens Böhm[○]

[○]Karlsruhe Institute of Technology (KIT), Germany
{patricia.iglesias, emmanuel.mueller, klemens.boehm}@kit.edu
oretta.irmler@student.kit.edu

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

ABSTRACT

Outlier ranking aims at the distinction between exceptional outliers and regular objects by measuring deviation of individual objects. In graphs with multiple numeric attributes, not all the attributes are relevant or show dependencies with the graph structure. Considering both graph structure and all given attributes, one cannot measure a clear deviation of objects. This is because the existence of irrelevant attributes clearly hinders the detection of outliers. Thus, one has to select local outlier contexts including only those attributes showing a high contrast between regular and deviating objects. It is an open challenge to detect meaningful local contexts for each node in attributed graphs.

In this work, we propose a novel local outlier ranking model for graphs with multiple numeric node attributes. For each object, our technique determines its subgraph and its statistically relevant subset of attributes locally. This context selection enables a high contrast between an outlier and the regular objects. Out of this context, we compute the outlierness score by incorporating both the attribute value deviation and the graph structure. In our evaluation on real and synthetic data, we show that our approach is able to detect contextual outliers that are missed by other outlier models.

1. INTRODUCTION

Outlier mining is an important task in the field of data management and knowledge discovery. It identifies unexpected, erroneous, rare, and suspicious data. Outlier ranking algorithms sort the objects according to their degree of deviation, instead of coming only to a binary decision for each object. This ranking eases a user-driven exploration of outliers, by looking at the most deviating objects first. In the past, outlier mining techniques have focused on vector data or graph data separately [1]. However, more and more applications such as network intrusion, rare protein interactions, financial fraud, or data cleaning demand outlier analysis on combinations of both. They store relationships between objects represented as a *graph* and additional *attributes* for each node, and mine outliers in this combined data space.

In particular, we consider electronic platforms as exemplary application of outlier mining on attributed graphs. Electronic marketplaces try to detect and delete fraudulent product placements since their reputation is highly affected by such fraud. Fake products, overpriced products, or manipulated reviews are examples for outliers that have to be detected. Such electronic platforms provide a large number of descriptive *attributes* for each product (e.g., prices of all sellers, ratings, and product reviews) and the product relations stored in the *graph* of frequently co-purchased products. All of this publicly available information can provide more information for the detection of outliers. However, with more and more information (attributes, nodes, edges) becoming available, not all of it is relevant for data analysis. For instance, an object may be an outlier only w.r.t. a *selection of the attributes* and a *local graph neighborhood*. We call this the *context of an outlier*, in line with publications on contextual outliers and community outliers [14, 30].

In Figure 1 we have illustrated a compact version of this problem setting on an electronic marketplace with both graph and attribute information. *Product 8* is an outlier for the following reason: It has an exceptionally high number of *Reviews*, in contrast to all of its co-purchased *Products 6, 7, 9, 10, and 11*. Although high values in this attribute are normal over the entire database, it is exceptional for this specific context (i.e., set of co-purchased products). Furthermore, *Product 8* belongs to a global graph partition described by products with similar prices (e.g., *Books* community). However, only the local context selection of *Product 8* (subgraph: {6, 7, 9, 10, 11}, {*Reviews, Price*}) in both graph and attributes reveals the local deviation of this outlier. With this work, we focus on the selection of such local contexts for each node in order to detect contextual outliers.

Traditional contextual outlier mining [30, 7] only consider the numeric attribute space neglecting the graph structure. On the other hand, current techniques [21, 15, 23, 29] combining both graph structure and multiple node attributes are not able to do an individual selection of the graph neighborhood and its relevant attributes for each node. Thus, they are not able to provide local contexts for each node in the database in order to compute accurately the outlierness of an object w.r.t. its own neighborhood. In the search for local contexts, one open challenge is the increasing number of attributes in today's applications. Not all the attributes show dependencies with the graph structure and they have

SSDBM '14 June 30 - July 02 2014, Aalborg, Denmark

ACM 978-1-4503-2722-0/14/06...$15.00.
http://dx.doi.org/10.1145/2618243.2618266.

product	sales	reviews	price
1	262	76	25
2	25	30	30
3	155	47	**150**
4	69	105	20
5	80	8	35
6	182	7	15
7	22	5	8
8	234	28	12
9	102	8	5
10	248	6	13
11	10	4	10
...	...	...	...

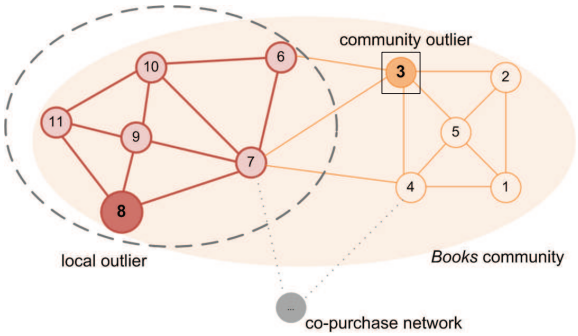


Figure 1: Toy Example: local contextual outlier

almost random values for the residual attributes [29]. In particular, only some attributes are relevant for a certain graph cluster [21, 15, 4]. A core problem is that, even if one has selected a specific graph neighborhood, some irrelevant attributes will scatter the full attribute space [6], and outlier detection is hindered [23, 29]. The outlierness measure is a further challenge, as both the graph structure as well as numeric deviation in the attribute space have to be considered. The definition of a scoring function poses challenges regarding the unification of these two properties.

We propose *ConOut*, the first statistical attribute selection, which enables the detection of contextual outliers in graphs with multiple node attributes. Our context selection allows a good distinction of outliers w.r.t. both the selected attributes and the local graph neighborhood. With this model we select relevant attributes that show similar attribute values for the selected graph neighborhood. Thus, we can discern outliers from regular objects even in the presence of many irrelevant attributes. This context selection allows the detection of local outliers that would not be detectable considering the entire graph or a global partition. Finally, we measure outlierness of each object by unifying structural and numeric information. With *ConOut*, we make the following contributions:

- Local context definition for outlier ranking
- A statistical context selection locally for each node combining both attribute values and graph structure
- Outlierness scoring unifying graph and attribute properties

In our experimental evaluation, we compare against several baselines [38, 7, 2] on either graph or numeric data and recent competitors using both graph and numeric data [14, 23, 29]. Finally, the results highlight the benefit of context selection in graphs with multiple numeric node attributes.

2. RELATED WORK

We discuss outlier mining in (1) vector data, (2) graphs, (3) combinations of both, and open challenges not yet addressed in literature:

Outlier Mining in Vector Data: For several decades, vector data has been studied [9], with different paradigms such as supervised [36], deviation-based [28], distance-based [18] or density-based methods [7]. In this work, we focus on density-based outlier ranking, which proposes scores to measure the deviation of each object w.r.t. the object's local neighborhood. More recent developments focus on subspaces [2, 19, 24, 17]. They rank objects based on any possible attribute combination. However, all of these approaches consider vector data only and do not address relations between objects given in graph databases.

Anomalous Nodes in Graph Structures: This work focuses on outlier nodes, and does not consider anomalous edges, irregular subgraphs, and other suspicious structural anomalies [26, 8, 13]. Recent research in the detection of anomalous nodes can be categorized by the underlying graph types and by the different anomaly models. Some approaches are limited to bipartite graphs [33, 35]. Others use the node neighborhood and its power law characteristics [3]. Graph clustering algorithms detect outliers as a byproduct [38, 32]. All these approaches succeed in the detection of outlier nodes based on the graph structure. However, they ignore information at each node such as numeric feature vectors.

Mining Graphs with Node Attributes: An emerging research field considers both graph and vector data. First, a variant of graph clustering that combines node attributes and graph structure has been proposed to obtain better clustering results [39]. However, it does not consider selection of relevant attributes. Although general approaches have been proposed as pre-processing step for the selection of relevant attributes [34, 29], they do not consider a local selection of the attributes w.r.t. the node neighborhood. Some clustering techniques have focused on a local selection of the attributes [21, 15]. In particular, they address the selection of multiple subspaces on the cluster level. In contrast to this, partitioning algorithms locally select a single projection of the attributes for each cluster [4, 16]. Nevertheless, they do not focus on a single and individual projection for each node for outlier mining. Regarding outlier mining, only few approaches consider attributed graphs. Some use semi-supervised learning in order to label nodes before searching for suspicious nodes [10]. Others aim at unsupervised mining. One algorithm [12] searches for irregular subgraphs within a graph with numeric node attributes, but it does not consider *individual* nodes. The work proposed in [14] detects outlier nodes that deviate from communities in attributed graphs (e.g., social networks). However, all attributes are considered for each community without locally excluding the irrelevant ones. To avoid this, an approach for subspace selection has been proposed as a pre-processing step for outlier mining on attributed graphs [29]. It extracts subsets of the attributes that are correlated with the entire graph structure. In contrast, *ConOut* selects the relevant attributes for each node neighborhood locally. Furthermore,

outlier ranking based on subspace clustering techniques has been introduced [22, 23]. The main drawback of all these approaches based on the subspace selection paradigm is their time complexity, as there is an exponential number of possible subspaces. Overall, it remains an open issue to efficiently select a local projection of the relevant attributes w.r.t. the individual graph neighborhood for each node.

3. CONOUT MODEL

Our general idea is to measure locally the outlierness of each object in a projection of the given attributes. Both outlierness measure and projection are determined within the local graph neighborhood of each object. In contrast to other graph mining approaches, we do not consider a global partitioning of nodes. This is because we aim to compute accurate ranking values w.r.t. the local neighborhood of each node. For each node neighborhood, our approach selects carefully only the subset of attributes showing similar attribute values. Hence, each object determines its own local neighborhood in conjunction with its relevant attributes. This local context selection for each node ensures a high contrast in this projection between an outlier and its neighbors, that serves as a basis for computing the deviation. In the following, we describe the problem overview in more detail before we propose our statistical selection of attributes in local graph neighborhoods and our novel outlierness measure.

3.1 Problem Overview

The aim of outlier ranking is to provide a sorting of all objects o given in a database DB. In our case, we model the database DB as an attributed graph formed by its graph structure $G = (V, E)$ and its attribute information A as follows:
(1) Each object o is a graph vertex $o \in V$ and connected by edges $(o, p) \in E$ to other nodes $p \in V \backslash \{o\}$ in the graph structure. We assume edges to be undirected and unweighted.
(2) Each object o is described by a vector $\vec{o} = (x_1, ..., x_d) \in \mathbb{R}^d$ where the attributes are named $A = \{A_1, \ldots, A_d\}$.
Outlier rankings score each object according to the *degree of deviation* measured by a function $score : DB \rightarrow \mathbb{R}$. This score provides a real-valued measure of the objects' outlierness.

Local Context Selection.

Local approaches for outlier ranking have shown to improve the quality w.r.t. global approaches as they are able to compare carefully each object with its own neighborhood. Thus, they are able to detect hidden outliers which cannot be detected if one considers the whole database [7, 3]. However, these traditional local approaches have focused on vector [7] or graph data [3]. Thus, they are not able to detect *community outliers* that appear in combination of the graph structure and the node attributes. For example, *Product 3* shown in Figure 1 is such a *community outlier*. It belongs to a community of related products (e.g., *Books*) with similar price values and it shows highly deviating values in the attribute *price*. Only a context selection combining both the graph structure with node attributes enables the detection of such outliers [14]. However, with more and more attributes describing these nodes in such attributed graphs, not all the attributes have to depend on the underlying graph structure. Hence, they have almost random values for the residual attributes (e.g., attribute *sales*). This effect hinders the clear distinction of outliers from regular objects as all nodes seem to be outliers if one considers all attributes [25, 29]. *Product 3* is only deviating w.r.t. the attribute *price*. It is essential for outlier ranking to consider only these relevant attributes for an accurate measurement of the deviation. In order to avoid this, pre-propressing techniques have been proposed for the selection of the relevant attributes [29]. Nevertheless, to ensure the correlation of the attributes with the entire graph structure is a global perspective of the database which does not allow the local extraction of the relevant attributes for each community. Following our previous example, related *Books* have similar prices if one only considers this community in a co-purchase network, but this attribute may be not relevant for other communities (e.g., *Hardware* products). To achieve this, one can use graph clustering techniques [21, 15, 4] in order to exploit local selections of attributes in each community for outlier ranking [22, 23]. Overall, all these techniques provide a global perspective on the database as they extract the relevant attributes from a global clustering result instead of analyzing the neighborhood of each node. So, these approaches are not able to detect local outliers in graphs with multiple node attributes as they are not able to provide a local context selection for each node. *Product 8* is an example of such a local outlier. It belongs to the global community of *Books* and it also shows a similar price w.r.t. them. However, its own local context consists of more specific products (e.g., *Tolkien's books*) that show not only similar prices but also similar number of reviews. Only such a local context selection allows us to detect this product as a local outlier. It highly deviates in a relevant attribute (e.g., number of *reviews*) of its own neighborhood. We define this as *local context* of an object o which consists of a tuple formed by a selected subgraph and its relevant attribute projection:

DEFINITION 1. ***Local Context***
Given an object o, we define its local context as the tuple $(C(o), R(o))$ consisting of the graph context $C(o) = (V', E')$, $V' \subseteq V$ and $E' \subseteq E$ and its relevant attribute projection $R(o) \subseteq A$.

Please note, that for our problem setting we do not consider isolated nodes. This is because they do not have a neighborhood regarding the graph structure. Thus, the set of relevant attributes based on their local graph neighborhood and their outlierness w.r.t. their local neighborhood cannot be determined. Given Definition 1, two main questions remain: (1) how to define the graph context showing similar graph structure between the nodes and (2) how to model the relevance of an attribute given this graph context. We address these questions in Section 3.2. Based on this careful selection of a local context, the ranking function is able to compute accurately the deviation of each object w.r.t. its neighborhood.

Context based Ranking.

Traditional scoring functions in the vector space [7, 1] are only based on the object attributes $\vec{o}$, while graph methods [3, 8] consider only the graph structure $G = (V, E)$ for the scoring function. In contrast to these traditional rankings, we propose a score that incorporates information of both resources based on a previous local context selection. The vector space provide essential information about the deviation of an object regarding the attribute values. On

the other hand, the graph structure can enrich this with valuable information about the affinity between the objects as observed in several studies [20, 11]. A strong connected subgraph of nodes is an evidence that they share some similarities in contrast to isolated nodes that can be the result of a casual relation. Thus, an object showing high deviation in a selected set of attributes within a highly connected subgraph should be ranked first in the result compared to an object low connected w.r.t. its local neighborhood. For this reason, the score has to integrate the information from the deviation within the relevant attributes w.r.t. the connections in its local context. This score gives way to new challenges, as one has to unify the information from both components defined in Definition 1: the deviation in the relevant attribute values and the connections within the graph context. We give more details on an instantiation of such score in Section 3.3.

3.2 Local Context Selection

In the following, we explain the local context selection of each object o formed by its graph context $C(o)$ and its relevant attribute projection $R(o)$ (cf. Definition 1).

Graph Context.

For each object o we select a subgraph $C(o) \subseteq V$. It represents its local context, which shows high similarity in the graph structure between nodes belonging to this context. Intuitively a context $C(o)$ has the following property:

$$\forall p, q \in C(o): \quad p \text{ is structurally similar to } q$$

As graph similarity, we rely on the shared nearest neighborhood (SNN) [38, 32]. Based on this similarity we define formally the graph context $C(o)$.

DEFINITION 2. ***Graph Context*** $C(v)$
Given two objects $v, p \in DB$ and a threshold $\varepsilon \in [0,1]$, the structural similarity is defined as:

$$sim(v,p) = \frac{|Adj(v) \cap Adj(p)|}{\sqrt{(|Adj(v)|) \cdot (|Adj(p)|)}}$$

where $Adj(v) = \{p \in V \mid \exists (v,p) \in E\} \cup \{v\}$. It forms the basis for the transitive closure of similar nodes in the graph context $C(o)$, as defined by:

$$C(v,\varepsilon) = \{p \in V \mid \begin{array}{l} \exists q_1, \ldots, q_k \in DB, \\ sim(q_i, q_{i+1}) \geq \varepsilon \\ \text{with } v = q_1 \text{ and } p = q_k\} \end{array}$$

Overall, we define the context of an object o as the reflexive transitive closure of adjacent nodes with high similarity. It restricts the object neighborhood by a similarity threshold ε, which controls the structural similarity of the context. This selection of the local neighborhood is only a first step in the context selection and it can be also achieved by other local graph context definitions (e.g., extensions of local PageRank [5]). Outliers show up if one focuses on a context of nodes which share common properties, both in structure and in attribute values. Hence, this selection of the local neighborhood is only a first step in the context selection. Further restrictions are defined by the attribute context.

Relevant Attribute Selection.

In addition to the graph context $C(o)$, we require a subset of the attributes $R(o) \subseteq A$ where the attributes show similar values. For many attributes the values show almost random distribution with high variance. These scattered attributes (i.e., showing high variance) are not relevant for the selected graph context. We propose a statistical test to exclude such irrelevant and scattered attributes for each individual object in the database. The idea is to include only attributes that show significantly lower variance in $C(o)$ than the overall data distribution.

DEFINITION 3. ***Attribute Context*** $R(o)$

$$R(o) = \{A_i \in A \mid \begin{array}{l} A_i \text{ has significantly lower variance} \\ \text{in } C(o) \text{ than the overall database} \end{array} \}$$

As basic properties we have to compute the mean $\mu_i(o)$ and variance $\sigma_i^2(o)$ of a given graph context $C(o)$, as follows:

$$\mu_i(o) = \sum_{p \in C(o)} \frac{p_i}{|C(o)|} \qquad \sigma_i^2(o) = \frac{\sum_{p \in C(o)} (p_i - \mu_i)^2}{|C(o)| - 1}$$

Similarly we compute the overall mean $\overline{\mu_i}$ and variance $\overline{\sigma_i^2}$ for attribute A_i in the entire database. Both the local distribution and the global distribution are compared to each other.

Our test is based on a statistical significance test aiming at reducing the probability that an irrelevant attribute passes into the set of relevant attributes. We test against the null hypothesis that objects are distributed with the same local and global distribution, i.e., $\sigma_i^2(o) = \overline{\sigma_i^2}$. We expect a relevant attribute to show significantly lower variance in a local context $C(o)$ when compared to the entire database. This means that the structural context has selected a subgraph with very similar attribute values in A_i. We exclude scattered attributes that do not fulfill this requirement. Furthermore, by setting a very low significance value $\alpha = 0.05$, we ensure that irrelevant attributes pass the test with a very low probability.

DEFINITION 4. ***Attribute Context Test***
For the global variance $\overline{\sigma_i^2}$ and the local variance $\sigma_i^2(o)$ in context $C(o)$ we define hypotheses H_0 and H_1:

$$\begin{array}{ll} H_0: & C(o) \text{ with similar distribution to } DB, \text{ i.e., } \sigma_i^2(o) = \overline{\sigma_i^2} \\ H_1: & C(o) \text{ with individual distribution, i.e., } \sigma_i^2(o) < \overline{\sigma_i^2} \end{array}$$

ensuring a significance level:

$$P(H_0 \text{ is rejected } | H_0 = TRUE) \leq \alpha$$

Depending on the data characteristics, different statistical tests can be applied for our novel attribute selection in graph contexts. In this work, we examine two different statistical tests and evaluate them in Section 5.

First, we use the F-Test as a statistical tool to analyze two Gaussian distributions by the comparison of their variances [27]. The F-test derives the threshold required for rejecting H_0 out of the degrees of freedom, i.e., the size of the context and the size of the database. As test statistic, this test uses the quotient of the two variances observed. Formally,

$$F = \frac{\overline{\sigma_i^2}}{\sigma_i^2(o)}$$

is the observed test statistic and $F_{k1,k2}$ is the critical value of a F-distribution with the degrees of freedom: $k1 = |DB| - 1$

and $k2 = |C(o)| - 1$. H_0 is rejected when $P(F_{k1,k2} \geq F)$ is under the significance level α. The F-Test ensures that $R(o)$ contains only attributes A_i with low variance in $C(o)$. In particular, we limit the probability of having an attribute with high variance in $R(o)$ by α. Let us illustrate this test with our toy example in Figure 1 and *Product 8* with its local context $C(o) = \{6, 7, 8, 9, 10, 11\}$. Testing attribute *sales* means to check if the local variance is lower than the variance of the entire database (e.g., the entire co-purchase network with size $|DB| = 36$). With $P(F_{35,5} \geq 0.7) = 0.76$, this attribute is clearly above the significance threshold α and is considered irrelevant. In contrast to this, *price* obviously shows low local variance in $C(o)$. In particular, $P(F_{35,5} \geq 5.2) = 0,01$. In general, attributes with *p-values* under the significance level will be selected as relevant attributes.

Second, we also analyze our approach with the two sample Kolmogorov Smirnov test that does not require any underlying assumption of the data distribution [31]. This test does not only consider variations in the variance to determine if two samples significantly differ, but it also considers mean variations. To achieve this, it considers the absolute distance between two empirical distribution functions, i.e., the empirical distribution functions of attribute A_i considering the whole database F_{DB} and the individual context $F_{C(o)}$. The calculated test statistic is defined as the maximal difference:

$$D = \sup_{x_{A_i}} |F_{DB}(x_{A_i}) - F_{C(o)}(x_{A_i})|$$

If the calculated test statistic D is larger than the critical value $K_{|DB|,|C(o)|}$, the null hypothesis is rejected with a significance level α with $P(K \geq D_{DB,C(o)}) < \alpha$.

In general, A_i is only relevant when the H_0 hypothesis is rejected. Without a selection of the attributes by a statistical test, scores are blurred by the high variance of irrelevant attributes. So, it ensures a high contrast between outliers and regular objects. This provides the basic means for the outlierness scores in the following Section. Regarding the use of a statistical test, other tests for the comparison of samples can be found in the literature. Some of them are non-parametric and aim to be more robust w.r.t. the presence of outliers (e.g., Wilcoxon signed-rank test [37]). Additionally, existing tests can also be modified to avoid an impact of the outliers on the test without assuming high homogeneity in the context (e.g., using the median instead of the mean to compute the variance). However, the focus of this work is not to analyze or improve the statistical tests for the selection of the attributes. We have only selected two well-established representatives to evaluate our framework. We do not expect any difficulty when instantiating the statistical test used with any other statistical test possible.

3.3 Context Based Outlier Ranking

As an essential property of our scoring, we measure outlierness locally for each object. We ensure an adaptive scoring in local contexts and aim at the local deviation of each object. So, we follow the well-established paradigm of local outlier ranking [7, 24]. Based on this general idea of local outliers we compare each object with its local neighborhood and measure its outlierness locally in contrast to this set of objects. Furthermore, one intrinsic challenge behind this intuitive outlier notion is that one has to ensure that outlier scores remain comparable. Using one scoring function for different subgraphs and different attribute sets will be biased (e.g., w.r.t. the context size). Hence, we have to normalize the score accordingly for each object. We propose such a normalized and unified score in the following. Before we introduce our novel contextual score to integrate the information of both node attributes and graph structure, we present first the measure to extract the deviation of an object in the vector space and the measure to calculate the edge density of an object w.r.t. its neighborhood.

Attribute-Based Score.

As attribute-based score we consider the deviation of each selected attribute $A_i \in R(o)$. We measure the deviation of an object o w.r.t. the local mean $\mu_i(o)$ in its graph context. We formalize the attribute-based deviation of a node in the following definition.

DEFINITION 5. ***Local Attribute Deviation*** $LAD(o)$
Given an object o and its relevant attributes $R(o)$, we define its LAD as:

$$LAD(o) = \frac{\sqrt{\sum_{A_i \in R(o)} \frac{(o_i - \mu_i(o))^2}{{\sigma_i}^2(o)}}}{|R(o)|}$$

where $\mu_i(o)$ and $\sigma_i(o)$ are the mean and standard deviation of attribute A_i in the graph context $C(o)$.

Regular objects with no deviation in their attribute values are clearly separated from outliers, i.e., a regular object o has a low deviation ($LAD(o) \approx 0$). We apply this definition within the local context of each node and we do not apply it for the entire database. Thus, we assume a normal distribution within the local contexts representing the inliers, and outliers are assumed to deviate from the mean of the distribution. These objects are regular observations and should end up at the bottom of our ranking. In contrast, highly deviating objects that are observed will be scored with high outlierness ($1 < LAD(o) < \infty$). Comparability is achieved by our normalization: It is neither biased by the number of selected attributes $|R(o)|$ nor by the different local densities resulting in highly different variance values ${\sigma_i}^2(o)$.

Graph-Based Score.

Second, we define the structural properties that compare the object connections to the ones of its local graph context. We follow the local adaptation in the attribute-based score and extend this idea to local graph densities.

DEFINITION 6. ***Local Graph Density*** $LGD(o)$
Given an object o and its graph context $C(o)$, we define its LGD as:

$$LGD(o) = \frac{con(o)}{\frac{\sum_{p \in C(o)} con(p)}{|C(o)|}}$$

with the average connectivity $con(p)$ at node p as:

$$con(p) = \frac{1}{|Adj(p)| - 1} \cdot \sum_{(p,q) \in E} sim(p, q)$$

With $con(p)$ we describe the average connectivity to nodes belonging to the same context. It is based on the same notion of SNN as the one used in our graph context definition.

For comparability (i.e., outlier scores in different contexts) we normalize connectivity of each object w.r.t. the connectivity of its neighborhood. For the local node density, we compare the connectivity of o with the average connectivity in its graph context $C(o)$. Low density ($0 < LGD(o) \leq 1$) highlights a node with only low connectivity in comparison to its local graph context. In these cases, o should get lower weights as a contextual outlier and should be ranked lower in comparison to highly connected nodes ($1 < LGD(o) < \infty$). With $LGD(o) = 1$, we have a baseline for the structural connectivity. In such cases, we consider only the attribute deviation.

Contextual Outlier Score.

Finally, we integrate graph-based and attribute-based measures to form a unified scoring function, which aims at contextual outliers combining the information from graph structure and attribute values. Our score aims to consider both attribute and graph properties: A local outlier may have a small attribute deviation from a densely clustered neighborhood, or it may have high deviation from a weakly connected neighborhood. Both cases get a high outlierness score. Overall, our outlier score aims to detect local deviation considering both graph and attribute properties.

DEFINITION 7. ***Contextual Outlier Score***

Given an object o with $|C(o)| \geq 2$ and $|R(o)| > 0$ we define its contextual outlier score as:

$$score(o) = LGD(o) \cdot LAD(o)$$

Please note that the product $LGD(o) \cdot LAD(o)$ achieves better outlier detection than its individual factors $LGD(o)$ and $LAD(o)$. It covers several cases of contextual outliers w.r.t. both structural and attribute information that cannot be detected by the individual measures in one of the two information sources. In addition to this, our contextual outlier score exploits the zero property of the multiplication ensuring that regular objects (e.g., objects $LAD = 0$) appear at the bottom of the ranking. In the following, we discuss some of these contextual outlier properties here, and show an empirical comparison to the individual measures and other aggregation functions such as minimum, maximum, and sum in Section 5.

$LGD > 1$ & $LAD > 1$
Strong structural connections and high deviation of attributes in this graph context is the most prominent case of a contextual outlier. Such an outlier will be scored extremely high. It shows high attribute deviation although the structural similarity gives way to the expectation of very similar attribute values.
$LGD = 1$ & $LAD > 1$
Average connectivity (similar to its local neighborhood) and high attribute deviation are scored with high outlierness as there is a graph context. However, attribute values are highly deviating from the residual objects in the context.
$LGD \approx 0$ & $LAD > 1$
Low structural density is an indicator for a weak graph context and lowers the overall score of the object.
$LGD \approx 0$ & $LAD \approx 0$
Lower attribute deviation and lower structural similarity is the other extreme case. In such cases there is no indication for a contextual outlier at all. These objects will be ranked last.

We also include the special case with those objects being hubs in the graph. These nodes belong to multiple contexts as they do not have high structural similarity to a single graph neighborhood and share different properties with different communities. In these cases, we score based on their adjacent neighbors and all relevant attributes of their neighbors, i.e., $C(o) = Adj(o)$ and $R(o) = \bigcup_{p \in Adj(o)} R(p)$. Hence, scoring is simply the average deviation from the neighboring contexts.

Summarizing the *ConOut* model, we have proposed a local context definition, a statistical selection of relevant attributes, and a scoring function for contextual outliers. Based on this formal model, we will sketch the algorithmic computation in the following section and examine the quality enhancement for outlier detection in Section 5.

4. ALGORITHM

In this section, we describe the *ConOut* algorithm. It computes the outlierness of each node in three steps: (1) compute the local graph context, (2) select its relevant attributes, and (3) compute the local outlierness. Finally, all nodes are sorted by their scores.

As parameter, we require only the threshold ε that states how similar objects have to be in the graph structure. In the first step, nodes v adjacent to o, which satisfy the structural similarity, are inserted into a queue. For each of these nodes, we recursively expand the local context with its adjacent nodes until no further nodes can be added into its context and we mark them as visited in the boolean vector (Lines 4-13). As the structural similarity is symmetric ($\forall v, o \in DB, sim(o, v) = sim(v, o)$), all nodes fulfilling this condition have the same context (Line 8). In the second step, we compare the distribution of attribute values in the local context to the distribution in the entire database. A statistical test for this comparison is applied to each attribute (Lines 14-19). Finally, we compute the outlierness of each object based on its local context and its relevant attributes (Lines 20-26).

Complexity Analysis.

Overall we have to iterate over all objects in our database $|DB| = n$. In the first step, we access the graph by means of an adjacency list for each node. This has a cost proportional to the degree of each node. Thus, the cost is linear with the number of edges m for each iteration (Line 5-13). In the worst case, when the whole graph represents the local context, it is $|V| + |E| = n + m$. In this case, all nodes are marked as visited in the first iteration of the main loop ($context[o] = true$), and the algorithm iterates only over the boolean vector without searching for new contexts. This is a rare case for a complete graph, or for a parametrization that is too permissive (e.g., $\varepsilon = 0$). In the second step, the computational cost is linear with the number of dimensions $d = |A|$ and the number of nodes of the context. Each attribute is tested once for each local context. To compute the outlierness, we iterate over each node of the local context, and the runtime of scoring is constant in each iteration, since we have pre-computed all values required for the scoring function. In the worst case, the local context is the whole graph, and we must compute the ranking for each node. Finally, the nodes are sorted by the score values. Overall, the runtime of *ConOut* depends on the local context selection,

Algorithm 1 *ConOut*

Input: $DB : (V, E)\&A$, and parameter ε
Output: Ranking of all $o \in DB$

```
Initialize boolean vector context for all o ∈ DB: false
for all o ∈ DB where context[o] = false do
  Mark context[o] as true
  insert all {p | (o,p) ∈ E} into queue Q
  while (Q ≠ ∅) do
    if p is similar (cf. Def. 1) then
      Insert p into C(o)
      Mark context[p] as true
      Insert non-visited q with (p,q) ∈ E into Q
    end if
    Label p as visited and remove p from Q
  end while
  for all A_i ∈ A do
    Compare distribution of A_i in C(o)
    with the distribution of A_i in DB
    if A_i relevant (cf. Def. 3) then
      Add A_i to R(o)
    end if
  end for
  for all v ∈ C(o) do
    if (|C(o)| ≥ 2) ∧ (|R(o)| > 0) then
      Compute score based on C(o), R(o) (cf. Def. 7)
    else
      Compute score based on Adj(v), A
    end if
  end for
end for
Sort all o ∈ DB by score(o)
```

the statistical test of relevant attributes and the sorting of the nodes. Thus, the worst case cost is $O(m+d+n \cdot log(n))$.

5. EXPERIMENTS

We evaluate the quality, runtime, parametrization and different scoring functions on synthetic and real world datasets. We compare *ConOut* to several competitors:

1. The clustering algorithm SCAN [38], which considers only the graph structure. It allows the detection of structural outliers.
2. Different paradigms considering only vector data: the full dimensional approach LOF [7] and the subspace outlier approach SOF [2] that analyzes the relevant subspaces in order to exclude irrelevant attributes that hinder outlier detection.
3. As full dimensional approach for attributed graphs, the community outlier mining algorithm CODA [14], which considers all the node attributes and the graph structure.
4. Two related algorithms based on the subspace selection paradigm that combine both resources: (1) outlier ranking on attributed graphs based on subspace cluster analysis *GOutRank* [23] and (2) a global subspace selection algorithm as pre-processing step *ConSub* for mining attributed graphs [29].

The quality of the obtained outlier ranking has been determined by the *area under the ROC curve* (AUC). For each position in the ranking, we compute the ratio of precision/recall and compute the AUC value as commonly used for the evaluation of outlier rankings [1]. We have implemented all algorithms in Java and performed experiments on an Intel CoreDuo running 1,8 GHz and 4 GB memory. To facilitate comparability of our experiments, we provide code, datasets, and parameter settings online on our project website[1].

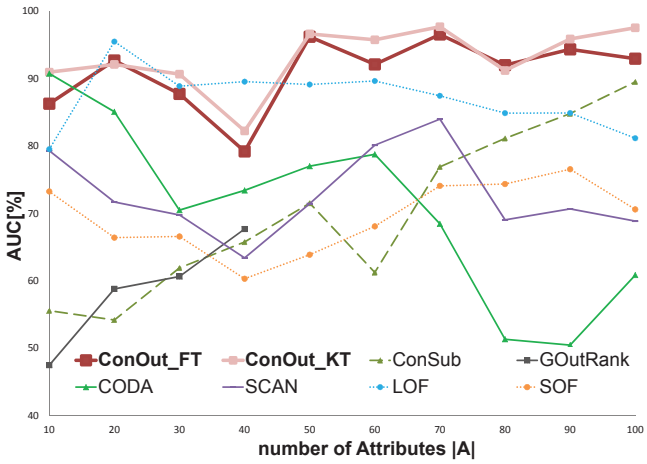


Figure 2: Quality w.r.t. number of attributes

5.1 Synthetic Data

Generation of the Synthetic DataSet.

We have based our generator on the graph generator described in [38]. It allows to generate structural outliers as well as hubs connected to multiple clusters. We have extended this generator with numeric node attributes. We generate graph clusters with intra-cluster connectivity of probability P_{in}, and inter-cluster probability of edges P_{out}. In our setup, P_{in} is higher than P_{out}. For each graph cluster, we randomly select $x \in (1, d]$ relevant attributes and choose their attribute values based on a Gaussian distribution. In contrast to this, all other attributes get values out of a uniform random distribution. The attribute values for hubs and structural outliers are chosen depending on the distributions of their direct neighborhood. In addition to hubs and structural outliers, we insert context outliers that are hard to identify. They are generated by selecting clustered nodes and manipulating a random number of their relevant attribute values. As ground truth for each object, we have marked the outliers generated with a respective label.

Experiment Configuration.

We generate different graphs with an increasing number of attributes. For each dimensionality, we generate three graphs to average over random effects in the generation process. Additionally, we generate one-dimensional datasets varying the number of nodes and edges for the runtime evaluation. On each of these datasets, we configure the different algorithms as follows: For the algorithm *CODA*, we set the exact value of the outlier ratio and the number of clusters since these parameters are known for each dataset generated.

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

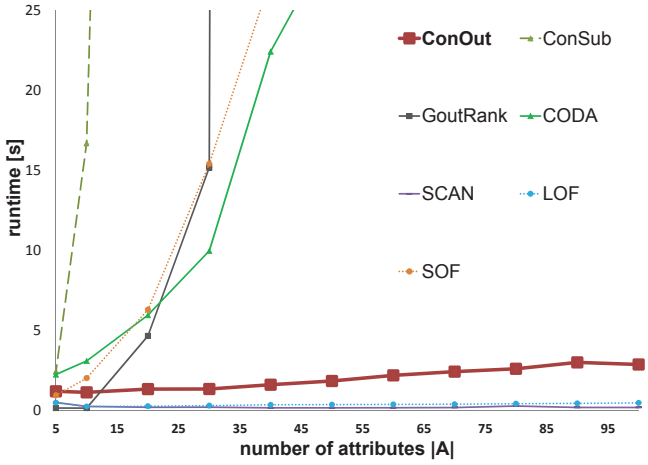
Additionally, we used 10 different initializations for *CODA* and used only the best result. Regarding the unknown parameters for the other algorithms, we try several parameter combinations. Finally, we use the results of the parameter combination showing the best quality results. Detailed information about the exact ranges of each parameter can be found in our website. In particular, *ConOut* achieves the best results with values of ϵ between 0.5 and 0.7.

Quality evaluation.

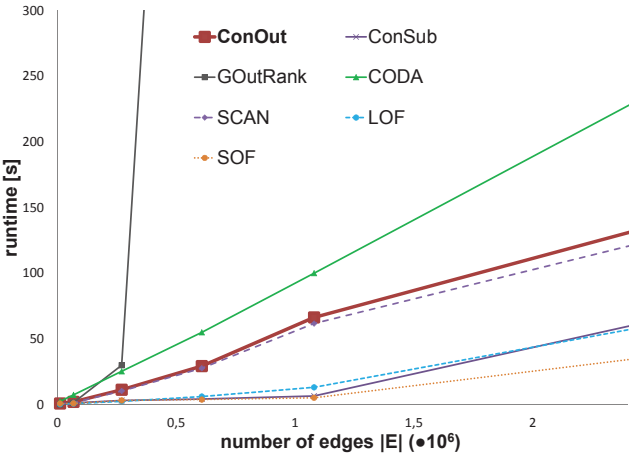
First, we evaluate the outlier detection quality w.r.t. the number of node attributes. We depict average AUC values for all competitors in Figure 2. For each algorithm, we have tried to find optimal parameter settings. In particular, for CODA we have tested 10 different initializations and have used only the best result. In addition, we evaluate two statistical tests for our approach. Experiments show that the selection of relevant attributes using the Kolgomorov-Smirnov test (*ConOut_KT*) achieves better results than the F-Test (*ConOut_FT*). This is because it is more robust by mean variations w.r.t. the global distribution. Not depending on this choice of statistical tests, our approach outperforms all competitors. It is the only algorithm that can detect the context outliers hidden in the graph. Due to our statistical selection of relevant attributes, we achieve high quality even for a large number of attributes. In contrast, traditional competitors tend to miss some hidden outliers as they only consider one data source (graph structure (SCAN) or vector data (LOF, SOF)). A detailed analysis of the detected outliers in Figure 2 reveals that SCAN is performing well on structural outliers having deviating attribute values. Regarding the local approach *LOF*, it neglects the information of the graph structure and it does also not select the relevant attributes for each neighborhood. Thus, its performance decreases with increasing dimensionality. Similar to this, CODA uses all the given attributes and fails because of the irrelevant attributes. Although *ConSub* selects the relevant attributes for the graph structure, this selection is done globally (for the whole graph). Thus, it is not able to select locally the relevant attributes for each neighborhood. Finally, the ranking functions of *GOutRank* heavily depend on the underlying subspace cluster definition and do not consider the local neighborhood of each node. Overall, we have shown that *ConOut* achieves significant quality improvement in the detection of context outliers.

Runtime Evaluation.

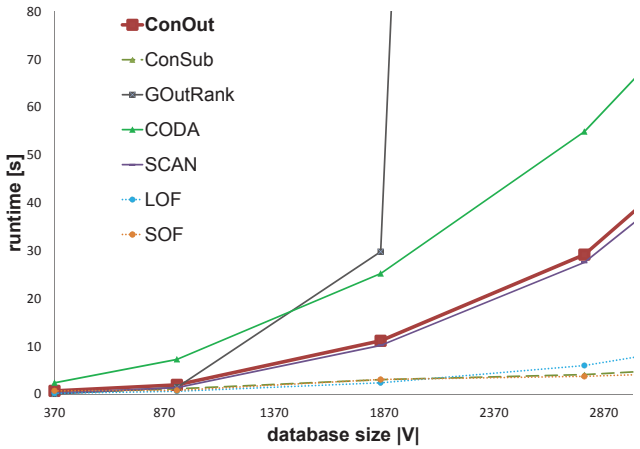
As explained in Section 4, the runtime of our algorithm depends on the database size $|V|$, number of edges $|E|$, and the number of attributes $|A|$. In Figure 3, we depict scalability results w.r.t. all of these properties in comparison to our competitors. Figure 3(a) shows the scalability with increasing number of attributes. The runtime scalability is slightly higher in comparison to traditional approaches due to the combination of both information sources (graph structure and vector data). We deem this tolerable due to the significant quality improvements shown in Figure 2. Compared to CODA, we show better scalability, as its runtime is quadratic in the number of attributes, due to matrix operations for the multi-variate likelihood function of the underlying Gaussian distribution. Additionally, approaches based on subspace selection show much higher runtimes w.r.t. the number of attributes in contrast to the linear time com-



(a) number of attributes



(b) number of edges, $|A| = 1$



(c) number of nodes, $|A| = 1$

Figure 3: Runtime scalability w.r.t. $|A|$,$|E|$, and $|V|$

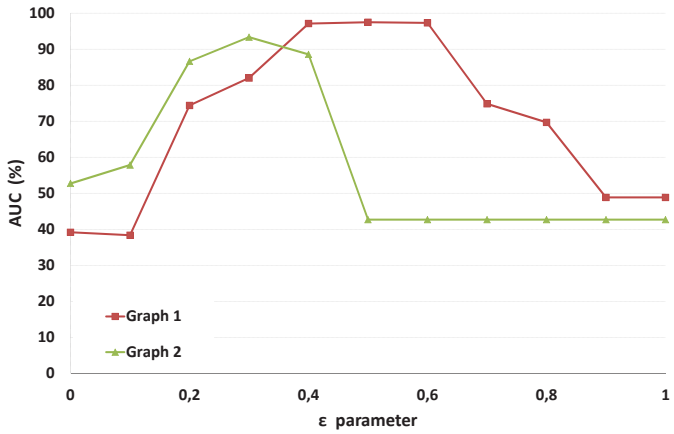


Figure 4: Quality w.r.t. parameter

plexity of *ConOut*. In particular, *GOutRank* does not scale with high dimensionality (up to 30 attributes). Furthermore, we analyze runtimes w.r.t. the database size and the number of edges in Figure 3(b) and Figure 3(c). In contrast to our approach, CODA and *GOutRank* do not scale with dense graphs over 2.5 million of edges as shown in Figure 3(b). Overall, *ConOut* scales well with increasing graph size ($|V|$, $|E|$, and $|A|$). Although CODA, *GOutRank* and *ConSub* consider both graph and attribute information, *ConOut* achieves both better quality and runtime performance.

Parameters.

ConOut uses the parameter ε to specify the local context of each node depending on its connectivity. To evaluate the sensitivity of our parameter we run experiments with different density characteristics (i.e., highly connected *Graph 1* and weaker connected *Graph 2*). Figure 4 shows the AUC quality measure w.r.t. the value of ε. We see that parametrization is robust for a range of top quality results, and there is the expected shift of optima w.r.t. the underlying graph density. Only extreme cases show significant decrease in quality: On the one hand, if the value of ε is too permissive (e.g., values between $0 \dots 0.2$), more nodes are part of the local context, and *ConOut* is hindered in its selection of relevant attributes in this large context. On the other hand, a restrictive setting of ε (e.g., $0.5 \dots 1$) causes very small contexts in which no outliers can be found.

Ranking Functions.

The scoring function of *ConOut* unifies the information from the local graph density (LGD) with the attribute deviation (LAD) in order to obtain accurate rankings for the contextual outliers. For the quality evaluation of our scoring function (cf. Definition 7), we compare it to different baseline aggregation functions (MIN, MAX, SUM) and the raw measures LAD and LGD. We measure the median AUC values obtained by different scoring functions on the 36 synthetic graphs used for the previous quality evaluation. In Figure 1 shows the quality results for the different scoring functions. Local graph density (LGD) and local attribute deviation (LAD) are not able to accurately detect contextual outliers. They fall prey to the information loss as they use only one of the information sources. Aggregation functions such as MAX and *MIN* use both sources. However, they are dominated by one of the measures. The score is not able to make a clear distinction of contextual outliers. For example, two nodes with high local graph densities can have the same score although the attribute deviation is different for each node. The best quality results for contextual outliers are obtained by sum and product which combine both values. However, due to the design of LAD and LGD (cf. Section 3.3), we achieve best results by weighting LAD with a LGD factor. Our proposed scoring function shows overall highest quality results in comparison with all other scores.

Ranking Function	AUC[%]
$LAD \cdot LGD$	**93.3**
$LAD + LGD$	90.44
LAD	90.63
LGD	51.4
$Max(LGD, LAD)$	75.45
$Min(LGD, LAD)$	87.82

Table 1: Results of the different ranking functions

5.2 Real World Data

We use two networks from different domains to evaluate our approach on real world datasets. First, we perform a thorough evaluation of our approach in a subgraph of the co-purchase Amazon network. On this dataset, we have the ground truth for objective quality assessment from a benchmark proposed in [23]. Second, we use the bibliographic repository provided by DBLP for the evaluation of our approach in a larger attributed graph.

5.2.1 Amazon Network

The dataset is a subgraph of the Amazon co-purchase network. It consists of of 124 nodes and 334 edges. Each product in the graph is described by attributes such as product prices, review ratings, and several more (30 attributes) [23]. Figure 5 shows the *Disney* network with three outlier examples and their connectivity to the co-purchase network. Additionally, we also provide their *Amazon Standard Identification Number* for manual verification [2]. In this real-world dataset, each object has been labeled manually by high school students, providing us the ground truth (object is an outlier or not) for quality assessment.

Table 2 gives an overview of quality results. Considering only one source of information – only attributes or graph structure – clearly misses some of the outliers. In particular, the full space technique (LOF) is hindered by the high dimensionality of the product features. On the other hand, subspace analysis (SOF) allows the detection of subspace outliers (e.g., O_2), which is a structural outlier found by graph-based techniques (SCAN) as well. However, none of the paradigms is able to reveal contextual outliers such as O_1 and O_3 (cf. Figure 5). For example, product O_1 is one of the contextual outliers that corresponds to the overpriced film *The Jungle Book (1994)* of *Rudyard Kipling's* hidden in a group of *Read-Along Disney* films. Its local context is not only characterized by the strong connectivity between the nodes in its graph context, but it is also has following relevant attributes: *number of reviews* and *price private seller*.

[2] `http://www.amazon.com/dp/ASIN_VALUE`

Used data	Paradigm	Algorithm	Parameters	AUC	Runtime	Speedup
(1) attributes	full space	LOF [7]	MinPts:20	56.85	41	0.20
	subspace selection	SOF [2]	$\phi : 10$, population : 20	65.88	825	4
(2) graph	graph clustering	SCAN [38]	$\mu : 2, \varepsilon : 0.5$	52.68	4	0.02
(3) both	full space	CODA [14]	$K : 8, r : \frac{6}{124}, \lambda : 0.1$	50.56	2596	13
	subspace cluster analysis	*GOutRank* [23]	configuration in [23]	**86.86**	**26648**	**134**
	global subspace selection	*ConSub* [29]	configuration in [29]	**81.77**	**8930**	**45**
	context selection	*ConOut*	$\varepsilon : 0.5$, *FT test*	**81.21**	**199**	**1**

Table 2: AUC[%] values, Runtime[ms] results and *ConOut's* speedup w.r.t all competitors on the Amazon database [Disney DVD selection].

These outliers can only be detected if graph structure and attributes are combined. CODA considers both data types, but it fails due to the existence of irrelevant attributes. Regarding approaches doing a selection of the attributes, the subspace selection techniques (*GOutRank* and *ConSub*) obtain high quality results, but at much higher runtimes. In contrast, *ConOut* selects a projection of relevant attributes in the local graph neighborhoods. Thus, it allows to identify highly deviating values. It is the most efficient approach in these graph and attribute contexts. As shown in Table 2, *ConOut* shows a 6.5% decrease w.r.t. the best algorithm (*GOutRank*) while being 134 times faster in the runtime. Therefore, it shows the best performance considering both quality and runtime results. It invests some extra runtime compared to traditional approaches for a significant quality improvement. On the other hand, it loses some quality compared to subspace techniques [29, 23], but is more efficient. Thus, it can be applicable for larger networks.

In the following, we discuss the ranking positions between these outliers considering its graph connectivity. These have been ranked at top positions by *ConOut*. Our approach assigns the fourth position to O_1, which is a local outlier with highly deviating attribute values in a strongly connected neighborhood. Second is object O_3 in the ranking, which is weakly connected to its neighborhood and deviates strongly in the rating values from the other co-purchased products. As our ranking function combines the graph and attribute information (cf. Def. 7), O_1 and O_3 have higher scores than the isolated co-purchased product O_2. Regarding the ranking functions, Figure 3 shows the outlier detection quality for each of them. The best AUC values are highlighted in bold, and high quality results that are within 2% are not grayed out. Results show that the unification of both information sources: local graph density and the attribute deviation obtains the highest results. However, the proposed ranking function (cf. Def. 7) outperforms the others. Overall, the evaluation on this real data set demonstrates the existence of local outliers hidden in combinations of the graph structure and the attribute values. We have shown that *ConOut* is more effective than existing algorithms and ranks local outliers accurately according to their degree of deviation in attributed graphs.

5.2.2 DBLP Network

In our second evaluation we use a larger database. We have extracted a part of the DBLP graph with authors represented as nodes and co-authorship as edges. In addition, we describe each author by a scientific profile containing 46 numeric attributes. These attributes provide information on

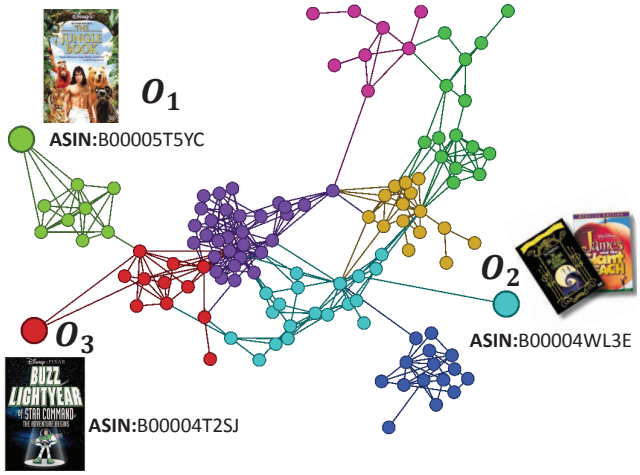


Figure 5: Visualization of 3 hidden outliers on Amazon database

the author's publication ratio at major database, data mining, artificial intelligence, and statistics conferences. The extracted graph consists of 44808 nodes with 119053 edges. In this graph, *ConOut* achieves a runtime of 11.26 seconds. We discuss the outlierness of individual authors w.r.t. their local context in DBLP. Note that we are not looking for truly extraordinary individuals, e.g., with an exceptionally high number of publications in DBLP. Hidden outliers are more local exceptions, e.g., deviating significantly from their co-authors. Let us discuss some of the top-ranked authors found by ConOut.

Pavan Vatturi: He is a structural outlier as Pavan has published only together with one author. He has also high deviating attribute values. His local context is identical to the one of his advisors' *Weng-Keen Wong*. Weng-Keen has a local context with high publishing ratios in *IJCAI*, *AAAI*, and *ICML*, but Pavan has never published in these conferences in contrast to the other authors in his advisors context (e.g., *Ugur Kuter, Santiago Ontañón, Victor R. Lesser*).

Christoph Heinz is a strong connected node in his context consisting of 18 authors (e.g., *Martin Schneider*, *Jens-Peter Dittrich*, *Dieter Korus*). In this context, authors publish frequently on database conferences (e.g., *VLDB, EDBT*, and several more) but they have never publish on the *CIKM* conference. In contrast to his context, Christoph has not

publish in database conferences, which are relevant for his context, but he is the only one that has published on *CIKM*.

Ina Müller-Gormann belongs to a highly connected local context (31 authors) with several relevant attributes (*SIGMOD, KDD, ICDE, ICDM*, and several more). She has published with many authors (e.g., *Arthur Zimek*, *Hans-Peter Kriegel*, ...) of this context, however, she has a clear deviation in the relevant attributes. She has not published in the relevant conferences of her local context.

All these authors are clearly local outliers. The strong connectivity in the graph structure and the highly deviating attribute values in the relevant attributes of their contexts cause their high ranks. Thus, they would not have been found without the local context selection provided by *ConOut*.

Ranking Function	AUC[%]
$LAD \cdot LGD$	**81,21**
$LAD + LGD$	79,66
LAD	78,10
LGD	50,28
$Max(LGD, LAD)$	75,14
$Min(LGD, LAD)$	78,81

Table 3: AUC results for the different ranking functions on Amazon database [Disney DVD selection]

6. CONCLUSION AND DISCUSSION

In this work, we have proposed *ConOut*, a context selection for outlier ranking in graphs with numeric node attributes. Our approach computes locally graph and attribute contexts for each object in the database. For each context, it selects a set of relevant attributes. Relevance of attributes is measured by a statistical test which compares the local and the global variance of each attribute. Thus, outlier ranking relies on a high contrast between outliers and their local context. Overall, *ConOut* computes a high quality outlier ranking that scales well with the number of attributes. Our thorough evaluation on synthetic and real world data shows that it finds local contexts, in contrast to other approaches.

ConOut balances quality with efficiency when joining attribute information with the graph structure. In contrast to approaches based on subspace selection, the runtimes of *ConOut* are significantly lower. To achieve this, we assume that attributes are independent. We do so to give way to an efficient selection of relevant attributes, in linear time. Efficiency is important when it comes to larger attributed graphs. As future work, we aim to design local efficient approaches without assuming the independence of the attributes.

Our approach focuses on numerical node attributes. Thus, a mixture of attribute types such as binary, categorical, and continues values is not explicitly considered in this work. The statistical test would require additional unification of the relevance measure to be applicable in the presence of such heterogeneity. Finally, we also aim at other graph definitions, e.g., considering edge attributes or directed graphs. Such data provides even more information for data mining, however, it also poses novel challenges regarding attribute selection.

Acknowledgments

This work is supported by the Young Investigator Group program of KIT as part of the German Excellence Initiative, by a post-doctoral fellowship of the research foundation Flanders (FWO), and by the German Research Foundation (DFG) within IME Graduate School at KIT.

7. REFERENCES

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

[2] C. Aggarwal and P. Yu. Outlier detection for high dimensional data. *ACM Sigmod Record*, 2001.

[3] L. Akoglu, M. McGlohon, and C. Faloutsos. oddball: Spotting anomalies in weighted graphs. In *PAKDD*, pages 410–421, 2010.

[4] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos. PICS: Parameter-free identification of cohesive subgroups in large attributed graphs. In *SIAM SDM*, pages 439–450, 2012.

[5] R. Andersen, F. R. K. Chung, and K. J. Lang. Local graph partitioning using pagerank vectors. In *FOCS*, pages 475–486, 2006.

[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. Kriegel, R. Ng, J. Sander, et al. LOF: identifying density-based local outliers. *Sigmod Record*, 29(2):93–104, 2000.

[8] D. Chakrabarti. Autopart: Parameter-free graph partitioning and outlier detection. In *PKDD*, pages 112–124, 2004.

[9] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 2009.

[10] D. H. Chau, S. Pandit, and C. Faloutsos. Detecting fraudulent personalities in networks of online auctioneers. In *PKDD*, pages 103–114, 2006.

[11] A. Clauset, M. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[12] M. Davis, W. Liu, P. Miller, and G. Redpath. Detecting anomalies in graphs with numeric labels. In *ACM CIKM*, pages 1197–1202, 2011.

[13] W. Eberle and L. B. Holder. Discovering structural anomalies in graph-based data. In *IEEE ICDM Workshops*, pages 393–398, 2007.

[14] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han. On community outliers and their efficient detection in information networks. In *ACM SIGKDD*, pages 813–822, 2010.

[15] S. Günnemann, I. Färber, B. Boden, and T. Seidl. Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. In *IEEE ICDM*, pages 845–850, 2010.

[16] S. Günnemann, I. Färber, S. Raubach, and T. Seidl. Spectral subspace clustering for graphs with feature vectors. In *IEEE ICDM*, 2013.

[17] F. Keller, E. Müller, and K. Böhm. HiCS: High contrast subspaces for density-based outlier ranking. In *IEEE ICDE*, pages 1037–1048, 2012.

Algorithm	O_1	O_2	O_3
	ASIN: B00005T5YC	**ASIN:** B00004WL3E	**ASIN:** B00004T2SJ
ConOut	**3**	**8**	**7**
CODA	×	×	×
SCAN	×	√	×
LOF	96	89	57
SOF	77	**2**	86
ConSub	8	**3**	**2**
GOutRank	12	29	20

Table 4: Ranking results from the top ranked outliers on Amazon database [Disney DVD selection]

[18] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *VLDB*, pages 392–403, 1998.

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

[20] M. McPherson, L. S. Lovin, and J. M. Cook. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[21] F. Moser, R. Colak, A. Rafiey, and M. Ester. Mining cohesive patterns from graphs with feature vectors. In *SIAM SDM*, pages 593–604, 2009.

[22] E. Müller, I. Assent, P. I. Sanchez, Y. Mülle, and K. Böhm. Outlier ranking via subspace analysis in multiple views of the data. In *IEEE ICDM*, pages 529–538, 2012.

[23] E. Müller, P. Iglesias, Y. Mülle, and K. Böhm. Ranking outlier nodes in subspaces of attributed graphs. In *IEEE ICDE Workshops*, 2013.

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

[25] M. E. Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, 2003.

[26] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *ACM SIGKDD*, pages 631–636, 2003.

[27] D. Rees. *Essential statistics*. Texts in statistical science. Chapman & Hall/CRC, 2001.

[28] P. Rousseeuw, A. Leroy, and J. Wiley. *Robust regression and outlier detection*. Wiley Online Library, 1987.

[29] P. I. Sanchez, E. Müller, F. Laforet, F. Keller, and K. Böhm. Statistical selection of congruent subspaces for mining attributed graphs. In *IEEE ICDM*, pages 647–656, 2013.

[30] X. Song, M. Wu, C. M. Jermaine, and S. Ranka. Conditional anomaly detection. *IEEE Trans. Knowl. Data Eng.*, 19(5):631–645, 2007.

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

[32] H. Sun, J. Huang, J. Han, H. Deng, P. Zhao, and B. Feng. gSkeletonClu: Density-based network clustering via structure-connected tree division or agglomeration. In *IEEE ICDM*, pages 481–490, 2010.

[33] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Relevance search and anomaly detection in bipartite graphs. *SIGKDD Explorations*, 7(2):48–55, 2005.

[34] J. Tang and H. Liu. Unsupervised feature selection for linked social media data. In *ACM SIGKDD*, pages 904–912, 2012.

[35] H. Tong and C.-Y. Lin. Non-negative residual matrix factorization with application to graph anomaly detection. In *SIAM SDM*, pages 143–153, 2011.

[36] P. Vatturi and W. Wong. Category detection using hierarchical mean shift. In *ACM SIGKDD*, pages 847–856, 2009.

[37] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[38] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *ACM SIGKDD*, pages 824–833, 2007.

[39] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *PVLDB*, 2(1):718–729, 2009.