Robust Contextual Outlier Detection: Where Context Meets Sparsity

Jiongqian Liang and Srinivasan Parthasarathy
Computer Science and Engineering, The Ohio State University, Columbus, OH, USA
{liangji,srini}@cse.ohio-state.edu

ABSTRACT

Outlier detection is a fundamental data science task with applications ranging from data cleaning to network security. Recently, a new class of outlier detection algorithms has emerged, called *contextual outlier detection*, and has shown improved performance when studying anomalous behavior in a specific context. However, as we point out in this article, such approaches have limited applicability in situations where the context is sparse (i.e., lacking a suitable frame of reference). Moreover, approaches developed to date do not scale to large datasets. To address these problems, here we propose a novel and robust approach alternative to the state-of-the-art called RObust Contextual Outlier Detection (`ROCOD`). We utilize a local and global behavioral model based on the relevant contexts, which is then integrated in a natural and robust fashion. We run `ROCOD` on both synthetic and real-world datasets and demonstrate that it outperforms other competitive baselines on the axes of efficacy and efficiency. We also drill down and perform a fine-grained analysis to shed light on the rationale for the performance gains of `ROCOD` and reveal its effectiveness when handling objects with sparse contexts.

1. INTRODUCTION

"An *outlier* is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism" [6]. Detecting outliers finds applications in a wide range of domains including cyber-intrusion detection, epidemiology studies, fraud detection, and data cleaning. A number of efforts in this space have treated all attributes, associated with a data point, in an egalitarian fashion. However, in many domains, some attributes are usually highly related to the outlier behavior, called *behavioral attributes* or *indicator attributes*, while other attributes only provide contexts of the behavior, termed *contextual attributes*. It has been demonstrated recently that by distinguishing contextual attributes from behavioral attributes, the precision of outlier detection can be increased [4, 13, 14, 17, 5]. Formally, *contextual outlier* or *conditional anomaly* is defined as an object with behavior deviating from other objects with similar contextual information [4, 13, 14]. Contextual attributes are used to define the contexts, and objects sharing similar contexts with an object form its *reference group*. Behavior attributes, on the other hand, are used for examining outlierness in a specific context, compared to the reference group.

One pitfall of existing contextual outlier detection methods is that they might fail to examine the outlierness of objects with sparse contexts. To intuitively show this, we use a toy example of credit card fraud detection. For simplicity, suppose we intend to detect suspicious transactions and only monitor two variables, the annual income of cardholders and the amount of each transaction (shown in Figure 1).

The Importance of Contextual Attributes: Though contextual attributes are not directly related to the anomalous behavior, they provide useful information on contexts for outlier detection. In the example of Figure 1, transaction amount and annual income can be respectively regarded as the behavioral attribute and contextual attribute. If we merely consider the behavioral attribute, then points G, E and F will not be flagged as outliers, which is not reasonable. Therefore, we need auxiliary information from the contextual attributes to pinpoint the outliers.

Incorporating Contextual Attributes: One conventional way to incorporate contextual attributes is treating them similarly with behavioral attributes by concatenating the two. In the example above, all the boundary points (A, B, ..., G) will be reported as outliers. Another way is to use existing contextual outlier detection techniques.Following the definition of the contextual outlier, we can examine the difference of behavioral attribute between a point and other points with the similar contextual attribute. One will then report point C and D as outliers since their behavior values (y values) are quite different from other points with similar contextual values (x values).

Limitations of Existing Approaches: These two methods have limitations on addressing objects with sparse contexts (A, B, E, F and G). The former approach tends to overestimate the outlierness of objects with unusual contexts because the outlierness score can be directly contributed by contextual attributes. Moreover, the latter approach fails to properly examine the outlierness of objects with sparse contexts. In fact, applying state-of-the-art contextual outlier detection algorithm [13] for the toy example, we obtain outlierness score ranking of $B > D > C > A \gg G, E, F$. However, a close look at our example reveals that A and B

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

http://dx.doi.org/10.1145/2983323.2983660

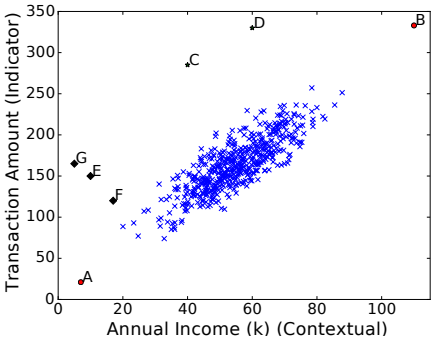


Figure 1: Toy example of contextual outliers.

should ideally not be flagged as outliers since they follow the normal pattern between the two attributes that a lower value in contextual attribute corresponds to a lower value in behavioral attribute and vice versa. Therefore, we need a more robust approach for outlier detection to distinguish E, F, G with A and B, giving A and B lower outlierness scores. To the best of our knowledge, this paper is the first piece of work attempting to rectify this problem.

To this end, we propose a refined approach, called ROBust Contextual Outlier Detection (ROCOD), to better exploit contextual attributes to assist outlier detection, ROCOD can particularly address the problems caused by the contextual sparsity, making it more robust towards broad outlier detection tasks. Specifically, we propose *local expected behavior* and *global expected behavior* models that seek to understand the dependence structure between behavioral attributes and contextual attributes. Local expected behavior models are designed to predict the behavior by referring to the objects with similar context, called *contextual neighbors*. Global expected behavior models learn the dependence structure between contextual attributes and behavioral attributes from the data, and infer the behavior in a holistic sense. We then propose a regularized integration function to naturally couple both types of behavior models based on the number of contextual neighbors, which naturally accommodates objects with sparse contexts. We run ROCOD on both synthetic and real-world datasets and compare it with *five* state-of-the-art outlier detection techniques. Our experimental results show that ROCOD outperforms all the baselines on both effectiveness (measured by different metrics) and efficiency.

2. ROCOD

To the best of our knowledge, none of the existing works addresses the problem caused by the sparsity of contexts in contextual outlier detection (see our extended paper [7] for a detailed review of related work). In this section, we introduce our RObust Contextual Outlier Detection (ROCOD) approach to tackle this problem in detail.

2.1 Problem Formulation

Given a series of objects, the i-th object can be represented as

$$z^{(i)} = \begin{pmatrix} x^{(i)} \\ y^{(i)} \end{pmatrix} = (x_1^{(i)}, x_2^{(i)}, ..., x_C^{(i)}, y_1^{(i)}, y_2^{(i)}, ..., y_B^{(i)})^T \quad (1)$$

where $z^{(i)}$ is the whole attribute vector, $x^{(i)}$ is contextual attribute vector and $y^{(i)}$ is behavioral attribute vector. Without loss of generality, we assume $x^{(i)}$ has C dimensions and $y^{(i)}$ has B dimensions. Following this, the whole dataset can be denoted as $Z = \langle z^{(1)}, z^{(2)}, ..., z^{(N)} \rangle$, where N is the total number of objects. Among them, we denote O as the set of outliers. Given the dataset Z, our goal is to assign each object i with an outlierness score S_i so that outliers in O have much higher values than other objects.

Conforming to the definition of the contextual outlier, the outlierness of an object arises from the abnormal behavioral attributes in its particular context. In other words, provided the contextual attributes, there is underlying pattern restricting the behavioral attributes to some expected values, beyond which one object will be considered as an outlier. We here call it the *dependent pattern* and define the expected behavior as follows.

DEFINITION 1. ***Expected Behavior.*** *For object i, the expected behavior is the values of its behavioral attributes predicted by the underpinning dependent pattern, given its contextual attributes $x^{(i)}$.*

Formally, denote the underpinning dependent pattern as a function $f(\cdot)$, then the expected behavior is

$$\hat{y}^{(i)} = f(x^{(i)}) \quad (2)$$

Following the definition, a contextual outlier is the object with behavioral attributes violating the dependent pattern under its contextual attributes. We can gauge the outlierness score of object i by measuring the difference of $\hat{y}^{(i)}$ and $y^{(i)}$. Therefore, revealing the expected behavior of each object is a crucial part of flagging contextual outliers.

However, it is nontrivial to find out the pattern function $f(\cdot)$ and the expected behavior since the data distribution is usually unknown and the data is inherently noisy. We next discuss how we approach this problem from both a local and global perspective and how to combine them.

2.2 Local Expected Behavior Modeling

We first study the dependent pattern and expected behavior from the local aspect. We define *contextual neighbors*:

DEFINITION 2. ***Contextual Neighbors.*** *Contextual neighbors of an object are the objects that are similar to it on contextual attributes.*

Formally, the set of contextual neighbors of object i is

$$CN_i = \left\{ j : j \in D \wedge j \neq i \wedge sim(x^{(i)}, x^{(j)}) \geq \alpha \right\} \quad (3)$$

where α is a predefined similarity threshold and $sim(\cdot)$ is a similarity function of two vectors. $D = \{1, 2, ..., N\}$ denotes the set of objects indexes. While $sim(\cdot)$ can be any reasonable similarity function, we choose cosine similarity here.

We then define *local expected behavior* as follows.

DEFINITION 3. ***Local Expected Behavior.*** *The local expected behavior of an object is the average values of behavioral attributes among all its contextual neighbors.*

This definition hinges on the underlying assumption of contextual outlier detection that objects with similar contextual attributes share similar behavioral attributes [4, 13, 14] and is also a natural extension of Tobler's first law of geography [16]. To formalize it, the expected behavior of object i given contextual attributes $x^{(i)}$ is

$$\hat{y}^{(i)} = \Phi(x^{(i)}) = \frac{\sum_{j \in CN_i} y^{(j)}}{|CN_i|} \quad (4)$$

where $\Phi(\cdot)$ denotes local behavior pattern.

The local behavior pattern is tightly tied to the definition of contextual outlier detection and is supposed to directly reflect the dependent relationship between behavioral attributes and contextual attributes. Moreover, it does not make any assumption on the distribution of data. While this local property has been widely used in spatial and temporal outlier detection [19, 8], we generalize it to a broader range of applications with arbitrarily defined contexts. However, since the local expected behavior relies on the contextual neighbors, it will be inapplicable if one object does not have contextual neighbors. Therefore, the local expected behavior cannot be inferred for all the objects and we need a more robust way to compute the expected behavior.

2.3 Global Expected Behavior Modeling

We now introduce a global approach to capture the underlying dependent pattern in the data and infer the expected behavior, called *global expected behavior*. A natural way of capturing the global dependent relationship between behavioral attributes and contextual attributes is to adopt regression models. For each behavioral attribute, we can learn a regression model considering contextual attributes as features and the behavioral attributes as the target values. In total, we learn B regressors from the dataset with B behavioral attributes. With the regression models, we hereby define global expected behavior.

DEFINITION 4. ***Global Expected Behavior.*** *The global expected behavior of an object is the values of behavioral attributes predicted by the regression models taking its contextual attributes as input.*

Formally, the global expected behavior of object i is

$$\hat{y}^{(i)} = \Psi(x^{(i)}) \tag{5}$$

where $\Psi(\cdot)$ incorporates the regression models learned from the whole dataset using contextual attributes as independent variables and behavioral attributes as dependent variables. $\Psi(\cdot)$ takes the contextual attributes $x^{(i)}$ as input and outputs the expected behavior attribute vector $\hat{y}^{(i)}$. Note that we can adopt any regression model here, either linear or non-linear. Obviously, this manner of behavior modeling is a holistic approach since the regression models are learned from the whole dataset.

2.4 Ensemble Expected Behavior

So far we have introduced two different perspectives to depict the underpinning relationship between behavioral attributes and contextual attributes, and used them to compute local and global expected behavior. Local expected behavior adopts the contextual neighbors as the reference frame while global expected behavior is predicted by adopting the regression models learned from the whole dataset.

These two ways of inferring expected behavior offer complementary benefits. Local expected behavior is a model-free approach and should have a lower bias when the number of contextual neighbors is large. However, local expected behavior contains larger variance in general and is prone to noise, especially when the number of contextual neighbors is small. In fact, it cannot be applied at all when there is no contextual neighbor. On the other hand, global expected behavior infers the values considering the dependent relationship between the two categories of attributes in a holistic way. Therefore, it is expected to contain smaller variance and is more robust against noise. However, its bias tends to be larger since it cannot capture the fine-grained local pattern for each object.

Considering the different advantages of the two approaches, we intend to find an appropriate manner to integrate them. One possible method is to simply take a weighted sum of them with pre-defined weights. However, it is not trivial to determine reasonable weights and the issue of zero contextual neighbors still remains.

To resolve these issues, we propose an adaptive weighted sum to integrate these two methods. Instead of fixing the weights for all objects, we adjust the weights according to the number of contextual neighbors of each object. Given an object i, we define the ensemble expected behavior as

$$\hat{y}^{(i)} = \lambda_i \cdot \Phi(x^{(i)}) + (1 - \lambda_i) \cdot \Psi(x^{(i)}) \tag{6}$$

where

$$\lambda_i = \frac{\sqrt{|CN_i|}}{\max\limits_{1 \le j \le N} \sqrt{|CN_j|}}. \tag{7}$$

Here, $\Phi(x^{(i)})$ and $\Psi(x^{(i)})$ are respectively local expected behavior and global expected behavior as defined in Equation 4 and Equation 5. The intuition behind this weighted combination is that if an object has a sufficient number of contextual neighbors, we believe the contextual neighbors are a reliable reference frame and place more weight on local expected behavior. Otherwise, we put more weight on the global metric. In addition, we take the square root transformation on $|CN_i|$ to improve the normality of $\lambda_i \cdot \Phi(x^{(i)})$ [2].

By setting weight on local expected behavior proportional to the square root of the number of contextual neighbors, the model is more robust and can appropriately deal with context sparsity. In particular, it properly solves the problem of zero contextual neighbors by naturally setting the weight of local expected behavior to zero when $|CN_i| = 0$.

With the ensemble expected behavior for each object, we measure the outlierness score of object i by the difference of expected behavior $\hat{y}^{(i)}$ and real behavior $y^{(i)}$, specifically by the L2-norm $\left\| y^{(i)} - \hat{y}^{(i)} \right\|_2$. Here, the L2-norm of the difference assumes each behavioral attribute contributes equally to the outlierness score. However, this might not be true since the attributes can have different credibility at flagging the outlier. For example, if the real values of one behavioral attribute are highly consistent with the expected ones, indicating the pattern is well captured by our approach as a whole, then a slight difference of real value and expected value should be a strong sign of outlierness. Following this intuition, we weight each behavioral attribute based on how good the expected behavior on capturing the real behavior values. For each behavioral attribute, we use the coefficient of determination [9] to measure consistency between the real behavior and the expected one. For j-th behavioral attribute, the coefficient of determination is calculated by

$$R^2(y_j, \hat{y}_j) = 1 - \frac{\sum\limits_{1 \le i \le N} ({y_j}^{(i)} - \hat{y}_j^{(i)})^2}{\sum\limits_{1 \le i \le N} ({y_j}^{(i)} - \bar{y}_j)^2} \tag{8}$$

where $\bar{y}_j = \frac{1}{N} \sum\limits_{1 \le i \le N} {y_j}^{(i)}$, ${y_j}^{(i)}$ is the value of j-th behavioral attribute for object i, and $\hat{y}_j^{(i)}$ is the expected value of it using our approach. We define the weight of j-th behavioral

attribute as $w_j = \max(R^2(y_j, \hat{y}_j), 0)$ and therefore the range of w_j is $[0, 1]$. We now can calculate the outlierness score of object i using

$$S_i = \left\| W^T (y^{(i)} - \hat{y}^{(i)}) \right\|_2 \quad (9)$$

where $W = (w_1, w_2, ..., w_B)^T$.

ROCOD detects outliers using Equation 9. Specifically, ROCOD uses it to capture the outlierness score for each object and the n objects with highest outlierness scores are selected as outliers, where n is the number of outliers defined by users.

Furthermore, we scale up our algorithm by leveraging Locality Sensitive Hashing (LSH) to efficiently find the contextual neighbors and parallelizing the computation. See more details in the extended version of this paper [7].

3. EXPERIMENTS AND ANALYSIS

In this section, we run ROCOD on several datasets and compare the performance with a series of baselines. For more details about experiment setup and analysis, see the extended paper [7].

3.1 Experimental Setup

3.1.1 Dataset and Data Preprocessing

Evaluating contextual outlier detection algorithms is challenging due to the lack of ground truth. In this paper, we employ six datasets to evaluate the performance of our algorithm, two of which contain labeled ground-truth outliers. For the four datasets without ground-truth outliers, we inject contextual outliers using the perturbation scheme described by Song *et al.* [13] – a de-facto standard for evaluating contextual outlier detection techniques. This scheme works as follows. To inject one outlier into a dataset with N objects, we uniformly select an object $z^{(i)} = (x^{(i)}, y^{(i)})^T$ at random. We then randomly select $p = \min(50, \frac{N}{4})$ objects from the dataset, among which we pick the object $z^{(j)} = (x^{(j)}, y^{(j)})^T$ such that the Euclidean distance between $y^{(i)}$ and $y^{(j)}$ is maximized. We add a new object $z' = (x^{(i)}, y^{(j)})^T$ into the dataset as a contextual outlier [1].

The basic information of the six datasets is shown in Table 1. It includes: 1) **Synthetic** dataset, which is generated using the CAD model [13]. 2) **Bodyfat** dataset from CMU statlib [2], where attributes on body fat percentage are treated as behavioral attributes and other physical features are considered as contextual attributes. 3) **ElNino** dataset from UCI ML repository [3], where we use the temporal and spatial attributes as contextual attributes and regard attributes on winds, humidity and temperature as behavioral attributes. 4) **Houses** dataset from CMU statlib, where we use the house price as the behavioral attributes and other attributes as contextual attributes (such as median income). 5) **YouTube-Twitter** dataset from previous work [1], where attributes from Twitter are adopted as contextual attributes while the attributes from YouTube are regarded as behavioral attributes. 6) **Kddcup99** dataset adapted from KDD Cup 1999 [4], where we only retain *u2r* and *r2l* attacks and treat them as outliers while removing other attacks [10]. Moreover, considering that *service*, *duration*, *src_bytes* and *dst_bytes* are most essential attributes for intrusion behavior [20], we use them as behavioral attributes and the rest of attributes are treated as contextual attributes.

[1]In total, we inject $1\% * N$ outliers into the dataset (except for Bodyfat due to the small size of the dataset).

[2]ftp://rcom.univie.ac.at/mirrors/lib.stat.cmu.edu/

[3]https://archive.ics.uci.edu/ml/datasets/El+Nino

[4]http://kdd.ics.uci.edu/databases/kddcup99/task.html

3.1.2 Baselines and Evaluation Metrics

We compare ROCOD to the state-of-the-art approaches on outlier detection. They contain 1) Conditional Anomaly Detection (**CAD**) proposed by Song *et al.* [13]. 2) Locality Sensitive Outlier Detection (**LSOD**) [18], which is a representative of distance-based anomaly detection algorithm leveraging locality-sensitive hashing and other techniques [11] for efficiency optimization. 3) Local Outlier Factor (**LOF**) [3], which is a local method comparing the local reachability density of each node to its neighbors. 4) Connectivity-based Outlier Factor (**COF**) [15], an extension of LOF differentiating low density from isolation. 5) Gaussian Mixture Model (**GMM**) [12] measuring the outlierness score by probability density.

These approaches output a full list of objects ranked by their outlierness scores with higher ones on the top, and return the first n objects as outliers. To comprehensively evaluate the performance of outlier detection, we use three different metrics: 1) AUC of Precision-Recall Curve. 2) Top-n Precision, defined as the precision of the objects ranked among top-n, which is also called *precision at n*. 3) Top-n normalized Discounted Cumulative Gain (nDCG), measuring the effectiveness of ranking for the first n objects.

3.2 Experiment Results and Analysis

We run the experiments on a Linux Machine with two Intel Xeon x5650 2.67GHz CPUs. It contains 12 cores and 48GB of RAM. All the algorithms are implemented in C++ and compiled using Intel compiler. OpenMP is used to exploit the parallelism. Also, we have the following settings for the outlier detection methods: 1) For CAD and GMM, we set the number of Gaussian components as 30 and the maximum number of EM iterations as 100. 2) For LSOD, we use the distance to the 30-th nearest neighborhood as the outlier score. 3) For COF and LOF, we set the range of size of neighborhoods from 10 to 100 as suggested by the authors. 4) For ROCOD, the cosine similarity threshold α is chosen by looking at the distribution of randomly sampled pairs of objects (more details in [7]). We choose Ridge regression for the linear model and tree regression for the nonlinear model, denoted as ROCOD_1 and ROCOD_2 respectively. As comparisons, we also use only local expected behavior and global expected behavior to flag outliers, denoted as **LEB** and **GEB**. Results are shown in Table 2, which presents the performance of each approach on detecting outliers in all the datasets, measured by three different metrics. We highlight some observations below.

1) We can see that ROCOD (either ROCOD_1 or ROCOD_2) performs the best at almost all the datasets. In some datasets, especially Synthetic, Houses and KDDcup99, ROCOD significantly outperforms other baselines on all three evaluation metrics. For example, the top-100 precision of ROCOD_2 on KDDcup99 is twice as much as the best of baselines (CAD). The advantage of ROCOD is more pronounced in terms of top-n metrics, indicating our method is able to show outliers at the top more precisely. In Synthetic and ElNino dataset,

Datasets	Outliers	# Objects N	# Outliers	Contextual Attributes	Behavior Attributes
Synthetic	Injected by perturbation scheme.	50,500	500	20	20
Bodyfat	Injected by perturbation scheme.	277	25	13	2
ElNino	Injected by perturbation scheme.	94,874	939	6	5
Houses	Injected by perturbation scheme.	20,846	206	8	1
YouTube-Twitter	Promotional users.	62,458	2,974	48	41
Kddcup99	u2r and r2l attacks.	98,372	1,094	47	69

Table 1: Basic information of the datasets. Outliers are injected by perturbation scheme for datasets without ground truth.

Synthetic									
Metrics	ROCOD_1	ROCOD_2	**LEB**	**GEB**	**CAD**	**GMM**	**LSOD**	**LOF**	**COF**
PRC (AUC)	0.392	**0.913**	0.781	0.679	0.628	0.055	0.265	0.384	0.135
Top-100 Precision	0.740	**0.950**	0.890	0.860	0.880	0.150	0.450	0.710	0.230
Top-100 nDCG	0.749	**0.960**	0.867	0.877	0.891	0.192	0.431	0.730	0.280
Bodyfat									
Metrics	ROCOD_1	ROCOD_2	**LEB**	**GEB**	**CAD**	**GMM**	**LSOD**	**LOF**	**COF**
PRC (AUC)	**0.750**	**0.750**	0.614	**0.750**	**0.750**	0.430	0.644	0.725	0.667
Top-10 Precision	0.900	**1.000**	0.800	0.900	0.900	0.400	0.300	0.400	0.300
Top-10 nDCG	0.936	**1.000**	0.875	0.933	0.936	0.287	0.206	0.305	0.215
ElNino									
Metrics	ROCOD_1	ROCOD_2	**LEB**	**GEB**	**CAD**	**GMM**	**LSOD**	**LOF**	**COF**
PRC (AUC)	0.670	**0.990**	0.964	0.883	0.220	0.461	0.806	0.797	0.767
Top-100 Precision	0.960	**1.000**	0.990	**1.000**	0.400	0.790	**1.000**	0.950	**1.000**
Top-100 nDCG	0.970	**1.000**	0.968	**1.000**	0.404	0.825	**1.000**	0.947	**1.000**
Houses									
Metrics	ROCOD_1	ROCOD_2	**LEB**	**GEB**	**CAD**	**GMM**	**LSOD**	**LOF**	**COF**
PRC (AUC)	0.656	**0.766**	0.312	0.634	0.232	0.135	0.101	0.116	0.119
Top-100 Precision	0.740	**0.840**	0.300	0.650	0.350	0.210	0.080	0.260	0.250
Top-100 nDCG	0.694	**0.860**	0.226	0.717	0.461	0.222	0.067	0.247	0.245
YouTube-Twitter									
Metrics	ROCOD_1	ROCOD_2	**LEB**	**GEB**	**CAD**	**GMM**	**LSOD**	**LOF**	**COF**
PRC (AUC)	0.141	**0.146**	0.136	0.124	0.125	0.131	0.124	0.105	0.106
Top-100 Precision	**0.530**	0.470	0.440	0.360	0.370	0.180	0.280	0.230	0.260
Top-100 nDCG	**0.605**	0.440	0.426	0.413	0.362	0.168	0.272	0.223	0.240
KDDcup99									
Metrics	ROCOD_1	ROCOD_2	**LEB**	**GEB**	**CAD**	**GMM**	**LSOD**	**LOF**	**COF**
PRC (AUC)	0.137	**0.143**	0.071	0.051	0.027	0.128	0.027	0.019	0.014
Top-100 Precision	0.390	**0.600**	0.070	0.020	0.300	0.000	0.070	0.020	0.000
Top-100 nDCG	0.293	**0.518**	0.129	0.031	0.316	0.000	0.085	0.015	0.000

Table 2: Performance comparisons of baselines on 6 datasets. ROCOD_1 uses the linear model in the global expected behavior while ROCOD_2 adopts the non-linear model. LEB and GEB are the approaches utilizing only local expected behavior and global expected behavior (with non-linear model) respectively. Three metrics are used to evaluate the performance (higher is better). Best performances w.r.t. each metric are shown in bold.

the top-100 precision and nDCG of our approach is almost perfect (very close to 1.0).

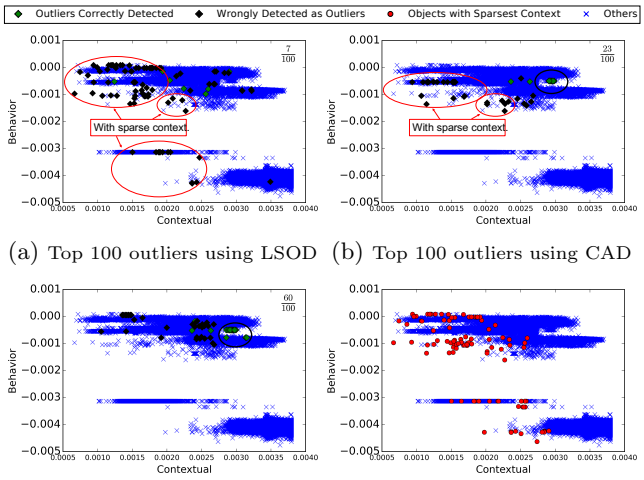
2) Without separating contextual and behavioral attributes, general outlier detection approaches (e.g., LSOD, LOF and COF), perform poorly on the datasets. This issue is more evident on Bodyfat and Houses dataset, which contain more contextual attributes than behavioral attributes. The main reason is that these approaches simply combine contextual attributes with behavioral attributes and the effect of contextual attributes on outlierness score may obfuscate the role of behavioral attributes. This phenomenon is obvious on the dataset of Houses and Kddcup99.

3) ROCOD with the non-linear model for global expected behavior (ROCOD_2) outperforms the one with the linear model (ROCOD_1). This is not surprising and consistent with our intuition that the non-linear model is more capable of modeling the complex relationship among attributes. In fact, ROCOD_2 obtains the best performance in all the datasets except YouTube-Twitter, where ROCOD_1 performs the best on top-100 precision and nDCG while ROCOD_2 is better in terms of AUC of the precision-recall curve.

We also measure the wall-clock running time of all the methods and find out that ROCOD is more efficient than other baselines. Refer to the extended paper for more experimental results on the scalability of ROCOD [7].

3.3 Drilling Down on Efficacy Gains

Here, we drill-down to distil the performance gains of ROCOD. We take Kddcup99 dataset as an example and visualize outliers flagged by different methods among all other objects in 2-D coordinates. In order to visualize high-dimensional data, we extract the largest component from the contextual attributes space and behavioral attributes space respectively and plot the data points directly in the 2-D coordinate. Figure 2 shows the visualization of all the objects. Green diamonds are outliers correctly identified while black diamonds are normal objects but flagged incorrectly as outliers by the approaches. We show the top-100 outliers detected by each method and the precision is shown at the upper right corner of each plot. We include the results from LSOD, CAD and ROCOD, which are the three best approaches in this dataset. Moreover, we also identify 100 objects with sparsest contexts, i.e., their contextual attributes are very different from others (marked as red dots in Figure 2d).



(a) Top 100 outliers using LSOD (b) Top 100 outliers using CAD

(c) Top 100 outliers using ROCOD (d) Objects with Sparsest Context

Figure 2: Visualization of Kddcup99 dataset in the 2-D coordinate. (a)-(c) show outliers detected by different approaches. Red round circles in (a) and (b) highlight objects with sparsest contexts. Black round circles in (b) and (c) highlight the same clusters of outliers correctly detected by CAD and ROCOD. (d) shows 100 objects with the sparsest contextual attributes, indicated by red dots.

Comparing Figure 2c with Figure 2a and Figure 2b, we notice that LSOD and CAD tend to mistakenly detect similar groups of normal objects as outliers, which are highlighted in red circles in the plots (Figure 2a and Figure 2b), while ROCOD avoids similar mistakes. To understand the reason for this observation, we look at these groups of objects in Figure 2d and find out that most of them are show in red dots and therefore are objects containing sparse contextual attributes. This strongly supports our statement that existing outlier detection techniques (such as LSOD and CAD) tend to assign higher outlierness scores to objects with anomalous contextual attributes though they are normal considering their behavioral attributes. Even state-of-the-art approach CAD cannot properly resolve this issue. ROCOD, however, is not affected much by these objects and is able to impartially measure the outlierness scores of them. Moreover, it is also interesting to observe that though ROCOD and CAD are two totally different approaches, they correctly detect the same group of outliers, circled by black bold rectangles in Figure 2b and 2c. In general, ROCOD is much better than CAD at correctly identifying outliers in this dataset.

4. CONCLUSION

We propose ROCOD to exploit contextual attributes for detecting outliers, particularly dealing with the sparsity of context. We introduce local expected behavior and global expected behavior models to infer the behavioral attributes and describe a natural algorithm to fuse them. Experimental results show that ROCOD detects outliers more accurately and efficiently than prior approaches.

Acknowledgements. This work is supported by NSF Award NSF-EAR-1520870 and NSF-DMS-1418265.

5. REFERENCES

[1] A. Abisheva *et al.* Who watches (and shares) what on youtube? and when?: using twitter to understand youtube viewership. In *WSDM*. ACM, 2014.

[2] M. Bartlett. The square root transformation in analysis of variance. *JASA*, pages 68–78, 1936.

[3] M. M. Breunig *et al.* Lof: identifying density-based local outliers. In *SIGMOD*. ACM, 2000.

[4] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *CSUR'09*, 41(3):15, 2009.

[5] M. Hauskrecht et al. Conditional outlier approach for detection of unusual patient care actions. In *AAAI'13*.

[6] D. M. Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[7] J. Liang and S. Parthasarathy. Robust contextual outlier detection: Where context meets sparsity. *arXiv preprint arXiv:1607.08329*, 2016.

[8] C.-T. Lu, D. Chen, and Y. Kou. Algorithms for spatial outlier detection. In *ICDM'03*. IEEE, 2003.

[9] N. J. Nagelkerke. A note on a general definition of the coefficient of determination. *Biometrika*, 1991.

[10] H. V. Nguyen *et al.* Mining outliers with ensemble of heterogeneous detectors on random subspaces. In *DSFAA*. Springer, 2010.

[11] G. H. Orair *et al.* Distance-based outlier detection: consolidation and renewed bearing. *VLDB*, 2010.

[12] S. Roberts and L. Tarassenko. A probabilistic resource allocating network for novelty detection. *Neural Computation*, 6(2):270–284, 1994.

[13] X. Song *et al.* Conditional anomaly detection. *TKDE*, 19(5):631–645, 2007.

[14] G. Tang *et al.* Mining multidimensional contextual outliers from categorical relational data. In *SSDBM*, page 43. ACM, 2013.

[15] J. Tang *et al.* Enhancing effectiveness of outlier detections for low density patterns. In *PAKDD*, pages 535–548. Springer, 2002.

[16] W. R. Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46:234–240, 1970.

[17] M. Valko *et al.* Conditional anomaly detection with soft harmonic functions. In *ICDM*, 2011.

[18] Y. Wang *et al.* Locality sensitive outlier detection: A ranking driven approach. In *ICDE*, 2011.

[19] A. S. Weigend, M. Mangeas, and A. N. Srivastava. Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. *IJNS*, 1995.

[20] K. Yamanishi *et al.* On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *DMKD*, 2004.