<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *IoTStream Workshop at ECML-PKDD 2020, Ghent-Belgium, September 14 –18, 2020*.

Citation for the original published paper:

Cheng, L., Sundaresh, S., Bouguelia, M-R., Dikmen, O. (2020)
Interactive Anomaly Detection Based on Clustering and Online Mirror Descent
In:

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-43315>

Interactive Anomaly Detection Based on Clustering and Online Mirror Descent

Lingyun Cheng, Sadhana Sundares, Mohamed-Rafik Bouguelia, and Onur Dikmen

Department of Intelligent Systems and Digital Design, Halmstad University, Sweden
{linche18,sadsun18}@student.hh.se,
{mohamed-rafik.bouguelia,onur.dikmen}@hh.se

Abstract. In several applications, when anomalies are detected, human experts have to investigate or verify them one by one. As they investigate, they unwittingly produce a label - true positive (TP) or false positive (FP). In this paper, we propose a method (called OMD-Clustering) that exploits this label feedback to minimize the FP rate and detect more relevant anomalies, while minimizing the expert effort required to investigate them. The OMD-Clustering method iteratively suggests the top-1 anomalous instance to a human expert and receives feedback. Before suggesting the next anomaly, the method re-ranks instances so that the top anomalous instances are similar to the TP instances and dissimilar to the FP instances. This is achieved by learning to score anomalies differently in various regions of the feature space. An experimental evaluation on several real-world datasets is conducted. The results show that OMD-Clustering achieves significant improvement in both detection precision and expert effort compared to state-of-the-art interactive anomaly detection methods.

Keywords: Interactive Anomaly Detection · Outlier Detection · User Feedback · Expert Effort

1 Introduction

Anomaly detection allows us to find instances that deviate significantly from the majority of data, indicating e.g., a system fault. Usual unsupervised anomaly detection methods are purely data-driven and do not benefit from valuable expert knowledge. However, many of the anomalies that real-world data exhibits are irrelevant to the user as they represent atypical but normal events. For example, as illustrated in Fig. 1, in domestic hot water heat-pump systems, the water reaches abnormally high temperatures once in a while to kill potential Legionella bacteria; this is an atypical but normal event. Moreover, anomalies are often subjective and depend on the application purpose and what the user considers as abnormal. For example, an abnormal train delay, which is due to a passenger who blocked the door, is not interesting for a diagnosis purpose. However, it can be interesting for planning purposes.

Fig. 1. Data from a real heat-pump system, where the goal is to detect compressor failures. Several anomalies are irrelevant as they are not related to compressor failure. These are just atypical (but reasonable) events. Nevertheless, they appear as abnormal.

In order to distinguish between relevant and irrelevant anomalies, this paper proposes an interactive anomaly detection algorithm that proactively communicates with an expert user to leverage her/his feedback and learn to suggest more relevant anomalies. The objective here is two-fold: (i) maximizing the precision on the instances verified by the expert (i.e., ideally, only relevant anomalies are presented to the user), and (ii) minimizing the effort spent by the expert to verify these instances.

Recently, several methods such as AAD [10], OMD [12], and OJRank [13] have been proposed to incorporate user feedback into anomaly detectors, to achieve the objective (i). All these methods learn to combine anomaly scores from members of an ensemble of anomaly detectors (e.g., trees in an Isolation Forest [2]). The method proposed in this paper differs from the existing interactive anomaly detection methods in two ways. First, instead of only considering an ensemble of anomaly detectors, the proposed method aims to learn regions of the feature space where relevant anomalies are present. Second, existing methods focus on achieving a good precision (objective (i)) with less attention towards minimizing the expert effort required to investigate anomalies (objective (ii)). The energy and the time of the expert user are often limited, and every opportunity to interact with her/him should be fully utilized. Our proposed method aims

to achieve both objectives by learning to score anomalies differently in various regions of the feature space.

The remainder of this paper is organized as follows. In section 2, we present the related work and discuss the similarity and differences between the proposed method and the existing ones. In section 3, we formalize our goals and describe our proposed method. In section 4, we present the experimental evaluation where the proposed method is compared against state-of-the-art methods on several real-world datasets. In section 5, we conclude and discuss future work.

2 Related work

Since the work proposed in this paper involves interactions with a human expert, it is related to, but different from, *active learning* (which selects informative instances to be labeled, according to a query strategy). It is also closely related to interactive learning methods designed to learn from the *top-1 feedback* (which selects the top instance to be labeled, without a query strategy). In this section, we discuss how our work relates to and differs from state-of-the-art methods in these categories.

Active learning methods

Usual active learning (AL) techniques such as [3, 4] aim to minimize the labeling cost required to train a high-performance classification model. This is achieved by sampling a small number of informative instances (according to a query strategy) which are presented to an expert for labeling. We refer the reader to [5] for a survey of AL strategies for classification. AL techniques have also been used for anomaly and novelty detection in [6–8]. In all these AL methods, the goal is to minimize the final error of the model (after querying ends) on new unseen instances. This is in contrast to our proposed method, which aims to minimize the number of irrelevant anomalies presented to the expert during querying (i.e., while she/he is investigating them). In this case, each query is about the most anomalous yet-unlabeled instance.

In [9], a method for detecting errors in insurance claims was proposed. The method aims at reducing the expert effort required to verify the insurance claim errors. It assumes that a classifier has been trained to predict errors in claims and use it to score new unlabelled claims. The top-scoring claims are then clustered, and the clusters are ranked. Insurance claims from the top cluster are presented to the expert for investigation. Presenting instances from the same cluster avoids switching between contexts and therefore reduces the expert effort. The method moves to the next cluster when the precision for the current cluster falls below a threshold. However, this method does not update the model based on user feedback. In contrast, our proposed method incorporates each user-feedback so that the next suggested anomaly is more likely relevant.

Interactive learning based on the top-1 feedback

In contrast to the above-mentioned methods, there have been a limited number of interactive anomaly detection methods based on the top-1 feedback. Here the goal is to maximize the number of true/relevant anomalies presented to the expert user. These methods can be summarized according to the general process described in Algorithm 1. The method we propose in this paper falls under this category.

Algorithm 1: Interactive Anomaly Detection from the Top-1 Feedback.

Input: raw dataset, budget of b queries;
Initialize an anomaly detection model h ;
for $t \leftarrow 1$ **to** b **do**
 Rank instances in the descending order of their anomaly scores;
 Present the top-1 (most anomalous) instance x to the expert;
 Get a feedback label $y \in \{1, -1\}$ (i.e., TP or FP);
 Update the model h based (x, y) (or all instances labelled so far);
end

In [10], a method called AAD (Active Anomaly Detection) was proposed to maximize the number of correct anomalies presented to the user. At each iteration, the instance having the highest anomaly score is presented to the expert, and label feedback is obtained. The set of all instances labeled so far is used to solve an optimization problem that combines anomaly scores from an ensemble of anomaly detectors. AAD initially used an ensemble of one-dimensional histogram density estimators (LODA) [1], and was extended later to use a tree-based ensemble such as Isolation Forest [2]. Some drawbacks of ADD are: (i) the fact that it does not care about minimizing the expert effort, and (ii) as the number of labeled instances grows with each new feedback, the optimization problem takes more and more time to solve (i.e., it is not updated in an online fashion). Later, the authors of AAD suggested a method referred to as FSSN (feature space suppression network) [11]. The method uses an ensemble of global anomaly detectors and learns their local relevance to specific data instances, using a neural network trained on all labeled instances. FSSN improves the precision over AAD, but suffers from the same drawbacks as AAD.

Most recently, two methods OJRank [13] (On-the-Job learning to Re-rank anomalies) and OMD [12] (using Online Mirror Descent) have been proposed to learn to score anomalies from the top-1 feedback. Both learn to combine scores of an ensemble of anomaly detectors and optimize a loss function in an online fashion based on each received feedback. In OMD, a convex point-wise loss function is optimized using online mirror descent, while in OJRank, a pair-wise loss function is optimized using stochastic gradient descent. Both OMD and OJRank aim to maximize the number of correct anomalies presented to the expert. However, only OJRank emphasizes minimizing expert effort. Nonetheless, OMD depends

on way fewer hyper-parameters than OJRank (i.e., only one hyper-parameter representing the learning rate). This is an interesting criterion as there is usually no way to fine-tune the value of hyper-parameters when labeled data is scarce.

Based on these observations, we propose an extension of OMD (called OMD-Clustering), which splits the feature space into various regions (using several clusterings) and learns (online from each feedback) to score anomalies differently in these different regions of the feature space. We show that the proposed method improves the precision (i.e., detects more relevant anomalies within a budget) and significantly reduces the effort spent by the expert to verify these instances.

3 Proposed method

3.1 Preliminaries and goals

We are given an unlabeled dataset $X \in \mathbb{R}^{n \times d}$ of n instances in a d -dimensional space, as well as an unsupervised anomaly detection model A that scores instances according to their abnormality, i.e. $A(X) = \{s_1, s_2, \dots, s_n\}$. The proposed method can use any base anomaly detection model A , however, to be consistent with the methods presented in Section 2, Isolation Forest [2] is used as a base model in this paper.

We consider a limited budget of b , which is the number of instances the expert can verify (i.e., the total number of feedbacks). For example, this could correspond to the number of faulty systems that experts can diagnose within a period or the number of potentially erroneous invoices that could be manually analyzed within a day of work [13].

Following the general process presented in Algorithm 1, the proposed method proceeds iteratively. At each iteration $1 \dots b$, instances are scored, the instance with the highest anomaly score is presented to the expert for verification, expert feedback is obtained, and the model is updated based on the obtained feedback. The goal here is to update the model such that:

1. The **precision** at the given budget b is maximized: This corresponds to the total number of genuinely anomalous instances (i.e., relevant anomalies) verified by the expert within the budget b . Ideally, we want the expert to only verify relevant anomalies. This metric is defined as:

$$\text{precision@b} = \frac{TP_b}{a}, \quad (1)$$

where TP_b is the number of true positives (relevant anomalies) among the b verified instances, and a is the total number of true anomalies in the dataset (constant).

2. The overall **expert effort** is minimized: This corresponds to the cost or effort that the expert spends (due to switching context) when verifying consecutive instances. If consecutive instances presented to the expert are very different (resp. similar), she/he would switch context more often (resp. less often).

We use the same definition of *expert effort* as in [13]; this is the distance between consecutive instances presented to the expert:

$$\text{expert effort} = \sum_{t=1}^{b-1} (1 - \text{cosim}(x_t, x_{t+1})), \quad (2)$$

where $\text{cosim}(\cdot, \cdot)$ denotes the cosine similarity, and x_t denotes the instance presented to the expert at the t^{th} iteration.

3.2 OMD-Clustering

As shown in Fig. 1, instances that are anomalous for the same reason (e.g., compressor failure) are usually located in a similar region of the feature space. Therefore, instead of learning to combine anomaly scores given by members of the ensemble (i.e., trees of the Isolation Forest) such as in [10–13], the proposed method learns to score anomalies differently in different regions of the feature space. To do this, we first split the feature space into diverse and potentially overlapping regions, as illustrated in Fig. 2 (a). Such regions are obtained by applying clustering several times based on various numbers of clusters, various combinations of features, and initial clustering conditions. A total of m overlapping clusters resulting from the different clusterings are obtained.

Next, a sparse scores matrix $Z \in \mathbb{R}^{n \times m}$ is defined where the n rows correspond to the instances and the m columns correspond to the clusters (i.e., regions). Each entry $Z_{i,j}$ is set to s_i if x_i belongs to cluster c_j , otherwise to 0, i.e.,

$$Z_{i,j} = \begin{cases} s_i & \text{if } x_i \in c_j \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where x_i is the i^{th} instance in dataset X , s_i is the score assigned by the anomaly detection model A to instance x_i (i.e., $s_i = A(x_i)$), and c_j is the j^{th} cluster. The process of clustering and the construction of the Z scores matrix are described in Algorithm 2. Note that the Z matrix needs only to be constructed once at the beginning (i.e., before starting the interactions with the expert).

Now that $Z \in \mathbb{R}^{n \times m}$ is defined, the remaining of this section explains how to incorporate the expert feedback at each iteration of the interaction loop.

Let $\mathbf{1}$ be a vector of all ones (i.e. m ones). It is worth noting that $(\frac{1}{m}Z\mathbf{1}) \in \mathbb{R}^n$ corresponds exactly to the original anomaly scores we get with the unsupervised anomaly detector $A(X) = \{s_1, s_2, \dots, s_n\}$. By using $\frac{1}{m}Z\mathbf{1}$ to compute the final scores, the same weight (i.e., $\frac{1}{m}$) is being used for to all the m regions of the feature space. Instead of this, we propose to assign different weights to the m regions and define the final anomaly scores based on a weighted sum of the regions scores. That is, we replace the sum $\frac{1}{m}Z\mathbf{1}$ with

$$\left(\text{scores} = \frac{1}{m}Z\mathbf{w} \right) \in \mathbb{R}^n \quad (4)$$

Fig. 2. Illustration of the process of splitting the space into various regions and the construction of the sparse scores matrix $Z \in \mathbb{R}^{n \times m}$. The n rows correspond to the instances and the m columns correspond to the clusters. Each entry $Z_{i,j}$ is set to the anomaly score s_i if x_i belongs to cluster c_j , otherwise to 0.

Algorithm 2: Construction of the Z scores matrix.

Input: dataset X , anomaly scores $A(X) = \{s_1, \dots, s_n\}$, number of clusterings C to perform;
 $m \leftarrow 0$; // total number of clusters
for $i \leftarrow 1$ **to** C **do**
 Pick a random subset of p features, and a random number of clusters k ;
 Perform k -means clustering on the subset of X induced by the p features;
 $m \leftarrow m + k$;
end
Construct a sparse scores matrix $Z \in \mathbb{R}^{n \times m}$ according to eq. 3.;
return Z ;

and we learn $\mathbf{w} \in \mathbb{R}^m$ based on subsequent expert feedbacks. This formulation of the anomaly scores allows us to assign lower weights to the regions of the feature

space containing nominal instances or irrelevant anomalies, and higher weights to the regions containing true/relevant anomalies. As a result, successive anomalies presented to the expert will more likely be from the same truly anomalous region, hence minimizing the expert effort and increasing the precision at the given budget.

In order to learn $\mathbf{w}$, we use the same optimization procedure (online mirror descent) and we minimize the same simple linear loss function as in OMD [12]. Consider that the current iteration is t . Let x_t be the top-1 anomalous instance in X based on the anomaly scores given by the current weights $\mathbf{w}$ (according to eq. 4); let $y_t \in \{+1, -1\}$ be the feedback label provided by the expert for instance x_t (with $+1$ = relevant anomaly, and -1 = nominal or irrelevant anomaly); let z_t be row vector from Z corresponding to the instance x_t . Then, the (convex) loss function is simply defined as follows:

$$f(\mathbf{w}) = -y_t(z_t \cdot \mathbf{w}) \quad (5)$$

Note that when $y_t = +1$, the function in eq. 5 gives a smaller loss than if $y_t = -1$, since the instance presented to the expert (scored based on weights $\mathbf{w}$) was a relevant anomaly. Finally, the main interactive anomaly detection algorithm, which minimizes the loss based on each feedback, is provided in Algorithm 3.

Algorithm 3: OMD-Clustering

Input: dataset $X \in \mathbb{R}^{n \times d}$, model A , budget b , learning rate η ;

Construct the scores matrix $Z \in \mathbb{R}^{n \times m}$ using Algorithm 2;

$\theta \leftarrow \mathbf{1}$; (initialize weights to ones, $\theta \in \mathbb{R}^m$)

for $t \leftarrow 1$ **to** b **do**

$\mathbf{w} \leftarrow \arg \min_{\hat{\mathbf{w}} \in \mathbb{R}^+} \|\hat{\mathbf{w}} - \theta\|$; (constrains the weights to be positive)

 scores $\leftarrow \frac{1}{m} Z \cdot \mathbf{w}$; (scores $\in \mathbb{R}^n$, see eq. 4)

 Let x_t be the instance with the maximum score in scores;

 Get feedback $y_t \in \{+1, -1\}$ for x_t from the expert;

$X \leftarrow X - \{x_t\}$;

$\theta \leftarrow \theta - \eta \partial f(\mathbf{w})$; (take a gradient step to minimize the loss)

end

4 Evaluation

In this section, we assess the performance of the proposed method OMD-Clustering on several real-world datasets. We compare the proposed method to two state-of-the-art interactive anomaly detection methods OJRank [13] and OMD [12], as well as an unsupervised anomaly detector (Isolation Forest [2]) as a reference. The evaluation is done on a set of 13 datasets with available ground truth (true

label consisting of "nominal" and "anomaly"). Eleven of these are real-world outlier detection datasets from a publicly available repository [14], and two of them (toy and toy2) were artificially generated. Details of the datasets are summarized in Table 1. The performance of the methods is evaluated in terms of:

- The precision at a given budget (precision@b) as described in eq. 1. In this case, we produce curves that show how the precision of the different methods changes according to various values of the budget b .
- The expert effort as described in eq. 2. Here, we also produce curves to show how the expert effort changes according to various values of the budget b .
- The area under the precision and the expert effort curves (resp. AUC_{prec} and AUC_{effort}).

In all experiments, we use the hyper-parameter values recommended (for the same datasets) in the original papers of OJRank [13] and OMD [12]. For the proposed method, the number of clusterings (used in Algorithm 2) is set to $C = 30$ for all datasets.

Name	Size	Nbr. Features	Nbr. Anomalies.	Anomalies %
Abalone	1920	9	29	1.51
ann_thyroid_lv3	3251	21	73	2.25
cardiotocography	1700	21	45	2.65
covtype_sub	2000	54	19	0.95
kddcup_sub	2000	91	77	3.85
Mammography	11183	6	260	2.32
Mammography_sub	2000	6	46	2.30
shuttle	12345	9	867	7.02
shuttle_sub	2000	9	140	7.00
weather	13117	8	656	5.00
yeast	1191	8	55	4.62
toy	485	2	20	4.12
toy2	485	2	35	7.22

Table 1. Details of the benchmark datasets

The precision and expert effort curves obtained by the four methods (OJRank, OMD, OMD-Clustering, and Unsupervised) are shown in Fig. 4 (a-b) and Fig. 5 (a-c). Fig. 4 (a-b) shows the results of the two artificial datasets illustrated in Fig. 3, while Fig. 5 (a-c) shows the results on three of the real-world datasets (cardiotocography, abalone, and mammography). As one expects, we can see from these figures that, in general, all the interactive anomaly detection methods achieve a higher precision at budget compared to the unsupervised anomaly detection method, which confirms that interacting with the expert help to get more relevant anomalies. The proposed method (OMD-Clustering) achieves a precision which is higher than or equal to the other interactive methods, while always resulting in a significantly smaller expert effort. This indicates that clus-

tering helps to aggregate similar instances from which relevant anomalies can be more easily detected.

Fig. 3. Simple artificial datasets: toy (with a single anomalous region), and toy2 (with two different anomalous regions).

The results on all the remaining datasets are summarized more compactly in Fig. 6 (a-b). Fig. 6 (a) shows bars plots corresponding to the area under the precision curves (AUC_{prec}). Fig. 6 (b) shows bars plots corresponding to the area under the expert effort curves (AUC_{prec}). The same observations can be made from these figures. Once again, we can see that all the interactive methods outperform the unsupervised one, highlighting the importance of interacting with a human expert. Moreover, the proposed OMD-Clustering method usually achieves a higher precision with a lower effort than the other methods on most of the datasets.

5 Conclusion and future work

In this paper, we developed an interactive anomaly detection method, where a human expert can provide feedback while verifying/investigating anomalies. The proposed method incorporates each feedback in an online fashion and learns to assign weights to various regions of the feature space. As a result, regions of the feature space with irrelevant anomalies would contribute less to the final anomaly score, while regions with more relevant anomalies would contribute more. The proposed method was evaluated on various real-world datasets and compared to state-of-the-art interactive anomaly detection methods. The results show that the proposed method is more precise at detecting relevant anomalies within a budget, while at the same time, reducing the expert effort significantly.

The existing interactive anomaly detection methods (including the one proposed in this paper) require a fixed dataset and re-rank all instances after each incorporated feedback. As future work, it would be interesting to investigate how such interactive methods can be extended to a streaming setting where data is continuously arriving and need to be processed as soon as it is available.

Fig. 4. Evaluation of the precision and expert effort according to various values of the budget, on the two synthetic datasets: toy and toy2.

References

1. Tomas Pevny. "Loda: Lightweight on-line detector of anomalies." Machine Learning 102, no. 2 (2016): 275-304.
2. Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest." In 2008 Eighth IEEE International Conference on Data Mining, pp. 413-422. IEEE, 2008.
3. Mohamed-Rafik Bouguelia, Slawomir Nowaczyk, K. C. Santosh, and Antanas Verikas. "Agreeing to disagree: Active learning with noisy labels without crowdsourcing." International Journal of Machine Learning and Cybernetics 9, no. 8 (2018): 1307-1319.
4. Mohamed-Rafik Bouguelia, Yolande Belaid, and Abdel Belaid. "An adaptive streaming active learning strategy based on instance weighting." Pattern Recognition Letters 70 (2016): 38-44.
5. Burr Settles. "Active Learning, volume 6 of Synthesis Lectures on Artificial Intelligence and Machine Learning." Morgan & Claypool (2012).

Fig. 5. Evaluation of the precision and expert effort according to various values of the budget, on three real-world datasets: cardiotocography, abalone, and mammography.

6. Nico Görnitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld. "Toward supervised anomaly detection." *Journal of Artificial Intelligence Research* 46 (2013): 235-262.

(a) Area under the precision curve (AUC_{prec}) for all methods on each dataset. (b) Area under the expert effort curve (AUC_{effort}) for all methods on each dataset.**Fig. 6.** Overall evaluation results: area under the precision and the expert effort curves for all methods and datasets.

7. Nir Nissim, Aviad Cohen, Robert Moskovitch, Assaf Shabtai, Mattan Edry, Oren Bar-Ad, and Yuval Elovici. "Alpd: Active learning framework for enhancing the detection of malicious pdf files." In 2014 IEEE Joint Intelligence and Security Informatics Conference, pp. 91-98. IEEE, 2014.

8. Pelleg, Dan, and Andrew W. Moore. "Active learning for anomaly and rare-category detection." In *Advances in neural information processing systems*, pp. 1073-1080. 2005.
9. Rayid Ghani, and Mohit Kumar. "Interactive learning for efficiently detecting errors in insurance claims." In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 325-333. 2011.
10. Shubhomoy Das, Weng-Keen Wong, Thomas Dietterich, Alan Fern, and Andrew Emmott. "Incorporating expert feedback into active anomaly discovery." In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 853-858. IEEE, 2016.
11. Shubhomoy Das, and Janardhan Rao Doppa. "GLAD: GLocalized Anomaly Detection via Active Feature Space Suppression." *arXiv preprint arXiv:1810.01403* (2018).
12. Md Amran Siddiqui, Alan Fern, Thomas G. Dietterich, Ryan Wright, Alec Theriault, and David W. Archer. "Feedback-guided anomaly discovery via online optimization." In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2200-2209. 2018.
13. Hemank Lamba, and Leman Akoglu. "Learning on-the-job to re-rank anomalies from top-1 feedback." In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pp. 612-620. Society for Industrial and Applied Mathematics, 2019.
14. Shebuti Rayana: ODDS Library, <http://odds.cs.stonybrook.edu>, Stony Brook University, Department of Computer Sciences, 2016