2018

Precision and Recall for Range-Based Anomaly Detection

Tae J. Lee
Microsoft

Justin E. Gottschlich
Intel, gojustin@cis.upenn.edu

Nesime Tatbul
Intel

Eric Metcalf
Brown University

Stan Zdonik
Brown University

Precision and Recall for Range-Based Anomaly Detection

Abstract

Classical anomaly detection is principally concerned with point- based anomalies, anomalies that occur at a single data point. In this paper, we present a new mathematical model to express range- based anomalies, anomalies that occur over a range (or period) of time.

Precision and Recall for Range-Based Anomaly Detection

Tae Jun Lee*
Microsoft

Justin Gottschlich, Nesime Tatbul
Intel Labs

Eric Metcalf, Stan Zdonik
Brown University

ABSTRACT

Classical anomaly detection is principally concerned with *point-based anomalies*, anomalies that occur at a single data point. In this paper, we present a new mathematical model to express *range-based anomalies*, anomalies that occur over a range (or period) of time.

1 INTRODUCTION

Anomaly detection (AD) seeks to identify atypical events. Anomalies tend to be domain or problem specific, and many occur over a period of time. We refer to such events as *range-based anomalies*, as they occur over a range (or period) of time[1]. Therefore, it is critical that the accuracy measures for anomalies, and the systems detecting them, capture events that occur over a range of time. Unfortunately, classical metrics for anomaly detection were designed to handle only fixed-point anomalies [1]. An AD algorithm behaves much like a pattern recognition and binary classification algorithm: it recognizes certain patterns in its input and classifies them as either normal or anomalous. For this class of algorithms, *Recall* and *Precision* are widely used for evaluating the accuracy of the result. They are formally defined as in Equations 1 and 2, where *TP* denotes true positives, *FP* denotes false positives, and *FN* denotes false negatives.

$$Recall = TP \div (TP + FN) \tag{1}$$

$$Precision = TP \div (TP + FP) \tag{2}$$

While useful for point-based anomalies, classical recall and precision suffer from their inability to capture, and bias, classification correctness for domain-specific time-series anomalies. Because of this, many time-series AD systems' accuracy are being misrepresented, as point-based recall and precision are used to measure their effectiveness [9]. Furthermore, the need to accurately identify time-series anomalies is growing due to the explosion of streaming and real-time systems [2, 4, 7, 8, 10]. To address this, we redefine recall and precision to encompass range-based anomalies. Unlike prior work [2, 6], our mathematical definitions are a superset of the classical definitions, enabling our system to subsume point-based anomalies. Moreover, our system is broadly generalizable, providing specialization functions to control a domain's bias along a multi-dimensional axis that is necessary to accommodate the needs of specific domains.

In this short paper, we present novel formal definitions of recall and precision for range-based anomaly detection that both subsume those formerly defined for point-based anomaly detection as well as being customizable to a rich set of application domains. Empirical data has been omitted to meet the venue's compressed format.

*The work was done while a Brown student.

[1]Range-based anomalies are a specific type of collective anomalies [3]. Moreover, range-based anomalies are similar, but not identical, to sequence anomalies [11].

SysML'18, February 2018, Stanford, CA, USA
2018.

Table 1: Notation

Notation	Description
R	set of real anomaly ranges
R_i	the i^{th} real anomaly range
P	set of predicted anomaly ranges
P_j	the j^{th} predicted anomaly range
N_r	number of real anomaly ranges
N_p	number of predicted anomaly ranges
α	relative weight of existence reward
β	relative weight of overlap reward
$\gamma()$	overlap cardinality function
$\omega()$	overlap size function
$\delta()$	positional bias function

2 RANGE-BASED RECALL

Classical *Recall* rewards an AD system when anomalies are successfully identified (i.e., TP) and penalizes it when they are not (i.e., FN). It is computed by counting the number of anomalous points successfully predicted and then dividing that number by the total number of anomalous points. However, it is not sensitive to domains where a single anomaly can be represented as a range of contiguous points. In this section, we propose a new way to compute recall for such range-based anomalies. Table 1 summarizes our notation.

Given a set of real anomaly ranges $R = \{R_1, .., R_{N_r}\}$ and a set of predicted anomaly ranges $P = \{P_1, .., P_{N_p}\}$, our $Recall_T(R, P)$ formulation iterates over the set of all real anomaly ranges (R), computing a recall score for each real anomaly range ($R_i \in R$) and adding them up into a total recall score. This total score is then divided by the total number of real anomalies (N_r) to obtain an average recall score for the whole time-series.

$$Recall_T(R, P) = \frac{\sum_{i=1}^{N_r} Recall_T(R_i, P)}{N_r} \tag{3}$$

When computing the recall score $Recall_T(R_i, P)$ for a single real anomaly range R_i, we take the following aspects into account:

- *Existence*: Identifying an anomaly (even by a single point in R_i) may be valuable in some application domains.
- *Size*: The larger the size of the correctly predicted portion of R_i, the higher the recall score will likely be.
- *Position*: In some cases, not only size, but also the relative position of the correctly predicted portion of R_i may be important to the application (e.g., early and late biases).
- *Cardinality*: Detecting R_i with a single predicted anomaly range $P_j \in P$ may be more valuable to an application than doing so with multiple different ranges in P.

We capture these aspects as a sum of two reward terms weighted by α and β, respectively, where $0 \leq \alpha, \beta \leq 1$ and $\alpha + \beta = 1$. α represents the relative importance of rewarding *existence*, whereas β represents the relative importance of rewarding *size*, *position*, and

```
function ω(AnomalyRange, OverlapSet, δ)
    MyValue ← 0
    MaxValue ← 0
    AnomalyLength ← length(AnomalyRange)
    for i ← 1, AnomalyLength do
        Bias ← δ(i, AnomalyLength)
        MaxValue ← MaxValue + Bias
        if AnomalyRange[i] in OverlapSet then
            MyValue ← MyValue + Bias
    return MyValue/MaxValue
```

(a) Overlap Size

```
// Flat positional bias
function δ(i, AnomalyLength)
    return 1

// Front-end positional bias
function δ(i, AnomalyLength)
    return AnomalyLength - i + 1

// Tail-end positional bias
function δ(i, AnomalyLength)
    return i
```

(b) Positional Bias

Figure 1: Example Functions for $\omega()$ and $\delta()$

cardinality, which all stem from the overlap between R_i and the set of all predicted anomaly ranges ($P_i \in P$).

$$Recall_T(R_i, P) = \alpha \times ExistenceReward(R_i, P) + \beta \times OverlapReward(R_i, P) \quad (4)$$

If anomaly range R_i is identified (i.e., $|R_i \cap P_j| \geq 1$ across all $P_j \in P$), then an existence reward of 1 is earned.

$$ExistenceReward(R_i, P) = \begin{cases} 1 & \text{, if } \sum_{j=1}^{N_p} |R_i \cap P_j| \geq 1 \\ 0 & \text{, otherwise} \end{cases} \quad (5)$$

Additionally, an overlap reward, dependent upon three application-defined functions $0 \leq \gamma() \leq 1$, $0 \leq \omega() \leq 1$, and $\delta() \geq 1$, can be earned. These functions capture the *cardinality* (γ), *size* (ω), and *position* (δ) of the overlap. The cardinality term serves as a scaling factor for the rewards earned from size and position of the overlap.

$$OverlapReward(R_i, P) = CardinalityFactor(R_i, P) \times \sum_{j=1}^{N_p} \omega(R_i, R_i \cap P_j, \delta) \quad (6)$$

The cardinality factor is largest (i.e., 1), when R_i overlaps with at most one predicted anomaly range (i.e., it is identified by a single prediction range). Otherwise, it receives a value $0 \leq \gamma() \leq 1$ defined by the application.

$$CardinalityFactor(R_i, P) = \begin{cases} 1 & \text{, if } R_i \text{ overlaps with at most one } P_j \in P \\ \gamma(R_i, P) & \text{, otherwise} \end{cases} \quad (7)$$

The $Recall_T$ constants (α and β) and functions ($\gamma()$, $\omega()$, and $\delta()$) are tunable according to the needs of the application. Next, we illustrate how they can be customized with examples.

The cardinality factor should generally be inversely proportional to $Card(R_i)$, i.e., the number of distinct prediction ranges that a real anomaly range R_i overlaps. For example, $\gamma(R_i, P)$ can simply be set to $1/Card(R_i)$.

Figure 1a provides an example for the $\omega()$ function for size, which can be used with many different $\delta()$ functions for positional bias as shown in Figure 1b. If all index positions are equally important, then the flat bias function should be used. If earlier ones are more important than later ones (e.g., early cancer detection [5], real-time apps [2]), then the front-end bias function should be used. Finally, if later index positions are more important (e.g., delayed response in robotic defense), then the tail-end bias function should be used.

Our recall formula for range-based anomalies subsumes the classical one for point-based anomalies (i.e., $Recall_T \equiv Recall$) when:

(i) all $R_i \in R$ and $P_j \in P$ are represented as single-point ranges (e.g., range $[1, 3]$ represented as $[1, 1], [2, 2], [3, 3]$), and
(ii) $\alpha = 0, \beta = 1, \gamma() = 1$, $\omega()$ is as in Figure 1a, and $\delta()$ returns flat positional bias as in Figure 1b.

3 RANGE-BASED PRECISION

Classical *Precision* is computed by counting the number of successful prediction points (i.e., TP) in proportion to the total number of prediction points (i.e., TP+FP). The key difference between *Precision* and *Recall* is that *Precision* penalizes FPs. In this section, we extend classical precision to handle range-based anomalies. Our formulation follows a similar structure as $Recall_T$.

Given a set of real anomaly ranges $R = \{R_1, .., R_{N_r}\}$ and a set of predicted anomaly ranges $P = \{P_1, .., P_{N_p}\}$, $Precision_T(R, P)$ iterates over the set of predicted anomaly ranges (P), computing a precision score for each range ($P_i \in P$) and then sums them. This sum is then divided by the total number of predicted anomalies (N_p), averaging the score for the whole time-series.

$$Precision_T(R, P) = \frac{\sum_{i=1}^{N_p} Precision_T(R, P_i)}{N_p} \quad (8)$$

When computing $Precision_T(R, P_i)$ for a single predicted anomaly range P_i, there is no need for an *existence* reward, because precision by definition emphasizes prediction quality, and existence by itself is too low a bar for judging the quality of a prediction. This removes the need for α and β constants. Therefore:

$$Precision_T(R, P_i) = CardinalityFactor(P_i, R) \times \sum_{j=1}^{N_r} \omega(P_i, P_i \cap R_j, \delta) \quad (9)$$

$\gamma()$, $\omega()$, and $\delta()$ are customizable as before. Furthermore, $Precision_T \equiv Precision$ under the same settings as in Section 2 (except α and β are not needed). Note that, while $\delta()$ provides a potential knob for positional bias, we believe that in many domains a flat bias function will suffice for $Precision_T$, as an FP is typically considered uniformly bad wherever it appears in a prediction range.

4 CONCLUSION

In this paper, we note that traditional recall and precision were invented for point-based analysis. In range-based anomaly detection, anomalies are not necessarily single points, but are, in many cases, ranges. In response, we offered new recall and precision definitions that take ranges into account.

Acknowledgments. This research has been funded in part by Intel.

REFERENCES

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

[2] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. 2017. Unsupervised Real-time Anomaly Detection for Streaming Data. *Neurocomputing* 262 (2017), 134–147.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly Detection: A Survey. *ACM Computing Surveys* 41, 3 (2009), 15:1–15:58.

[4] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. 2016. Robust Random Cut Forest Based Anomaly Detection on Streams. In *International Conference on Machine Learning (ICML)*. 2712–2721.

[5] Konstantina Kourou, Themis P. Exarchos, Konstantinos P. Exarchos, Michalis V. Karamouzis, and Dimitrios I. Fotiadis. 2015. Machine Learning Applications in Cancer Prognosis and Prediction. *Computational and Structural Biotechnology Journal* 13 (2015), 8–17.

[6] Alexander Lavin and Subutai Ahmad. 2015. Evaluating Real-Time Anomaly Detection Algorithms - The Numenta Anomaly Benchmark. In *IEEE International Conference on Machine Learning and Applications (ICMLA)*. 38–44.

[7] Tae Jun Lee, Justin Gottschlich, Nesime Tatbul, Eric Metcalf, and Stan Zdonik. 2018. Greenhouse: A Zero-Positive Machine Learning System for Time-Series Anomaly Detection. https://arxiv.org/abs/1801.03168/. In *SysML Conference*.

[8] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. 89–94.

[9] Nidhi Singh and Craig Olinsky. 2017. Demystifying Numenta Anomaly Benchmark. In *International Joint Conference on Neural Networks (IJCNN)*. 1570–1577.

[10] Twitter. 2015. AnomalyDetection R Package. https://github.com/twitter/AnomalyDetection/. (2015).

[11] Christina Warrender, Stephanie Forrest, and Barak Pearlmutter. 1999. Detecting Intrusions using System Calls: Alternative Data Models. In *IEEE Symposium on Security and Privacy*. 133–145.