S. Guha, N. Mishra, G. Roy, and O. Schrijvers. Robust Random Cut Forest Based Anomaly Detection On Streams. In ICML, 2016.

Jungtaek Kim

Department of Computer Science and Engineering,
Pohang University of Science and Technology,
Pohang 37673, Republic of Korea

Sep 06, 2016

Table of Contents

Motivation

- Most anomaly detections are unsupervised learning tasks.
- Also, dynamic data streams are often handled.
- We can raise two main questions about anomalies.
  - How do we define anomalies?
  - What data structure do we use to efficiently detect anomalies over dynamic data streams?

Robust Random Cut Tree (RRCT)

Definition 1 *A* **robust random cut tree** *(RRCT) on point set S is generated as follows:*

1. *Choose a random dimension proportional to* $\frac{\ell_i}{\sum_j \ell_j}$, *where* $\ell_i = \max_{x \in S} x_i - min_{x \in S} x_i$.
2. *Choose* $X_i \sim Uniform[\min_{x \in S} x_i, \max_{x \in S} x_i]$
3. *Let* $S_1 = \{x | x \in S, x_i \leq X_i\}$ *and* $S_2 = S \setminus S_1$ *and recurse on* S_1 *and* S_2.

- It divides a space based on a probabilistic approach.

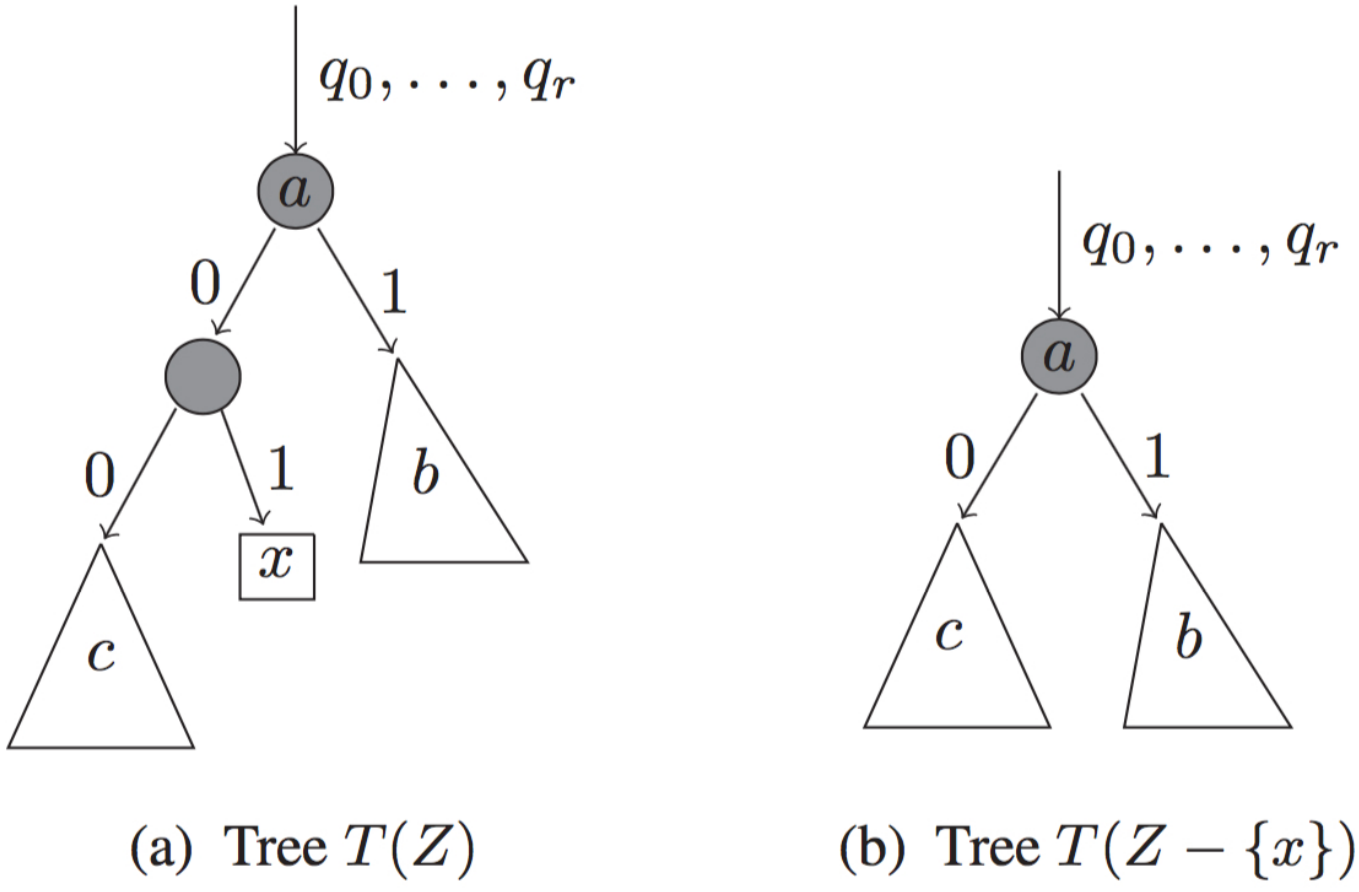
Determining a Dimension to Split

Theorem 1 *Consider the algorithm in Definition 1. Let the weight of a node in a tree be the corresponding sum of dimensions* $\sum_i \ell_i$. *Given two points* $u, v \in S$, *define the tree distance between* u *and* v *to be the weight of the least common ancestor of* u, v. *Then the tree distance is always at least the Manhattan distance* $L_1(u, v)$, *and in expectation, at most* $O\left(d \log \frac{|S|}{L_1(u,v)}\right)$ *times* $L_1(u, v)$.

Maintaining a Tree

Theorem 2 (Section 3) *Given a tree T drawn according to $\mathcal{T}(S)$; if we delete the node containing the isolated point x and its parent (adjusting the grandparent accordingly, see Figure 2), then the resulting tree T' has the same probability as if being drawn from $\mathcal{T}(S - \{x\})$. Likewise, we can produce a tree T'' as if drawn at random from $\mathcal{T}(S \cup \{x\})$ is time which is $O(d)$ times the maximum depth of T, which is typically sublinear in $|T|$.*

Maintaining a Tree



(a) Tree $T(Z)$ (b) Tree $T(Z - \{x\})$

Figure 2. A correspondence of trees

Maintaining a Tree

Definition 2 *Define the bit-displacement or displacement of a point x to be the increase in the model complexity of all other points, i.e., for a set Z, to capture the externality introduced by x, define, where $T' = T(Z - \{x\})$,*

$$\textsc{Disp}(x, Z) = \sum_{T, y \in Z - \{x\}} \mathbb{P}r\left[T\right] \left(f(y, Z, T) - f(y, Z - \{x\}, T') \right)$$

- If displacement is large, that point increases the complexity.

Maintaining a Tree

Lemma 3 *Given point p and set of points S with an axis parallel minimal bounding box $B(S)$ such that $p \notin B$:*

(i) For any dimension i, the probability of choosing an axis parallel cut in a dimension i that splits S using the weighted isolation forest algorithm is exactly the same as the conditional probability of choosing an axis parallel cut that splits $S \cup \{p\}$ in dimension i, conditioned on not isolating p from all points of S.

(ii) Given a random tree of $RRCF(S \cup \{p\})$, conditioned on the fact the first cut isolates p from all points of S, the remainder of the tree is a random tree in $RRCF(S)$.

Duplicate Resilience

Definition 3 *The* **Collusive Displacement** *of x denoted by* $\textsc{CoDisp}(x, Z, |S|)$ *of a point x is defined as*

$$\mathop{\mathbb{E}}_{S \subseteq Z, T} \left[\max_{x \in C \subseteq S} \frac{1}{|C|} \sum_{y \in S - C} \Big(f(y, S, T) - f(y, S - C, T'') \Big) \right]$$

- A colluder set can be deleted at once.

Algorithm ForgetPoint

Algorithm 1 Algorithm ForgetPoint.

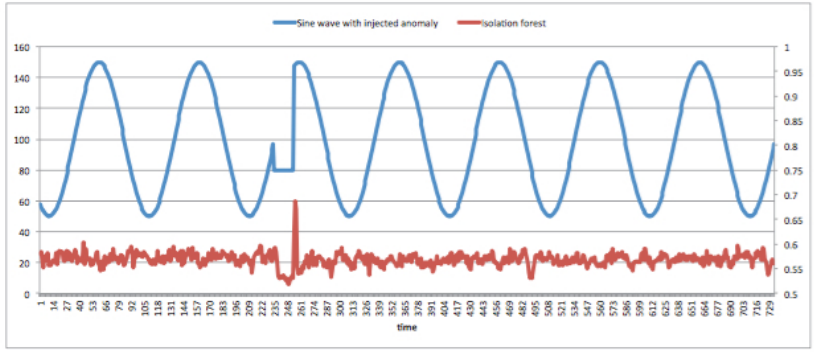
1: Find the node v in the tree where p is isolated in T.
2: Let u be the sibling of v. Delete the parent of v (and of u) and replace that parent with u (i.e., we short circuit the path from u to the root).
3: Update all bounding boxes starting from u's (new) parent upwards – this state is not necessary for deletions, but is useful for insertions.
4: Return the modified tree T'.

Algorithm InsertPoint

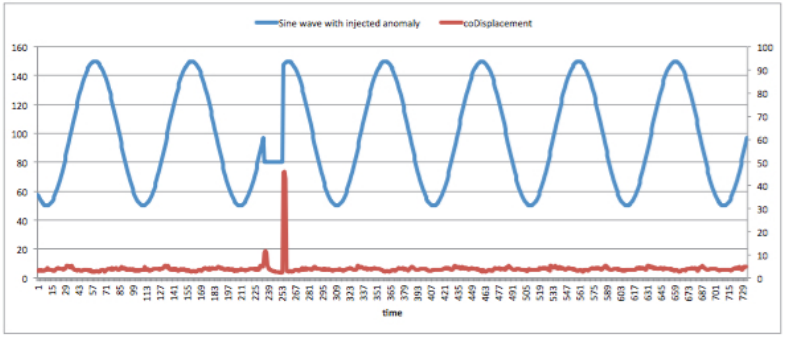
Algorithm 2 Algorithm InsertPoint.

1: We have a set of points S' and a tree $T(S')$. We want to insert p and produce tree $T'(S' \cup \{p\}$.
2: If $S' = \emptyset$ then we return a node containing the single node p.
3: Otherwise S' has a bounding box $B(S') = [x_1^\ell, x_1^h] \times [x_2^\ell, x_2^h] \times \cdots [x_d^\ell, x_d^h]$. Let $x_i^\ell \leq x_i^h$ for all i.
4: For all i let $\hat{x}_i^\ell = \min\{p_i, x_i^\ell\}$ and $\hat{x}_i^h = \max\{x_i^h, p_i\}$.
5: Choose a random number $r \in [0, \sum_i (\hat{x}_i^h - \hat{x}_i^\ell)]$.
6: This r corresponds to a specific choice of a cut in the construction of $RRCF(S' \cup \{p\})$. For instance we can compute $\arg\min\{j | \sum_{i=1}^{j} (\hat{x}_i^h - \hat{x}_i^\ell) \geq r\}$ and the cut corresponds to choosing $\hat{x}_j^\ell + \sum_{i=1}^{j} (\hat{x}_i^h - \hat{x}_i^\ell) - r$ in dimension j.
7: If this cut separates S' and p (i.e., is not in the interval $[x_j^\ell, x_j^h]$) then and we can use this as the first cut for $T'(S' \cup \{p\})$. We create a node – one side of the cut is p and the other side of the node is the tree $T(S')$.
8: If this cut does not separate S' and p then we throw away the cut! We choose the exact same dimension as $T(S')$ in $T'(S' \cup \{p\})$ and the exact same value of the cut chosen by $T(S')$ and perform the split. The point p goes to one of the sides, say with subset S''. We repeat this procedure with a smaller bounding box $B(S'')$ of S''. For the other side we use the same subtree as in $T(S')$.
9: In either case we update the bounding box of T'.

Synthetic Data: Experiments



(a) The bottom red curve reflects the anomaly score produced by IF. Note that the start of the anomaly is missed.



(b) The bottom red curve represents the anomaly score produced by RRCF. Both the beginning and end of the anomaly are caught.

Figure 4. The top blue curve represents a sine wave with an artificially injected anomaly. The bottom red curve shows the anomaly score over time.

NYC Taxicabs: Experiments

Table 1. Comparison of Baseline Isolation Forest to proposed Robust Random Cut Forest

Method	Sample Size	Positive Precision	Positive Recall	Negative Precision	Negative Recall	Accuracy	AUC
IF	256	0.42 (0.05)	0.37 (0.02)	0.96 (0.00)	0.97 (0.01)	0.93 (0.01)	0.83 (0.01)
RRCF	256	0.87 (0.02)	0.44 (0.04)	0.97 (0.00)	1.00 (0.00)	0.96 (0.00)	0.86 (0.00)
IF	512	0.48 (0.05)	0.37 (0.01)	0.97 (0.01)	0.96 (0.00)	0.94 (0.00)	0.86 (0.00)
RRCF	512	0.84 (0.04)	0.50 (0.03)	0.99 (0.00)	0.97 (0.00)	0.96 (0.00)	0.89 (0.00)
IF	1024	0.51 (0.03)	0.37 (0.01)	0.96 (0.00)	0.98 (0.00)	0.94 (0.00)	0.87 (0.00)
RRCF	1024	0.77 (0.03)	0.57 (0.02)	0.97 (0.00)	0.99 (0.00)	0.96 (0.00)	0.90 (0.00)

Method	Segment Precision	Segment Recall	Time to Detect Onset	Time to Detect End	Prec@5	Prec@10	Prec@15	Prec@20
IF	0.40 (0.09)	0.80 (0.09)	22.68 (3.05)	23.30 (1.54)	0.52 (0.10)	0.50 (0.00)	0.34 (0.02)	0.28 (0.03)
RRCF	0.65 (0.14)	0.80 (0.00)	13.53 (2.05)	10.85 (3.89)	0.58 (0.06)	0.49 (0.03)	0.39 (0.02)	0.30 (0.00)

Table 2. Segment-Level Metrics and Precision@K

Q & A