Isolation Forest - 2021년 P-sat 논문스터디 3회차

김원구, 김재희, 오정민

2021년 2월 2일

'Isolation Forest' 논문을 바탕으로 Isolation Forest에 대한 전반적인 내용을 소개하는 자료입니다.

1. Introduction

Anomaly detection은 대다수의 정상 데이터들과 다른 양상을 보이는 희귀한 케이스를 탐지하는 걸 목표로 한다.

1.1 기존 Anomaly detection의 문제점

Anomaly detection의 기존 방법들은 대부분 정상 범주를 정의한 다음, 정상 범주에 들어가지 않으면 anomaly라고 판단하는 식으로 작동했다.(classification-based나 clustering-based 등)

이러한 접근법에는 크게 다음과 같은 두가지 문제점이 있다.

- 1. 모형이 normal 포인트에 대해서 최적화되어 있기 때문에 anomaly detection 성능은 떨어진다는 점.(정상 데이터들을 anomalies로 잘못 분류하거나 anomalies를 너무 적게 찾아내는 등)
- 2. 기존의 모델들은 high computational complexity로 인해 대용량 데이터이나 고차원 데이터에는 적용하기 어렵다는 점

1.2 Isolation Forest 접근

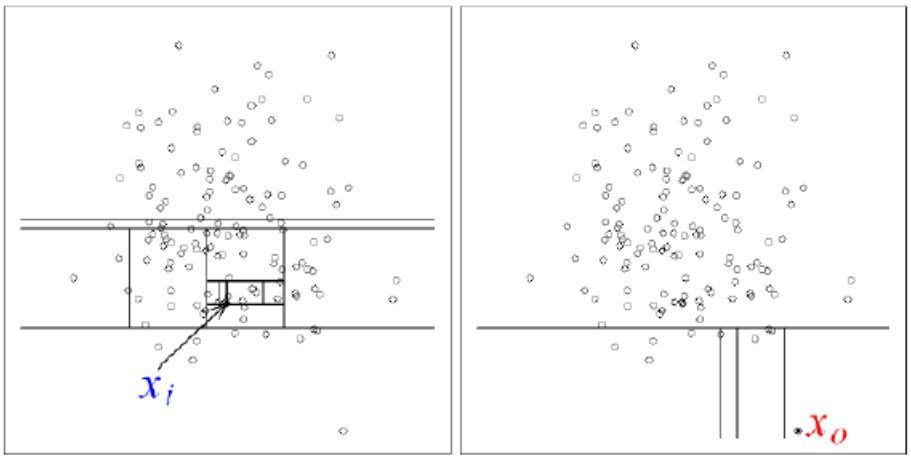
이러한 문제점들을 극복하기 위해서 Isolation Forest는 정상 범주를 정의하는 방식이 아닌 이상치를 기준(이상치를 isolate하는)으로 모델을 생성하는 방법론을 제시한다.

Isolation Forest는 anomaly에 대해서 다음과 같은 2가지 성질을 가정한다.

- 1. 전체 데이터에서 anomaly가 차지하는 비율이 작다.
- 2. anomaly는 정상 데이터 포인트들과 다른 데이터 특성을 가지고 있다.

요약하자면 'few & different'라고 생각하면 된다.

위에서 언급한 anomaly의 'few & different' 특징에 기반하여, Tree 구조는 모든 single 포인트들을 isolate할 수 있는 효과적인 구조다. isolate는 '1개의 데이터 포인트를 나머지 point들과 구별하는 것'을 의미한다. 이상 데이터들은 Tree의 깊이는 얕을 것이고, 정상 데이터들은 Tree 깊이가 깊을 것이다. Tree의 깊이가 얕다는 것은 결국 더 쉽게 구분할 수 있다는 의미가 되는데 다음의 그림을 통해 살펴보도록 하자.

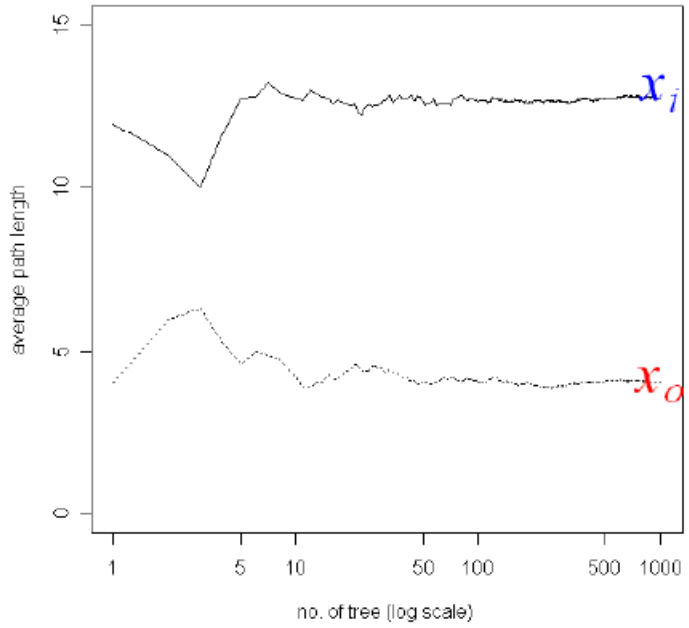


(a) Isolating x_i (b) Isolating x_o

feature space에 데이터 포인트들이 아래의 그림과 같이 존재한다고 생각해보자. 이때, 빨간색으로 표시된 x_0는 anomaly 포인트이고 파란색으로 표시된 x_i는 normal 포인트이다. 해당 그림을 보면 알 수 있듯이 normal 포인들이 anomaly 데이터보다 고립시키는데 더 많은 partition이 필요함을 알 수 있다.

여기서 partition은 임의의 변수와 임의의 값(최댓값과 최솟값 사이에 존재하는)으로 진행되었다.

partition이 random하게 진행되기 때문에 개별 tree에서 구해진 partition 값들이 같지는 않다. 그래서 우리는 여러개의 tree로부터 평균적인 path length를 구하게 되고, 아래 그림을 보면 알 수 있듯이 그림을 보면 알 수 있듯이 x_0, x_i에 대한 평균적인 path length가 tree의 개수가 증가함에 따라 수렴한다는 것을 확인할수 있다.

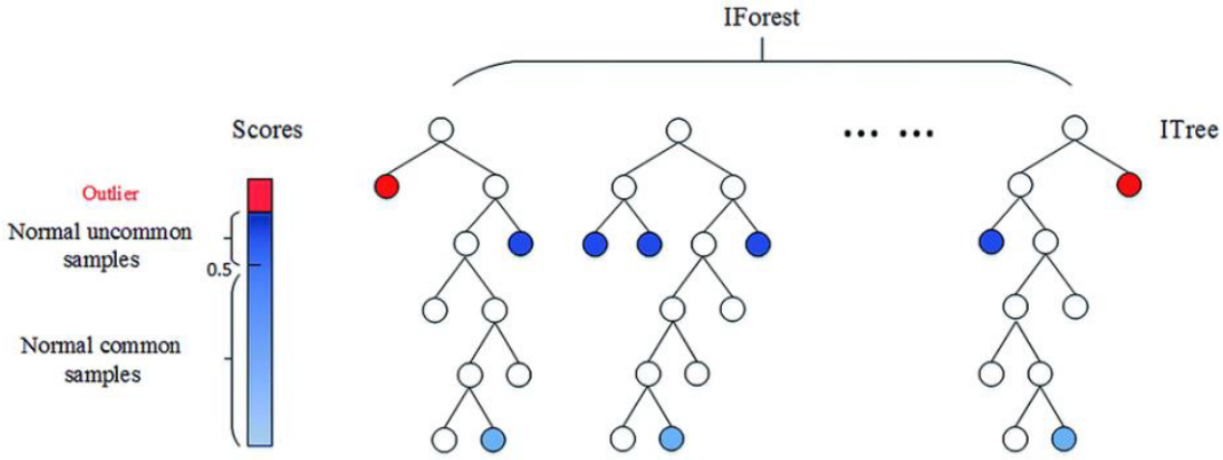


(c) Average path lengths converge

그림을 보면, x_i는 12.82로, x_0은 4.02로 수렴함을 확인 할 수 있다.

1.3 Isolation Forest 형태

위에서 언급했던 내용을 바탕으로 생각해보면 결국 여러개의 Tree(Isolation Tree)들을 만들게 되고 이를 앙상블하여 만든것이 바로 Isolation Forest이다. 전반적인 형태는 다음과 같다.



2. 용어 정리

2.1 Isolation Tree

통칭 iTree라고 표현하며, 각 노드에서 0개 혹은 2개의 자식노드를 가지는 이진트리 형식이다. 주어진 샘플 데이터 X에 대하여, 이 데이터 셋 X를 다음과 같은 조건을 만족할 때까지 랜덤하게 선택한 변수 q와 split value p로 재귀적으로 나누는 과정을 포함하고 있다.

- 1. 제한된 트리 깊이의 한계에 도달 했을 때 → 계산 제한
- 2. 데이터가 한개만 남았을 때
- 3. 같은 값들을 가지는 데이터들만 남았을 때 → 한계적

2.2 Path length

$h(x)$로 표현하며, 포인트 x가 root node에서 terminal node로 split된 횟수 (iTree에서 isolation 되는데 사용한 split 횟수)

2.3 Anomaly score

anomaly score는 다음과 같이 표현한다.

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

- $E(h(x))$는 average path length of x를 의미한다.

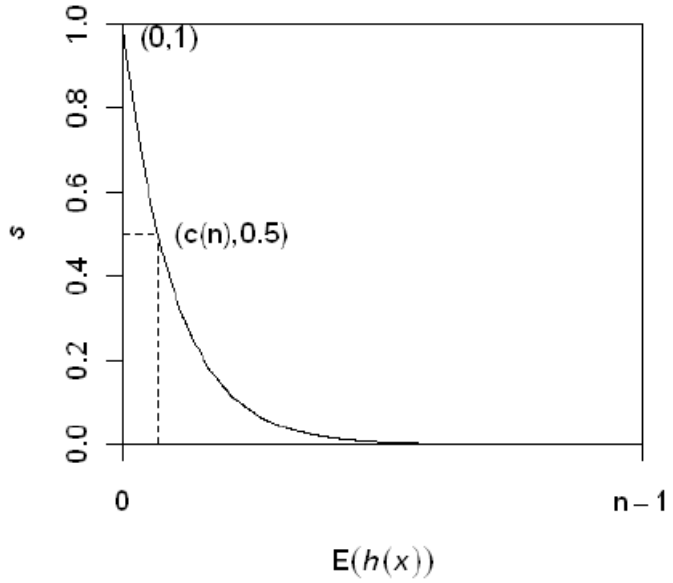
- $c(n)$은 $h(x)$를 normalize 하기 위해 사용된 상수. (average path length of unsuccessful search in BST (Binary Search Tree)) 여기서 BST의 방식을 사용하는 이유는 iTree의 External node termination이 BST의 Unsuccessful search와 동일한 구조를 가지고 있기 때문이다.

$c(n)$에 대한 식은 다음과 같다.

$$c(n) = 2H(n-1) - (2(n-1)/n)$$
$$where\ H(i) \approx ln(i) + 0.5772156649$$

오일러상수

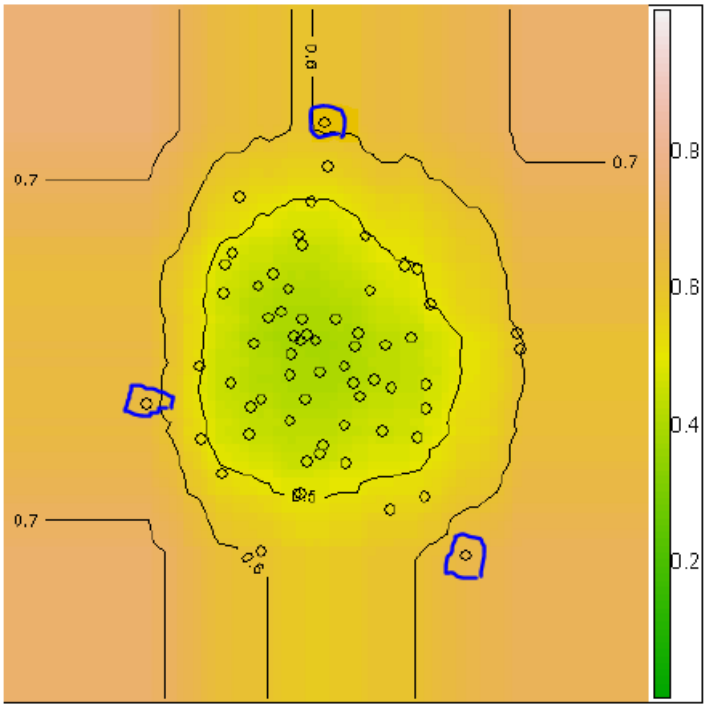
다음의 그림은 $E(h(x))$와 s의 관계를 보여주고 있다.



s는 $0 < s \leq 1$, $h(x)$는 $0 < h(x) \leq n-1$의 범위를 가진다.

- $E(h(x)) \to c(n)$, $s \to 0.5$
- $E(h(x)) \to 0$, $s \to 1$
- $E(h(x)) \to n-1$, $s \to 0$

anomaly score가 0에 가까울 수록 정상 상황에 가깝고, 1에 가까울수록 이상치에 가깝다. 만약 대부분의 데이터들이 0.5에 가까운 값을 가지면, 전체 데이터에서 이상치를 발견하지 못한것으로 간주할 수 있다.



위의 그림은 anomaly score를 시각화한 그림이다. 파란색으로 표시된 점들이 $s \geq 0.6$인 잠재적인 이상치로 생각할 수 있다.

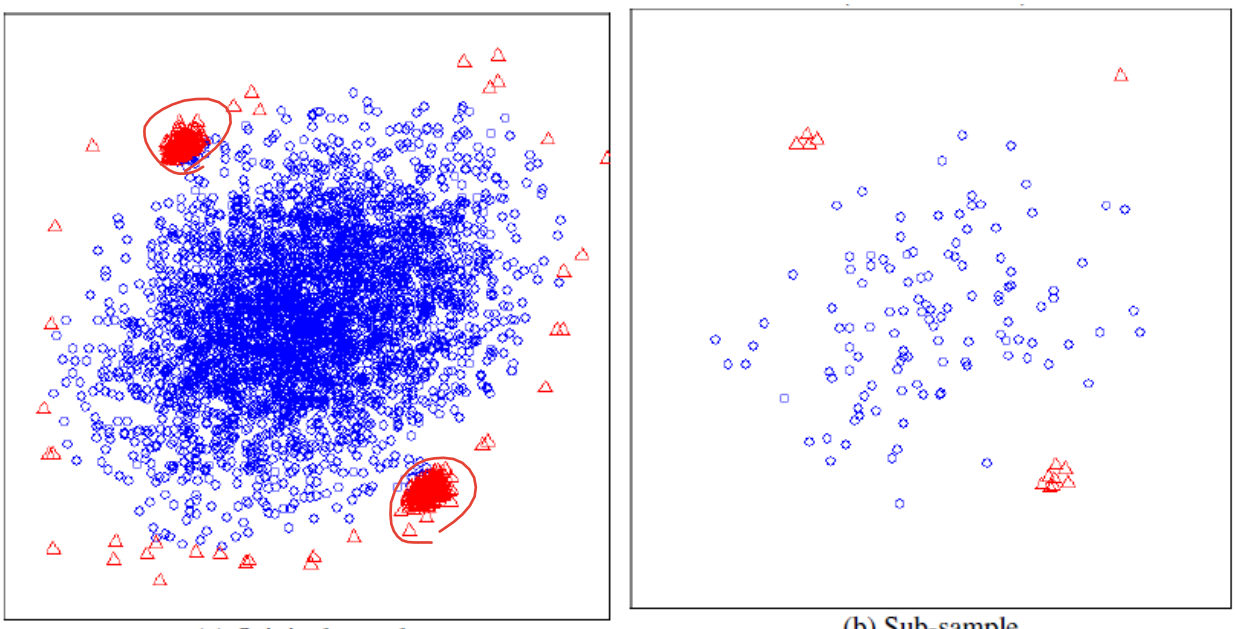
3. Isolation Tree들의 성질

기존의 Anomaly dection 방식들이 많은 sample size를 가질 때, 더 좋은 성과를 나타내는 것과 다르게 isolation 방법은 적은 sample size에서 좋은 성능을 보여준다. 많은 sample size는 이상치들을 isolate 하는 것을 방해하여 성능을 저하시키기 때문이다. 따라서 sub-sampling을 이용하는 것이 iforest에 좋은 성능을 발휘할 수 있도록 도와준다.

Anomaly detection과 관련하여 Swamping & Masking에 관한 문제들이 존재한다.

- Swamping: 정상치를 이상치로 잘못 분류하는 현상 (ex.이상치가 정상치와 가까이 위치하여 구분하기 힘든경우)
- Masking: 이상치가 너무 많아 그들을 이상치로 구분하지 못하는 현상 (ex. 이상치가 군집화 되어 있어 partition 횟수의 증가로 인해 이상치로 판단하지 못하는 경우)

Swamping과 Masking 문제 모두 결국 많은 데이터들을 사용할 때 생기는 문제인데, sub-sampling을 사용하는 Isolation Tree의 성질이 이 문제점들을 완화해준다.



(a) Original sample
(4096 instances)

(b) Sub-sample
(128 instances)

(a)의 그림을 보면 2개의 이상치 군집(빨간색)이 하나의 정상 군집(파란색)에 가깝게 위치하고 있음을을 확인할 수 있다. 하지만 (b)의 그림은 이상치 군집의 구성하는 데이터 포인들의 수도 적어지고 정상군집과 확실하게 분리되어 있는 모습이다. 그림에서도 느껴지는 것처럼 original sample을 사용하는 것보다 sub-sample들을 사용하는게 더 isolation하기 쉬움을 알 수 있다.

실제 결과도 전체 sample을 사용했을 때는 AUC 0.67, sub-sample들을 사용했을 때는 AUC 0.91로 더 좋은 결과를 보여주었다.

현실적으로 그래야 한다.