Data Analytics
EEE 4774 & 6777

Module 4 - Classification

Anomaly Detection

Spring 2022

Not a standard classification problem

- Classification is a supervised learning problem, i.e., there are samples from each class to train on.
- Although anomaly vs. no-anomaly (nominal) decision looks like a binary classification problem, anomaly detection is inherently not a supervised learning problem.
- The "anomaly class" is, by definition, an open-ended set – we want to detect any pattern that doesn't look nominal,
    - e.g., cyberattack, traffic accident, robbery, device failure, fraudulent transaction, tumor, etc.
- Even if we may have some training data from a type of anomaly, we still want to detect unseen anomaly types.
- Hence, in training we typically see only nominal data (semi-supervised learning) or a dataset which may contain some anomalies (unsupervised learning)

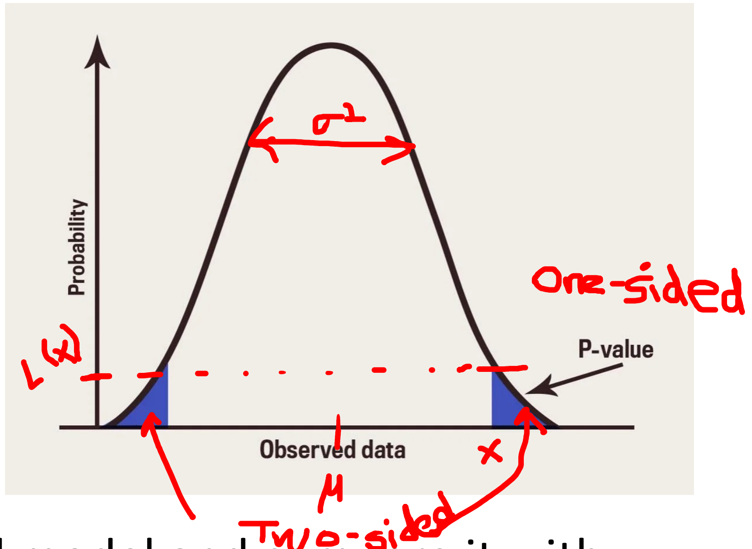
Generative Models

- p-value: probability of seeing a data instance as extreme as or more extreme than the observed one (the sum of tail probabilities)

  e.g., Gaussian generative (likelihood) model

- A test instance can be declared an anomaly/outlier if

its p-value is less than a critical threshold, e.g., $\underline{0.05}$, 0.01, 0.001, etc.



Assumption: Training data is clean of anomalies

Training: fit a suitable likelihood model to (nominal) training data

Testing: compute the likelihood of the test instance under the trained model and compare it with the threshold to decide.

Example: Biased coin detection

- Fitting binomial distribution to historical nominal data from unbiased coin give the heads (success) probability as $\theta = 0.5$
- Test data: 9 heads and 3 tails in 12 tosses

$$\text{One-sided p-value: } p = \sum_{h=9}^{12} \binom{12}{h} 0.5^h 0.5^{12-h} = \sum_{h=9}^{12} \binom{12}{h} 0.5^{12} = 0.073$$

$$\text{Two-sided p-value: } p = \sum_{h=9}^{12} \binom{12}{h} 0.5^{12} + \sum_{h=0}^{3} \binom{12}{h} 0.5^{12} = 0.073 + 0.073 = 0.146$$

- Both one-sided and two-sided p-values are higher than the commonly-used statistical significance level (threshold) of $\underline{0.05}$, so we can conclude that the coin used to obtain the test data is **nominal** (unbiased)
- However, there is a fundamental issue with p-value tests! See the following formulation of the same problem.

Example: Biased coin detection

- Consider a different generative model for the observed data: keep tossing the coin until 3 tails are observed.
- In this case, instead of binomial distribution, the most appropriate generative model that explains the experiment is the negative binomial distribution with pmf $\binom{N-1}{f-1}\theta^s(1-\theta)^f$
- Test data: 9 heads and 3 tails in 12 tosses

$$p = \sum_{h=9}^{\infty}\binom{3+h-1}{3-1}0.5^h 0.5^3 = \sum_{h=9}^{\infty}\binom{2+h}{2}0.5^{3+h} = 0.0327$$

- Interestingly, in this case, the p-value is smaller than the commonly-used statistical significance level (threshold) of 0.05.
- For the same dataset, we can now decide for an anomaly (biased coin) using the same threshold!
- Beware of threshold selection! Fixed thresholds do not work for every model.

Threshold Selection

- In binary classification, threshold determines the fundamental trade-off between two conflicting objectives:
  - Maximizing True Positive Rate (TPR), i.e., true alarms

$$TPR = \frac{\#\, true\ positives}{\#\, positives}$$

min misdet.

  - Minimizing False Positive Rate (FPR), i.e., false alarms

$$FPR = \frac{\#\, false\ positives}{\#\, negatives}$$

- Decreasing threshold will increase TPR but also increase FPR
- Increasing threshold will decrease FPR but also decrease TPR
- While setting the threshold the goal is to set a desired balance between TPR and FPR.
- A practical way to do that is to choose the lowest threshold that satisfies a false alarm constraint (FPR), e.g., 0.01

Anom. score e.g. 1-p val $\overset{\text{Alarm}}{>}$ thr
evidence $\underset{\text{No Alarm}}{<}$

Performance Evaluation

- Binary classification metrics TPR and FPR commonly used to evaluate the anomaly detection performance.
- ROC (receiver operating characteristic) curve plots TPR vs. FPR to show this trade-off for different threshold values
- AUC (area under the ROC curve) gives a scalar metric to compare detectors – higher AUC means better detector

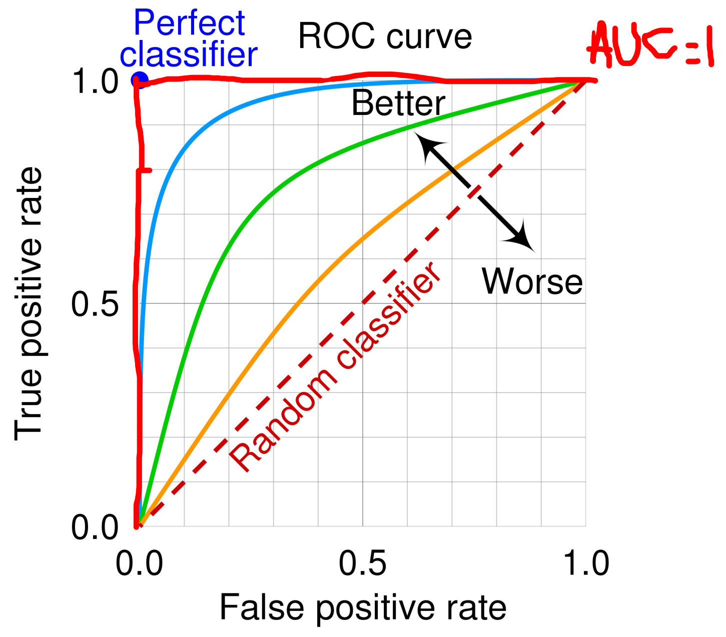
$$AUC \in (0,1)$$

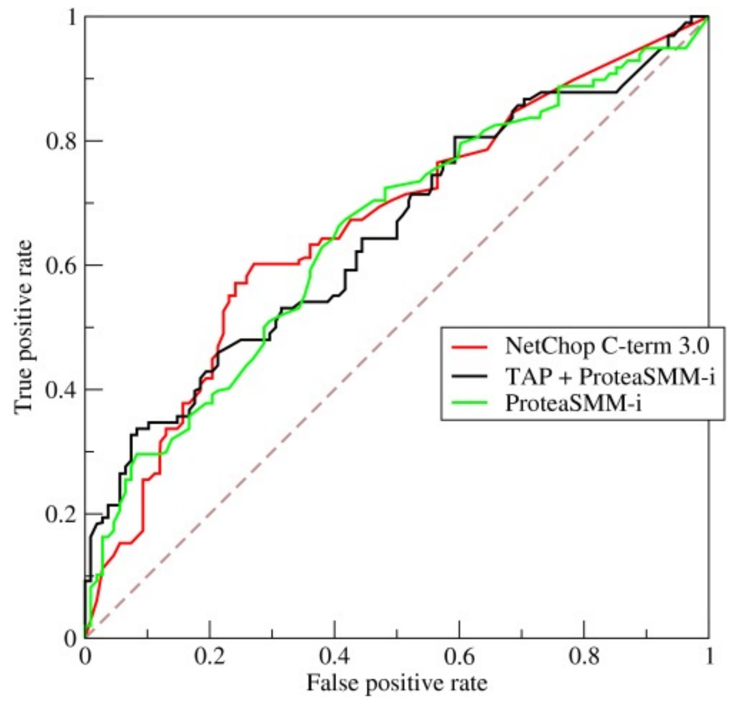
- F1 score similarly gives a scalar performance metric. It deals better with class imbalance.

$$F1 = 2\times \frac{Precision\times Recall}{Precision + Recall}$$

$$Precision = \frac{\#\ true\ positives}{\#\ true\ positives + \#\ false\ positives} = \frac{\#\ true\ alarms}{\#\ all\ alarms}$$

$$Recall = TPR$$

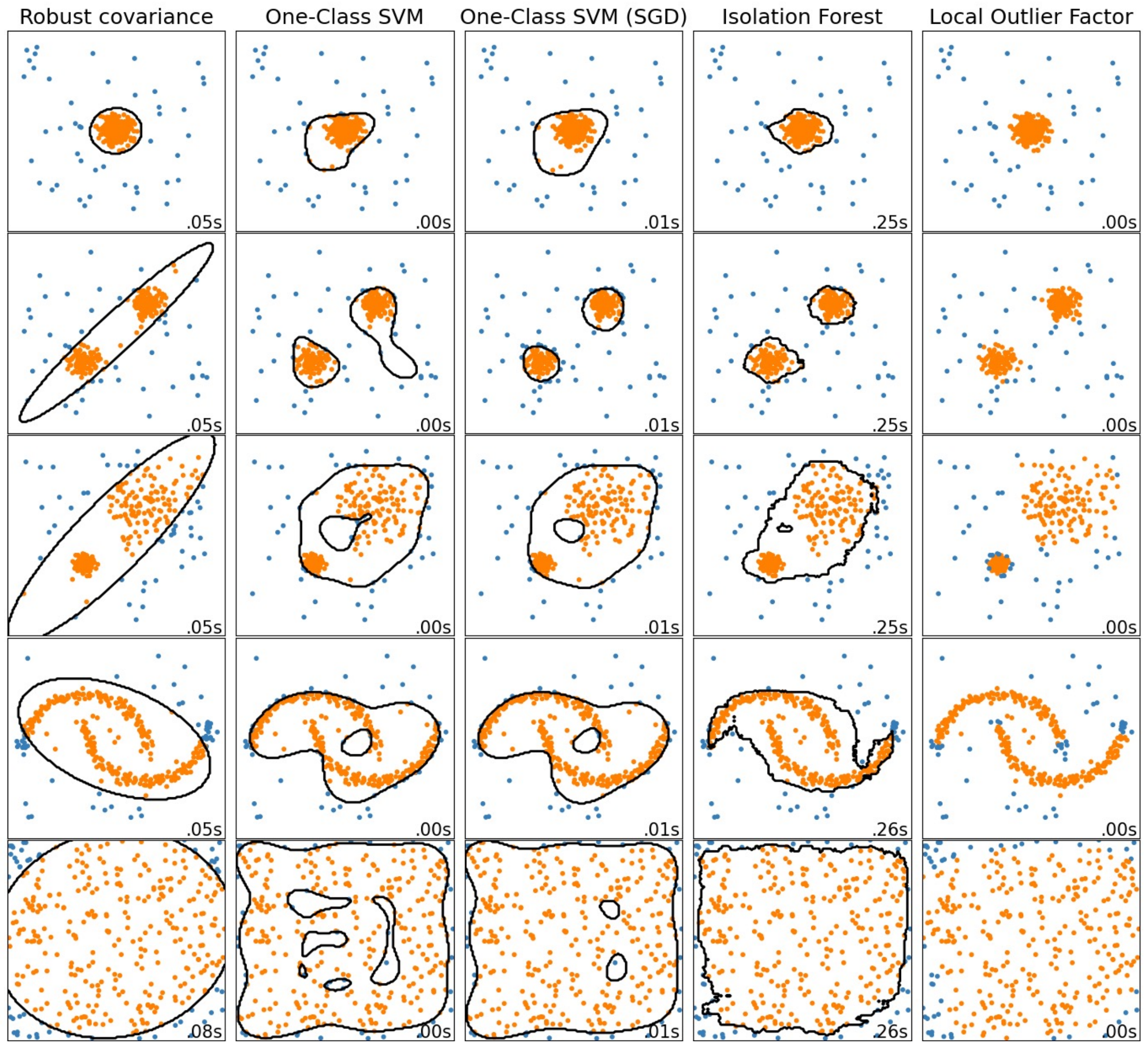




Other methods for anomaly detection

- kNN, sklearn.neighbors.LocalOutlierFactor (scikit-learn)
- One-Class SVM, svm.OneClassSVM (scikit-learn)
- Isolation Forest, ensemble.IsolationForest (scikit-learn)
- For more information see:

https://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection


Robust covariance
One-Class SVM
One-Class SVM (SGD)
Isolation Forest
Local Outlier Factor
.05s
.00s
.01s
.25s
.00s
.05s
.00s
.01s
.25s
.00s
.05s
.00s
.01s
.25s
.00s
.05s
.00s
.01s
.26s
.00s
.08s
.00s
.01s
.26s
.00s