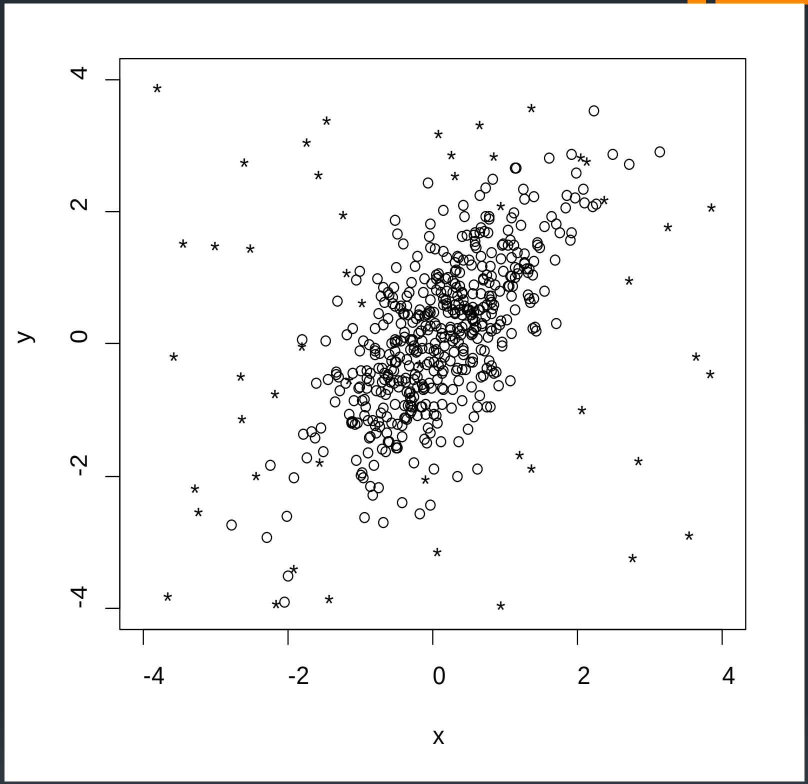
Introduction to Anomaly Detection

Tom Dietterich

Anomaly Detection

- Anomaly: A data point generated by a different process than the process that generates the normal data points
  - Example: Fraud Detection
    - Normal points: Legitimate financial transactions
    - Anomaly points: Fraudulent transactions
  - Example: Sensor Data
    - Normal points: Correct data values
    - Anomaly points: Bad values (broken sensors)



Three Settings

- Supervised
  - Training data labeled with “nominal” or “anomaly”
- Clean
  - Training data are all “nominal”, test data contaminated with “anomaly” points.
- Unsupervised
  - Training data consist of mixture of “nominal” and “anomaly” points

What Makes Anomaly Detection Hard

- Nominal distribution has “heavy tails”
  - Naturally has many outliers
- Anomaly distribution is very similar to nominal distribution
- Unsupervised anomaly detection with very frequent anomalies
  - High level of contamination makes learning the nominal distribution hard
- Anomalies are changing over time
  - Adversaries try to fool the anomaly detector

Approaches to Anomaly Detection: (1) Density Estimation

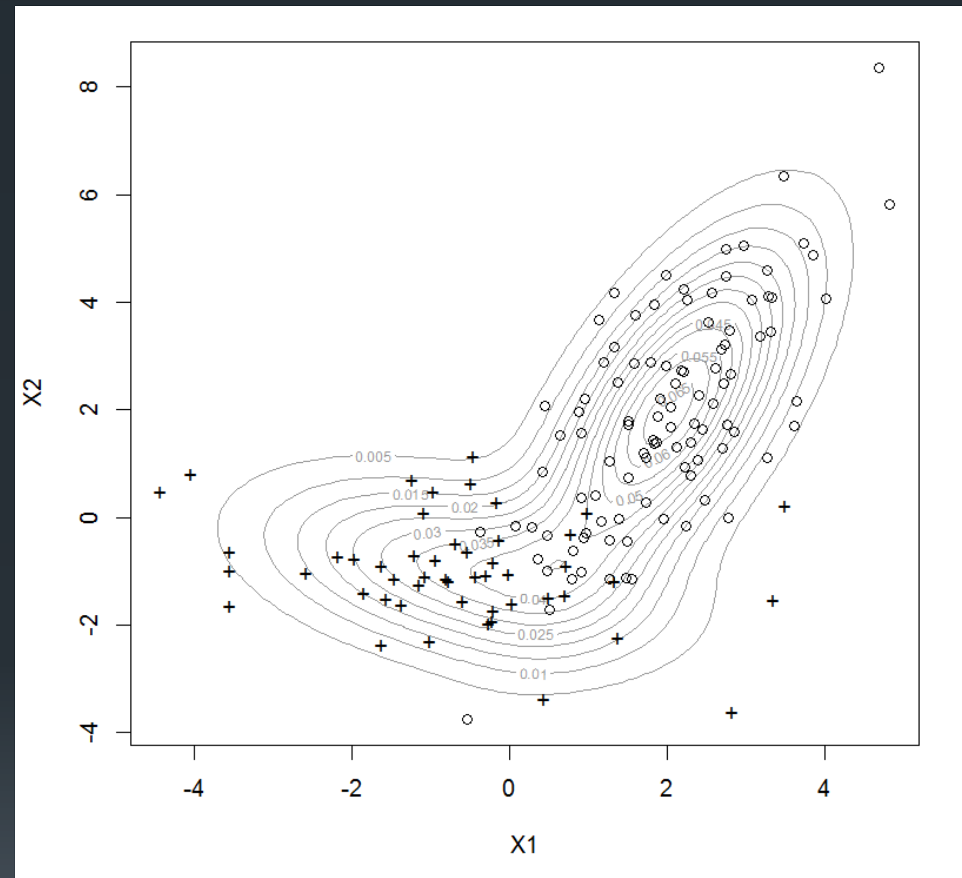
- Given data points $x_1, \ldots, x_N$ (each a feature vector of length d)
- Find a probability density function f to maximize

$$\sum_i -\log f(x_i)$$

- The function f must be constrained so that it cannot simply put density $\frac{1}{N}$ on each data point
- Anomaly score $A(x_q) = -\log f(x_q)$
- This is known as the "surprise"
  - $-\log 1 = 0$ "no surprise"
  - $-\log 0 = \infty$ "infinite surprise"

EGMM: Ensemble of Gaussian Mixture Models

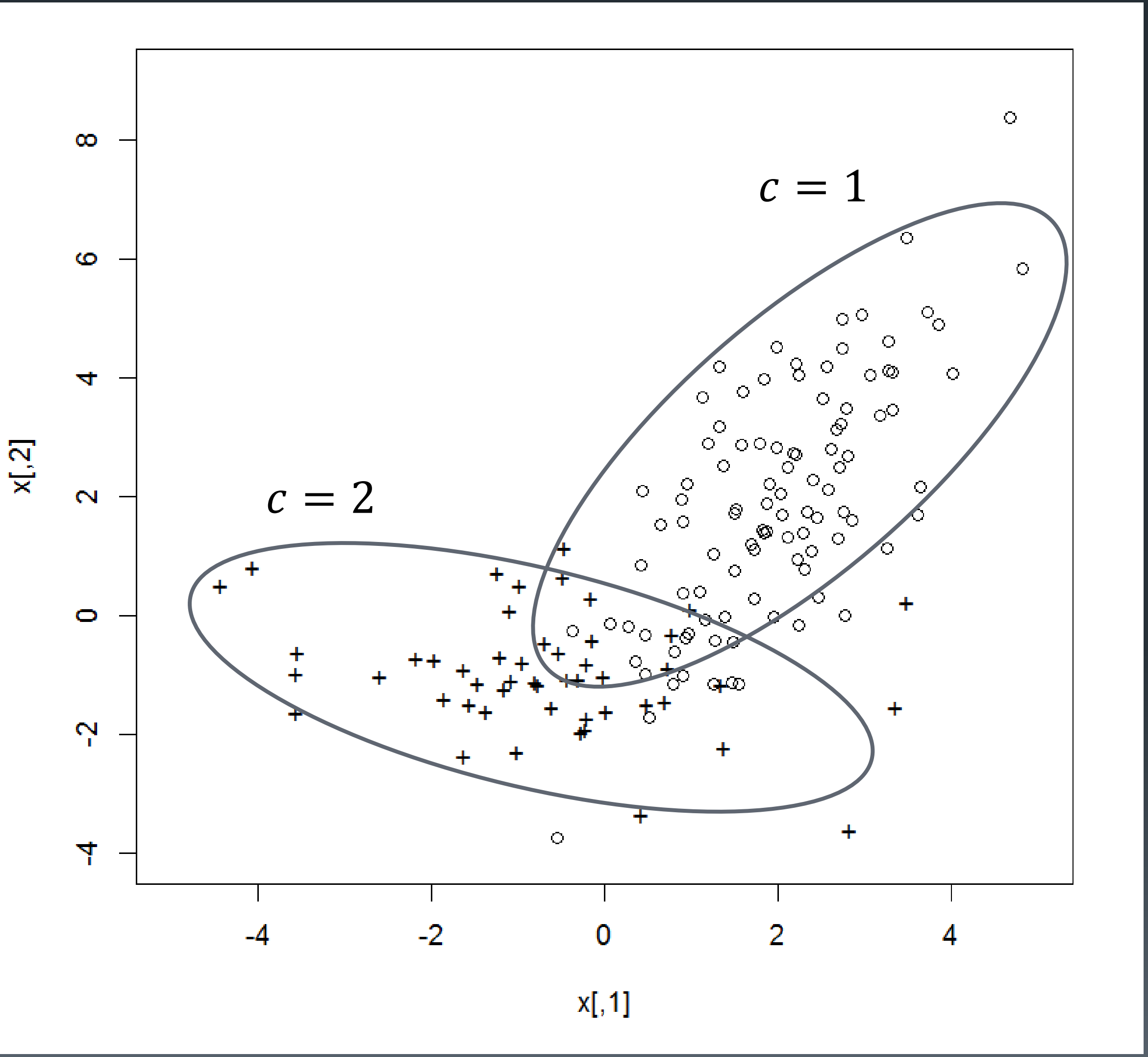
- Consider this complex cloud of points
- It is clearly not normally distributed
- But it can be modeled as the weighted sum ("mixture") of two Gaussian distributions



Mixture of Gaussians

- $P(c = 1) = 2/3$
- $P(c = 2) = 1/3$
- $P(x|c = 1)$ “o”
- $P(x|c = 2)$ “+”

- There are good algorithms for fitting GMMs to data

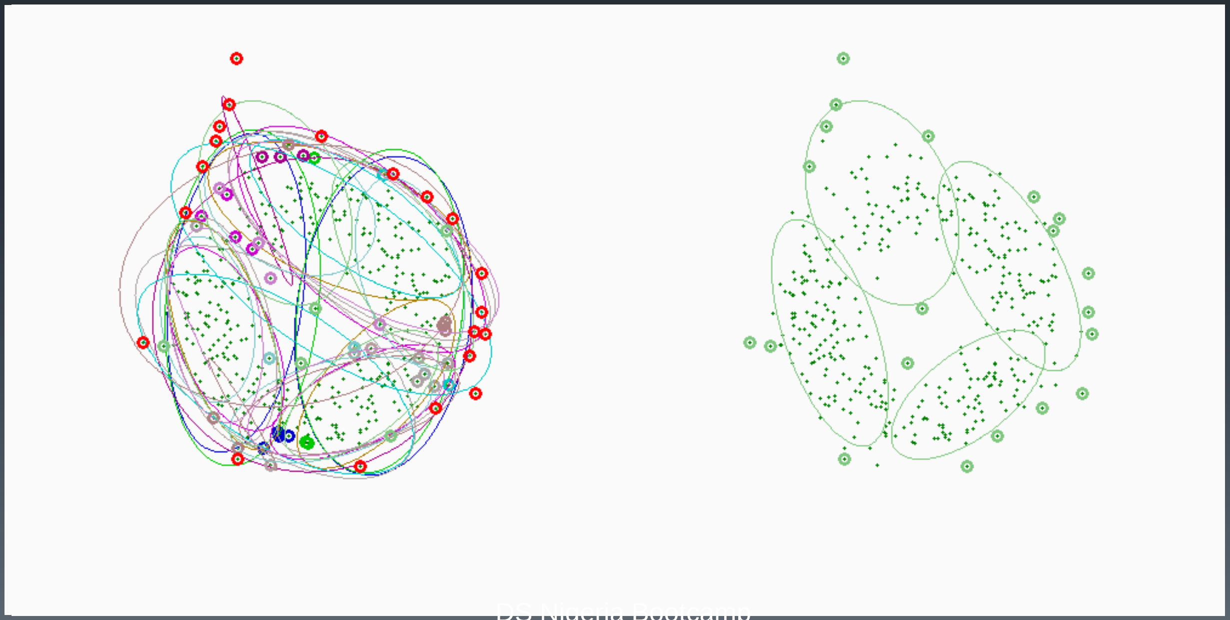


Fit a single Gaussian

- Give you $x_1, \ldots, x_N$
- mean: $\mu = \frac{1}{N}\sum_i x_i$
- variance:$\sigma^2 = \frac{1}{N}\sum_i (x_i - \mu)^2$
- $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-\frac{1}{2}(x-\mu)^2}{\sigma^2}}$
- Mixture:
- Goal: find two means: μ_1, μ_2 and two variances σ_1^2, σ_2^2 and the mixture proportion p
- $f(x) = p\frac{1}{\sqrt{2\pi}\sigma_1} e^{\frac{-\frac{1}{2}(x-\mu_1)^2}{\sigma_1^2}} + (1-p)\frac{1}{\sqrt{2\pi}\sigma_2} e^{\frac{-\frac{1}{2}(x-\mu_2)^2}{\sigma_2^2}}$

Ensemble of GMMs

- Train L independent Gaussian Mixture Models
- Train model $\ell = 1, \ldots, L$ on a bootstrap replicate of the data
- Vary the number of clusters K
- Delete any model with log likelihood < 70% of best model
- Compute average surprise: $A(x_q) = -\frac{1}{L}\sum_{\ell} \log f_{\ell}(x_q)$

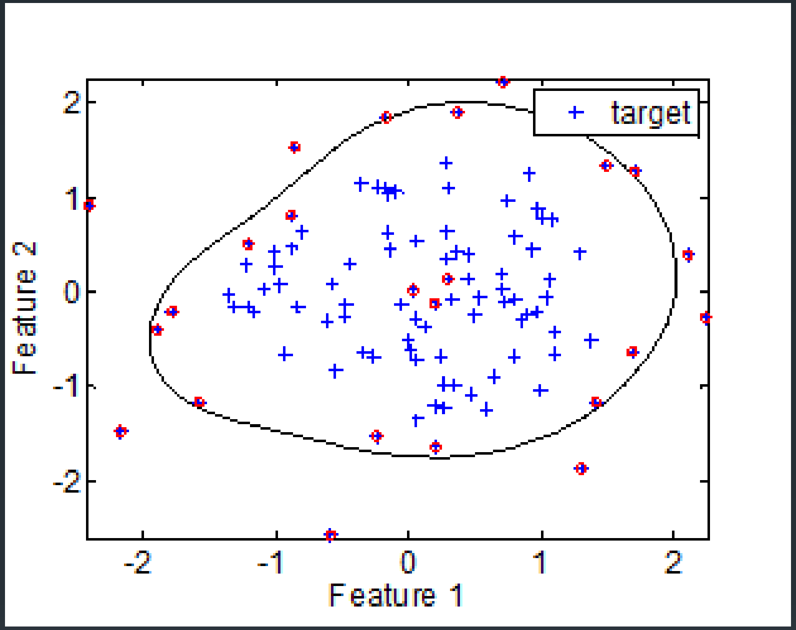


Advantages and Disadvantages of Density Estimation

- Advantages:
  - Clean theoretical understanding
  - Many methods:
    - Kernel density estimation
    - Ensemble of Gaussian Mixture Models
    - Deep density estimation
- Disadvantages:
  - General density estimation requires large amounts of data
  - Sample size grows as $\exp\frac{d+4}{2}$
  - If the anomaly points form a tight cluster, it will be assigned high probability density (= low anomaly score)

Approaches to Anomaly Detection: (2) Quantile Methods

- Find a smooth boundary that encloses fraction $1-\alpha$ of the data
- Map each data point x into an $(N-1)$-dimensional space based on its kernel distance to each of the other data points
- Surround $1-\alpha$ of the points with a surface:
- Linear surface:
  - One-class support-vector machine (OC-SVM)
- Hypersphere:
  - Support-vector data description (SVDD)



$A(x_q)$ = distance from the boundary

Advantages and Disadvantages of Quantile Methods

- Advantages:
  - Amount of training data needed grows as $\frac{1}{\epsilon^2}$, where ϵ is the accuracy of the $1-\alpha$ quantile
- Disadvantages:
  - Requires tuning a kernel function
  - Algorithms do not scale to large data sets
  - Does not perform very well for ranking

Approaches to Anomaly Detection: (3) Distance-Based Methods

- Choose a distance metric $||x_i - x_j||$ between any two data points x_i and x_j
- $A(x)$ = anomaly score = distance to k-th nearest data point
- Points in empty regions of the input space are likely to be anomalies

Advantages and Disadvantages of Distance Methods

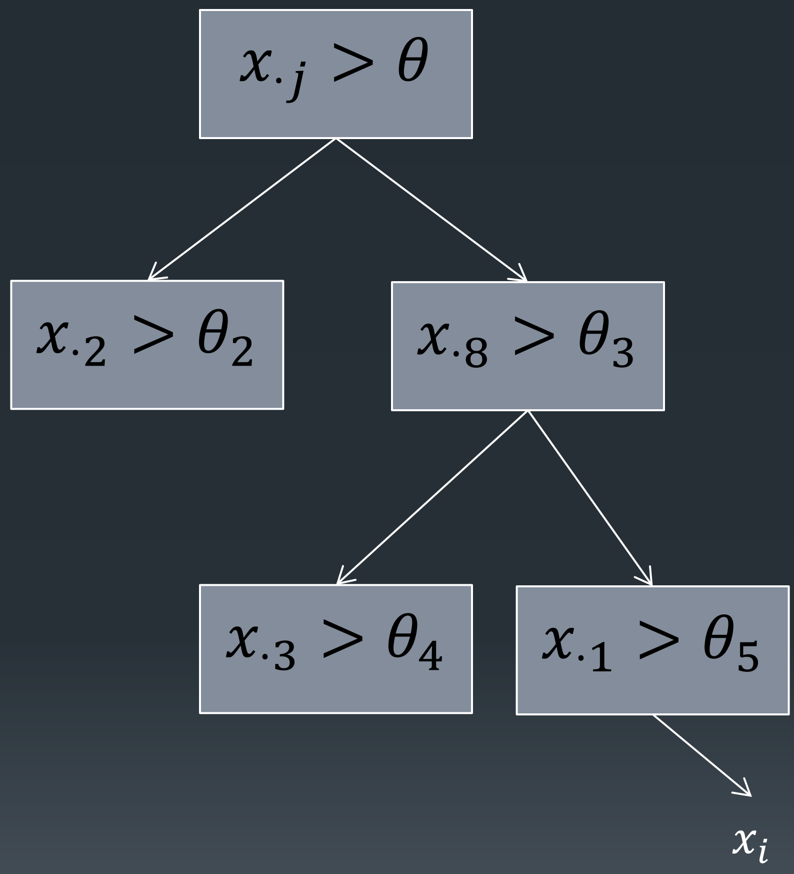
- Advantages:
  - Easy to understand
  - Easy to tune
  - Perform quite well
- Disadvantages
  - Fail when the anomalies form tight clusters
  - Naïve implementation requires computing all pairwise distances (time proportional to dN^2)
  - Must store the training instances

Approaches to Anomaly Detection: (4) Projection Methods

- Isolation Forest
- LODA

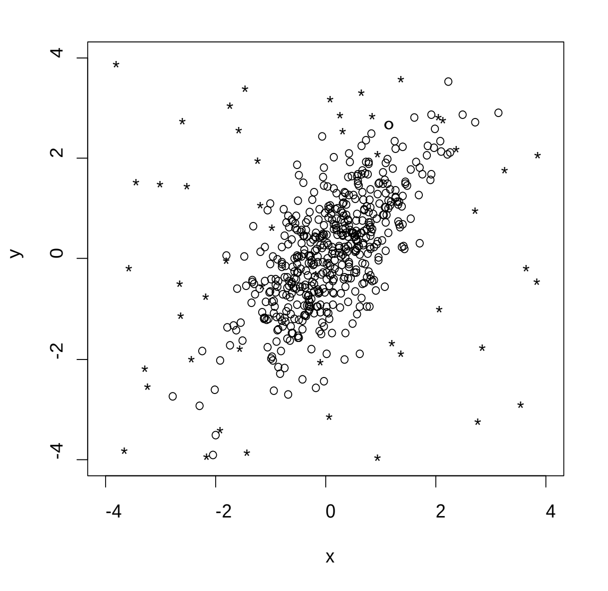
Isolation Forest [Liu, Ting, Zhou, 2011]

- Construct a fully random binary tree
  - choose attribute j at random
  - choose splitting threshold θ uniformly from $[\min(x_{\cdot j}), \max(x_{\cdot j})]$
  - until every data point is in its own leaf
  - let $d(x_i)$ be the depth of point x_i
- repeat 100 times
  - let $\bar{d}(x_i)$ be the average depth of x_i
  - $A(x_i) = 2^{-\left(\frac{\bar{d}(x_i)}{r(x_i)}\right)}$
    - $r(x_i)$ is the expected depth



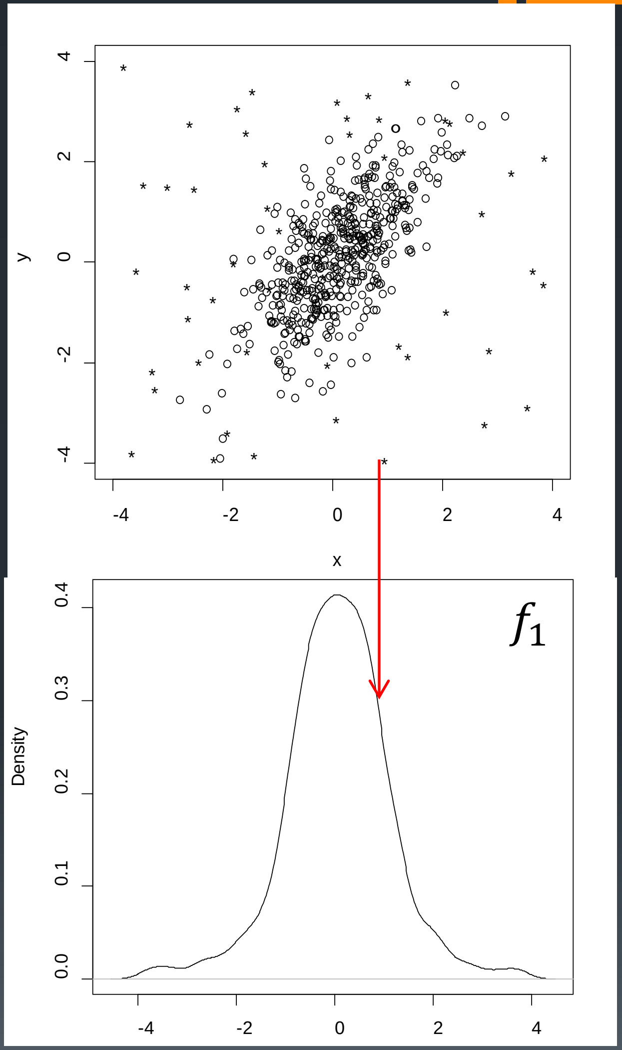
LODA: Lightweight Online Detector of Anomalies [Pevny, 2016]

- $\Pi_1, \ldots, \Pi_M$ set of M sparse random projections
- $f_1, \ldots, f_M$ corresponding 1-dimensional density estimators
- $S(x) = \frac{1}{M} \sum_m -\log f_m(x)$ average "surprise"



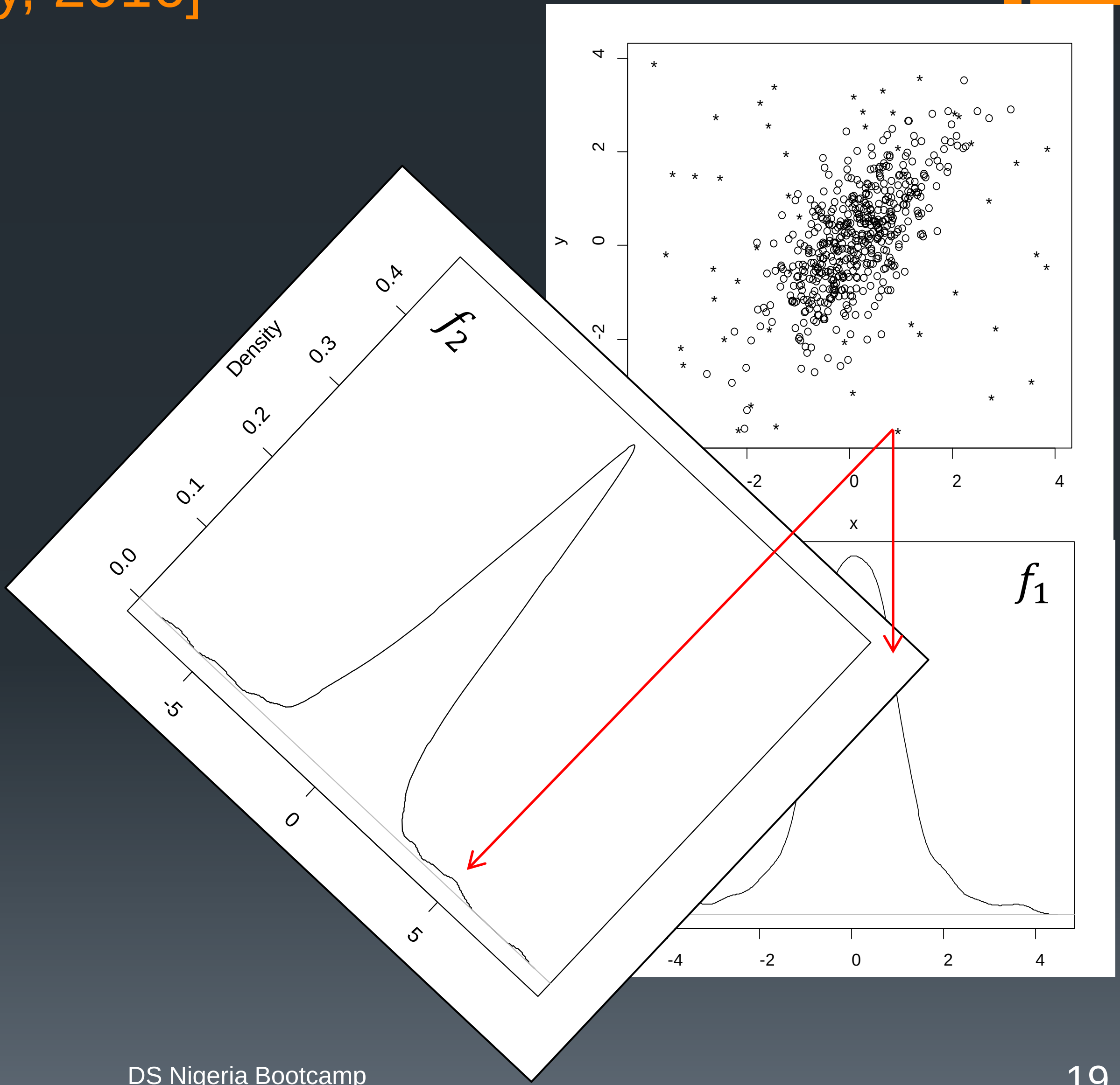
LODA: Lightweight Online Detector of Anomalies [Pevny, 2016]

- $\Pi_1, \ldots, \Pi_M$ set of M sparse random projections
- $f_1, \ldots, f_M$ corresponding 1-dimensional density estimators
- $S(x) = \frac{1}{M} \sum_m - \log f_m(x)$ average “surprise”



LODA: Lightweight Online Detector of Anomalies [Pevny, 2016]

- $\Pi_1, \ldots, \Pi_M$ set of M sparse random projections
- $f_1, \ldots, f_M$ corresponding 1-dimensional density estimators
- $S(x) = \frac{1}{M} \sum_m - \log f_m(x)$ average "surprise"



Benchmarking Study
[Andrew Emmott]

- Most AD papers only evaluate on a few datasets
- Often proprietary or very easy (e.g., KDD 1999)
- Research community needs a large and growing collection of public anomaly benchmarks

[Emmott, Das, Dietterich, Fern, Wong, 2013; KDD ODD-2013]
[Emmott, Das, Dietterich, Fern, Wong. 2016; arXiv 1503.01158v2]

Benchmarking Methodology

- Select 19 data sets from UC Irvine repository
- Choose one or more classes to be “anomalies”; the rest are “nominals”
- Manipulate
  - Relative frequency
  - Point difficulty
  - Irrelevant features
  - Clusteredness
- 20 replicates of each configuration
- Result: 25,685 Benchmark Datasets

Algorithms

- Density-Based Approaches
  - RKDE: Robust Kernel Density Estimation (Kim & Scott, 2008)
  - EGMM: Ensemble Gaussian Mixture Model (our group)
- Quantile-Based Methods
  - OCSVM: One-class SVM (Schoelkopf, et al., 1999)
  - SVDD: Support Vector Data Description (Tax & Duin, 2004)
- Neighbor-Based Methods
  - LOF: Local Outlier Factor (Breunig, et al., 2000)
  - ABOD: kNN Angle-Based Outlier Detector (Kriegel, et al., 2008)
- Projection-Based Methods
  - IFOR: Isolation Forest (Liu, et al., 2008)
  - LODA: Lightweight Online Detector of Anomalies (Pevny, 2016)

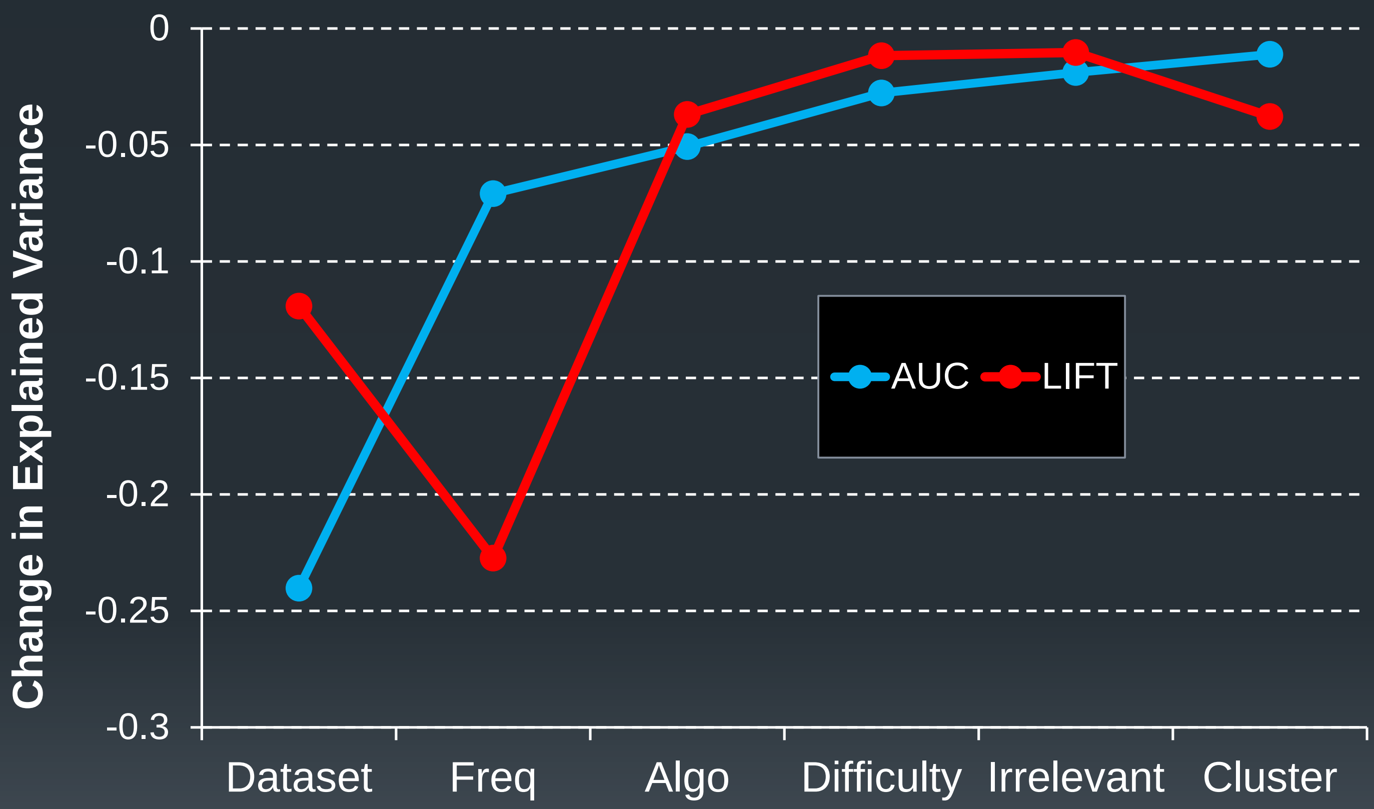
Analysis

- Linear ANOVA
  - $metric \sim rf + pd + cl + ir + mset + algo$
    - rf: relative frequency
    - pd: point difficulty
    - cl: normalized clusteredness
    - ir: irrelevant features
    - mset: “Mother” set
    - algo: anomaly detection algorithm
- Validate the effect of each factor
- Assess the *algo* effect while controlling for all other factors

Evaluation Metrics

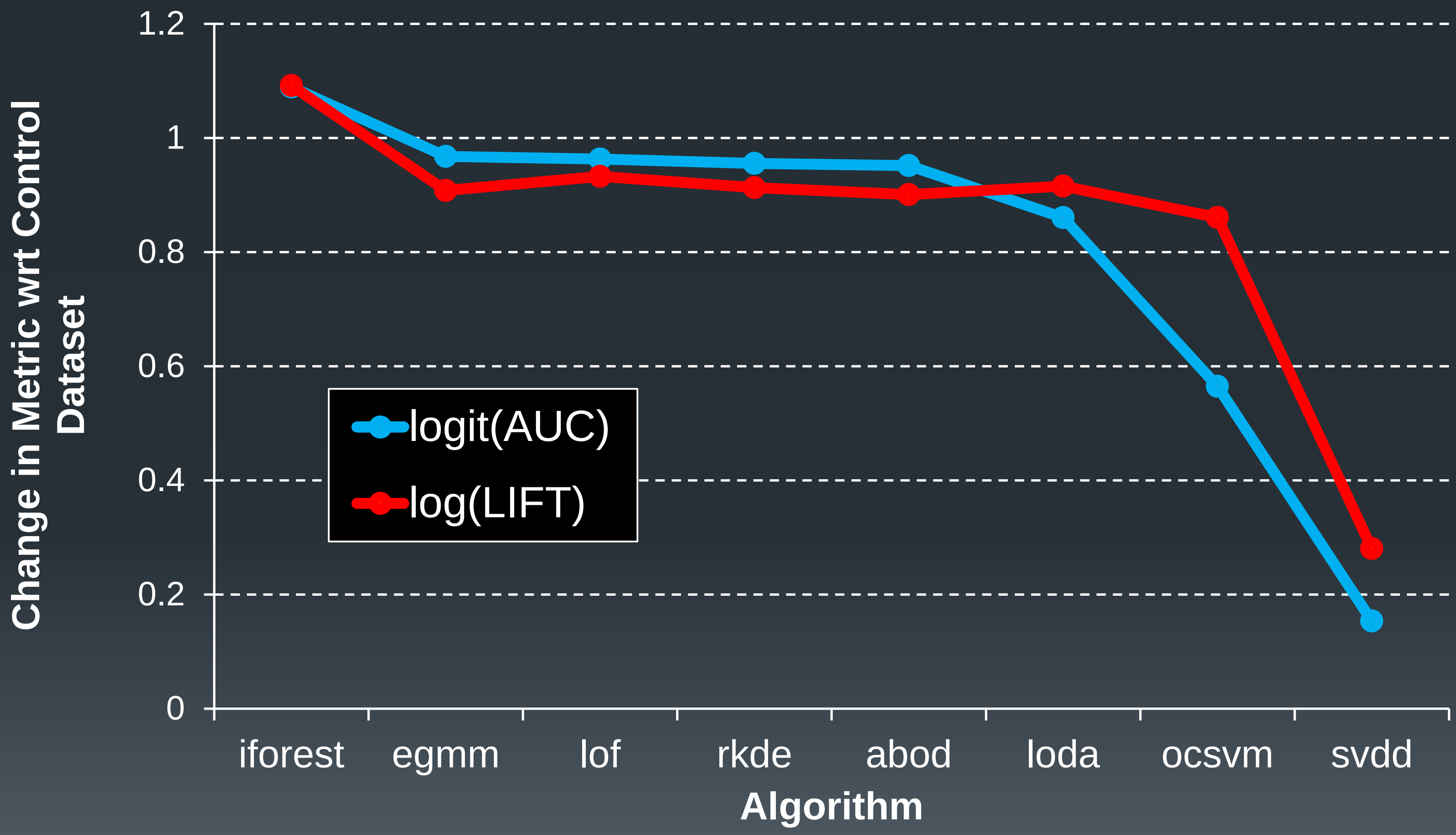
- AUC: Area Under the ROC Curve
  - binary decision: Nominal vs. Anomaly
  - what is the probability that the algorithm correctly ranks a randomly-chosen anomaly above a randomly-chosen nominal point?
  - We measure $\log \frac{AUC}{1-AUC}$
- LIFT: Ratio of precision of algorithm to precision of random guessing
  - Related to Average Precision (AP)
  - We measure $\log \frac{AP}{E[AP]}$

What Matters the Most?



- Problem and Relative Frequency!
- Choice of algorithm ranks third

Algorithm Comparison



iForest Advantages

- Most robust to irrelevant features
  - for both AUC and LIFT
- Second most robust to clustered anomaly points
  - for AUC

iForest Tricks of the Trade

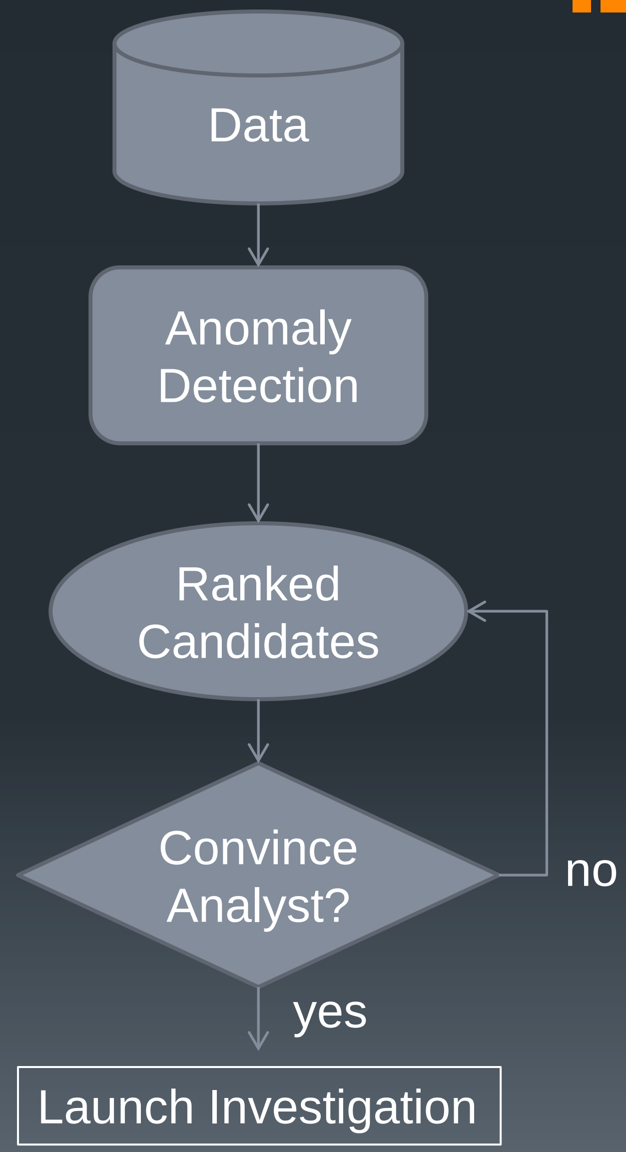
- If your training data are clean
  - Use bootstrap replicate samples to train each isolation tree
- If your training data are contaminated with anomalies
  - Use small random sub-samples
  - Typical sizes vary from 16 to 2048 points
  - This helps Isolation Forest be more robust to the contamination

Anomaly Detection Workflow

- Collect data
  - Do NOT perform feature selection
  - Normalize the data so that the inter-quartile range (25^{th} to 75^{th} quantiles) is 1.0 and centered on 0
- Fit the isolation forest
  - The number of trees should be chosen to ensure that the anomaly scores are stable (e.g., compare anomaly scores computed on bootstrap replicates of the isolation forest)
  - Smaller subsamples require larger forests
  - The forest must grow in size for large dimension d

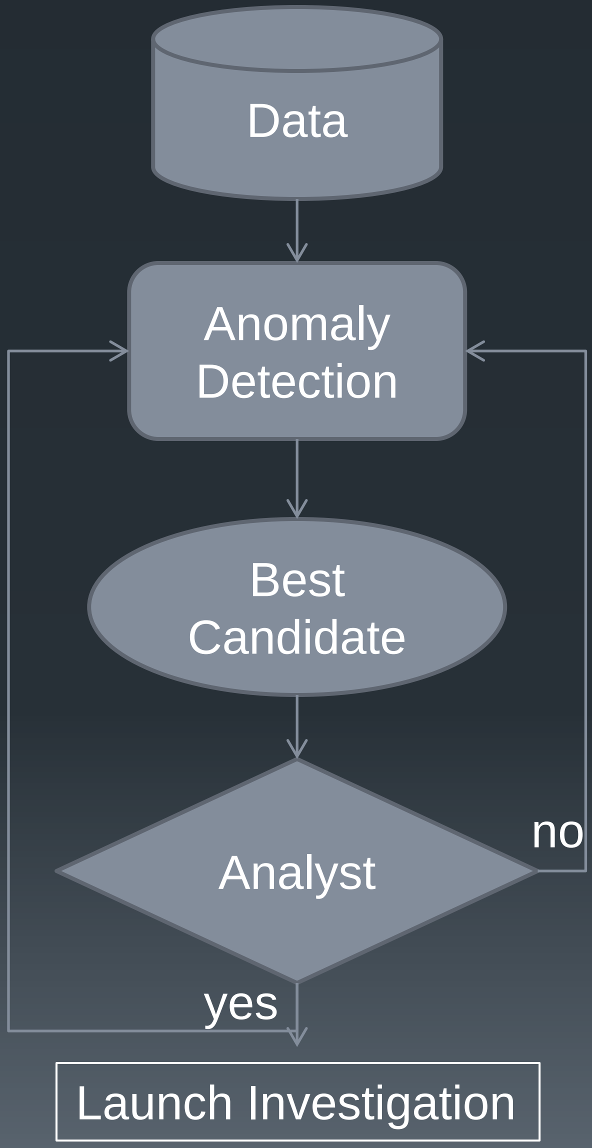
Deployment Workflow 1: Fraud Detection

- Most cases require a human in the loop
- Show human analyst the top-ranked anomaly
- The analyst decides whether to take action (e.g., launching a fraud investigation)



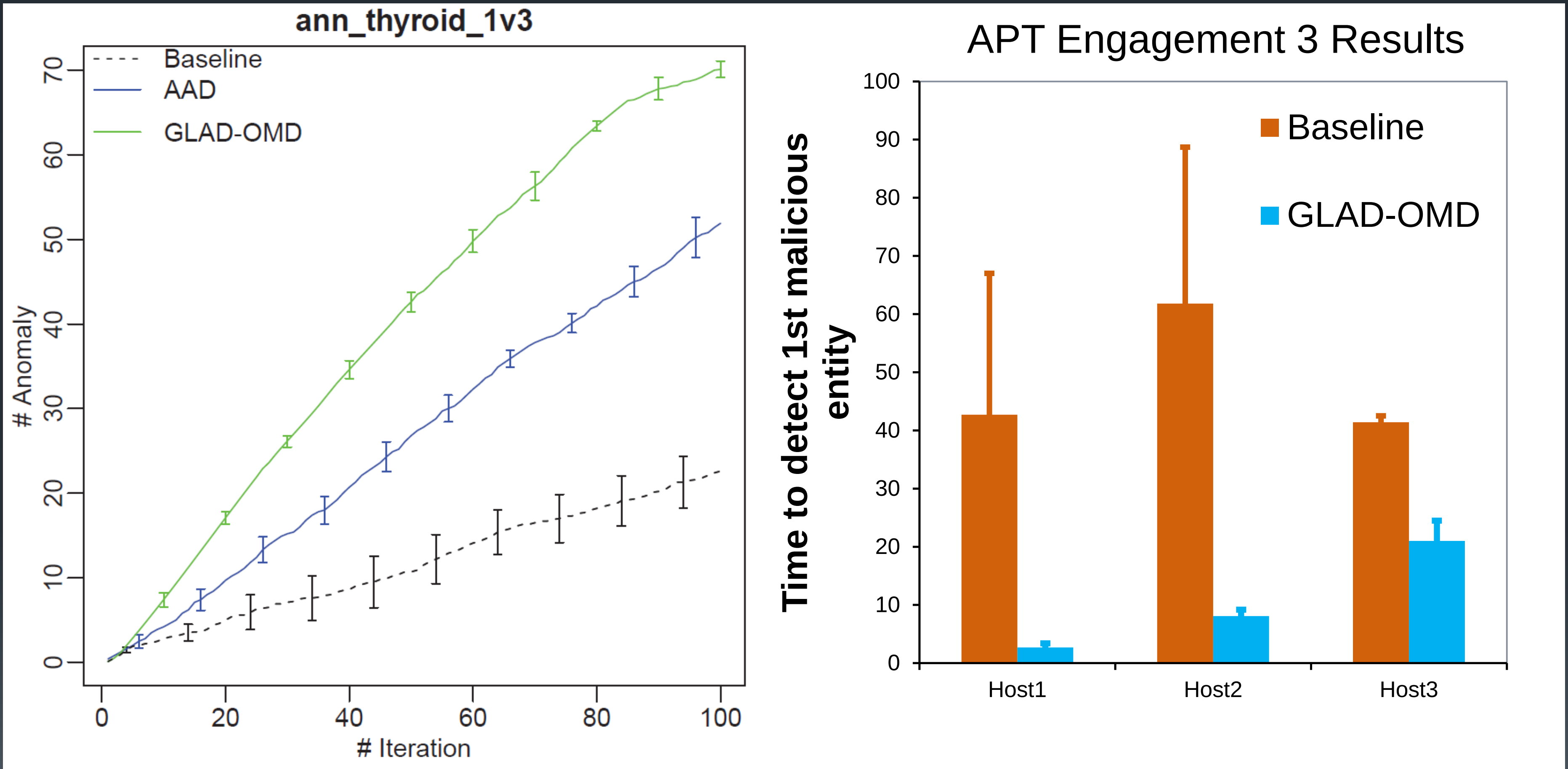
Incorporating Analyst Feedback

- Show top-ranked (unlabeled) candidate to the Analyst
- Analyst labels candidate
- Label is used to update the anomaly detector



[Das, et al, ICDM 2016]
[Siddiqui, et al., KDD 2018]

Analyst Feedback Yields Huge Improvements in Anomaly Discovery



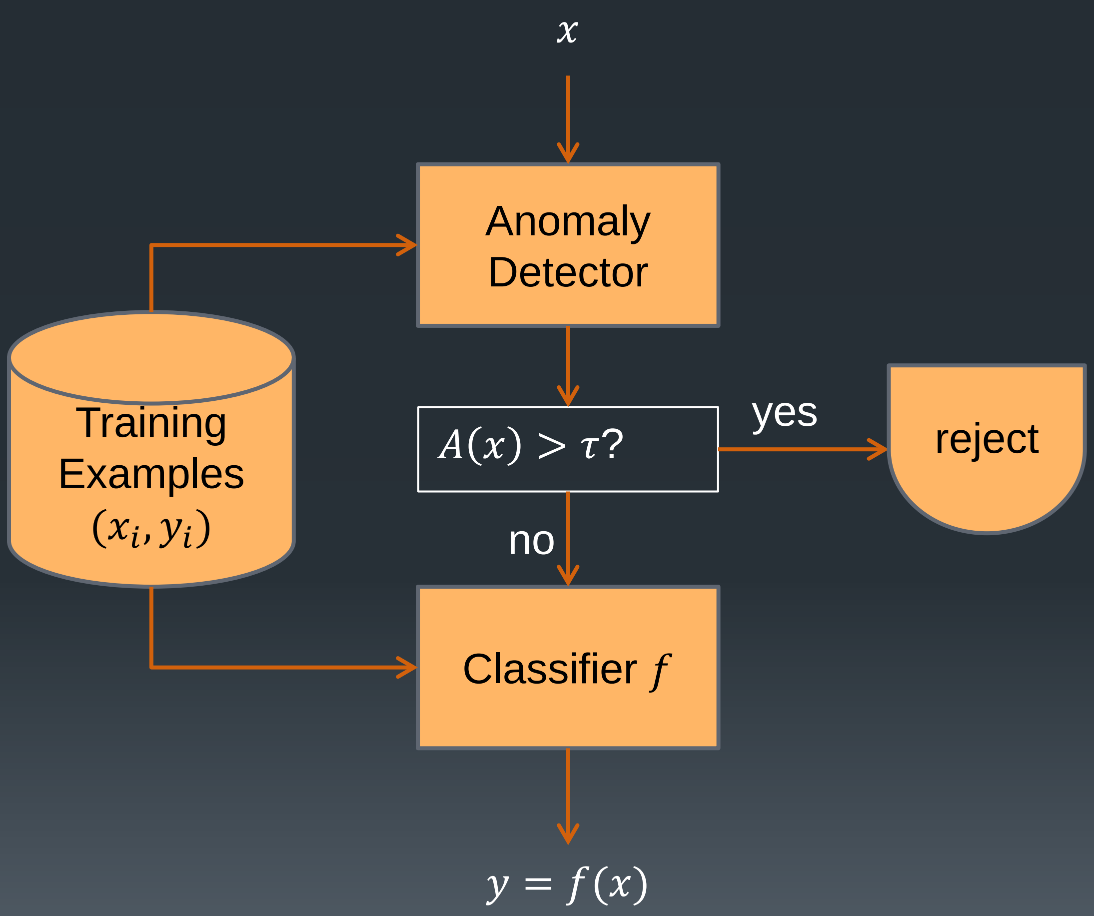
Method

- Transform the Isolation Forest into a gigantic linear model
  - Each node in each tree becomes a Boolean feature
  - Initial weight of each feature is 1.0, so that the weighted sum == total isolation depth
- Apply online convex optimization algorithms to learn from analyst feedback
  - Online Mirror Descent adjusts the weights to reduce the score of anomalies and increase the score of nominals

Deployment Workflow 2: Open Category Detection

- Training data for classes {1,…,K}
- Test data may contain queries corresponding to additional classes
- Can we detect them?

Prediction with Anomaly Detection



Automated Counting of Freshwater Macroinvertebrates

- Goal: Assess the health of freshwater streams
- Method:
  - Collect specimens via kicknet
  - Photograph in the lab
  - Classify to genus and species

www.epa.gov

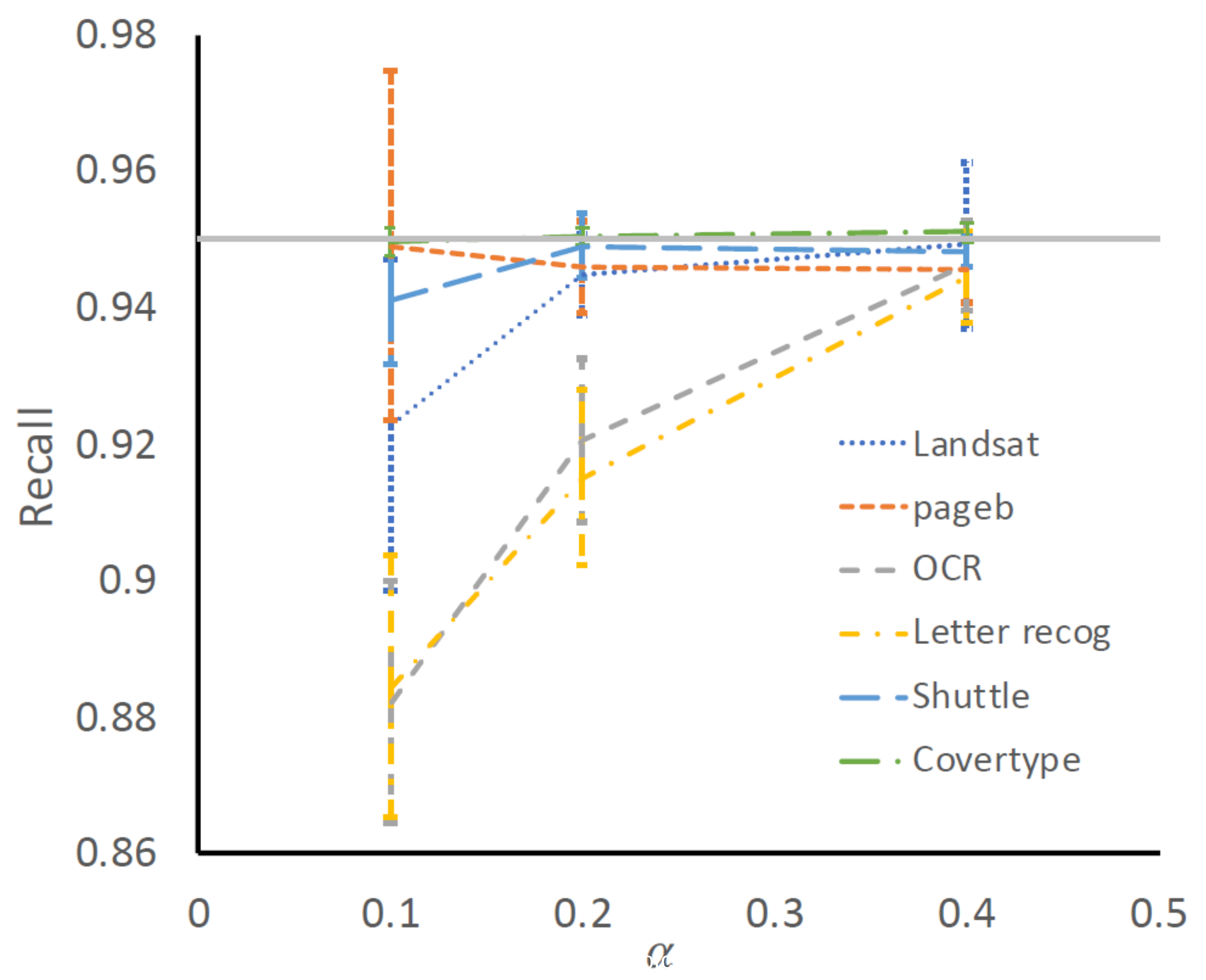
Open Category Object Recognition

- Train on 29 classes of insects
- Test set may contain additional species

Theoretical Guarantee for Open Category Detection

- Assumptions:
  - Clean training data
  - Second large (unlabeled) contaminated data set is available
  - Tight estimate on α, the fraction of anomalies in the contaminated data set
- Specify:
  - A desired quantile q and accuracy level ϵ
  - Our algorithm shows how to choose a threshold τ such that with high probability we will detect fraction $1-(q+\epsilon)$ of the anomalies

Results on six UCI benchmarks ($q = 0.95$)



Summary

- Anomaly detection has been less studied than other areas of machine learning
- Many important applications
  - fraud detection, cyber security
  - open category detection, robust ML
- Isolation Forest is a good method
- Analyst feedback can greatly improve the efficiency of detecting true anomalies

Open Research Questions

- Why does sub-sampling improve the robustness of anomaly detectors trained on contaminated data?
- Anomalies in time-series data
- Anomalies in spatial data
- Anomalies in spatial time-series data
- Anomalies in images
- Is anomaly detection fundamentally easier than density estimation?

Bibliography

- Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining* (pp. 413–422). Ieee. http://doi.org/10.1109/ICDM.2008.17
- Pevný, T. (2015). Loda: Lightweight on-line detector of anomalies. *Machine Learning*, (November 2014). http://doi.org/10.1007/s10994-015-5521-0
- Emmott, A., Das, S., Dietterich, T., Fern, A., & Wong, W.-K. (2015). *Systematic construction of anomaly detection benchmarks from real data*. http://arxiv.org/1503.01158v2
- Fern, A., Dietterich, T. G., Wright, R., Theriault, A., & Archer, D. W. (2018). Feedback-Guided Anomaly Discovery via Online Optimization. In *KDD 2018*.
- Liu, S., Garrepalli, R., Dietterich, T. G., Fern, A., & Hendrycks, D. (2018). Open Category Detection with PAC Guarantees. *Proceedings of the 35th International Conference on Machine Learning, PMLR*, *80*, 3169–3178.