Robust Machine Learning: Prediction with Confidence

Tom Dietterich
Professor (Emeritus)
Oregon State University

Outline

- The Need for Robust AI
- Predictions with Confidence
- Closed World Case
  - Point-wise Confidence Intervals
- Open World Case
  - Open Category Detection
  - Anomaly Detection Methods
- Dynamic World Case
  - Two-Sample Test
  - Covariate-Shift Correction

Self-Driving Cars

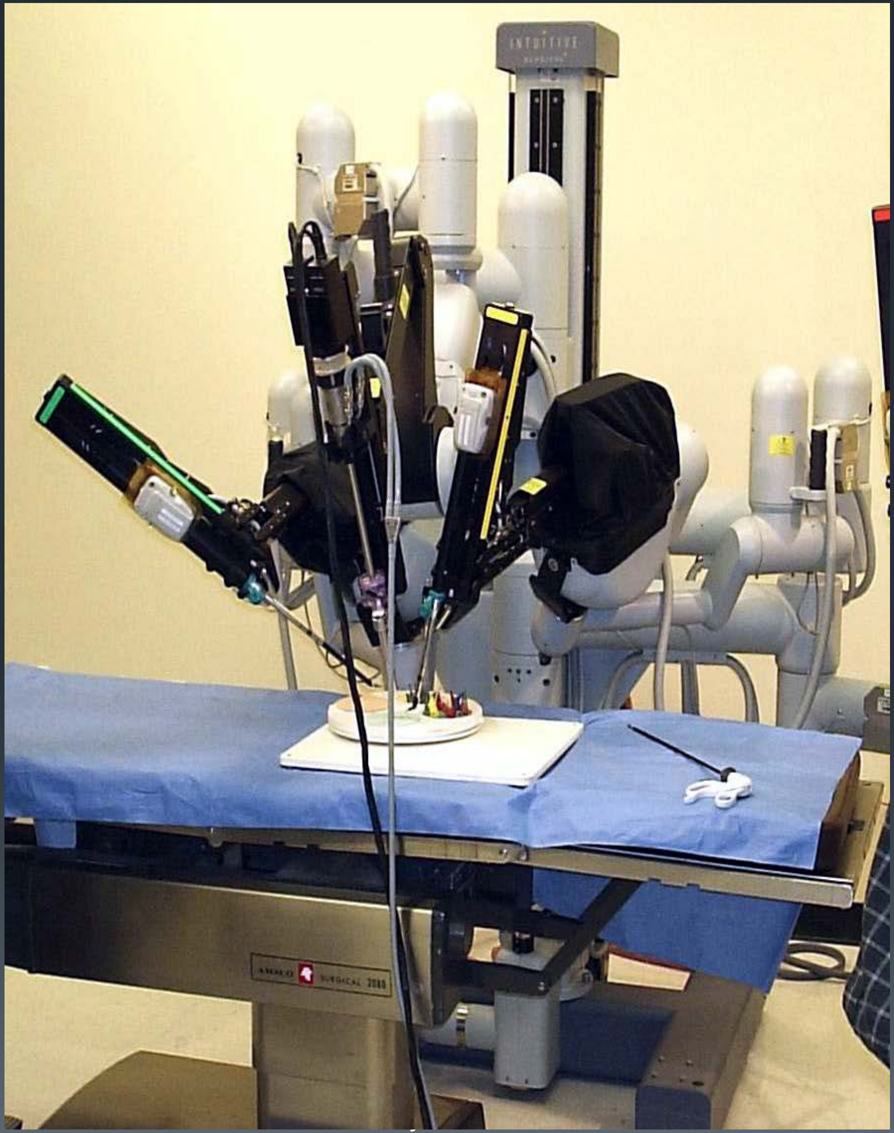
Credit: The Verge

Credit: delphi.com

Tesla AutoSteer

Credit: Tesla Motors

Automated Surgical Assistants



DaVinci

Credit: Wikipedia
CC BY-SA 3.0

AI Hedge Funds

WIRED

CADE METZ BUSINESS 01.25.16 7:00 AM

THE RISE OF THE ARTIFICIALLY INTELLIGENT HEDGE FUND

AI Control of the Power Grid

CONTROLLING THE POWER GRID WITH ARTIFICIAL INTELLIGENCE

02.07.2015

Credit: EBM Netz AG

DARPA Exploring Ways to Protect Nation's Electrical Grid from Cyber Attack

Effort calls for creation of automated systems to restore power within seven days or less after attack

Credit: DARPA

Autonomous Weapons

Northroop Grumman X-47B

Credit: Wikipedia

UK Brimstone Anti-Armor Weapon

Credit: Duch.seb - Own work, CC BY-SA 3.0

Samsung SGR-1

Credit: AFP/Getty Images

High-Stakes Applications Require Robust AI

- Robustness to
  - Human user error
  - Cyberattack
  - Misspecified goals
  - Incorrect models
  - Unmodeled phenomena

Basic Goal

- All machine learning models should understand their "region of competence"
- Given a query x_q
- Decide whether to output a classification or to abstain
- Guarantee that $1 - \epsilon$ of all individual predictions are correct with probability $1 - \delta$

Robust Predictions: Three Cases

- Closed World
  - fixed set of classes, iid training data
- Open World
  - novel classes at test time, iid training data
- Dynamic World
  - fixed set of classes, training data distribution changes over time

Closed World Predictions

- Train on training data
- Estimate error rate on validation data (n_{val} examples)
- If error on validation set is ϵ_{val} then with probability $1 - \delta$, the expected error ϵ on the test data set will be

$$\epsilon \leq \epsilon_{val} + \sqrt{\frac{\ln 1/\delta}{2n_{val}}}$$

- Note that this does not give us a point-wise guarantee
- Doesn't say what to do if ϵ is too big

Conformal Prediction

- Goal: Given query x_q output a *prediction set* $\Gamma(x_q)$ of class labels such that with probability $1 - \delta$ the true label is in that set
  - We want Γ to be as small as possible
  - If $|\Gamma| = 1$, then we have a unique predicted class
  - If $|\Gamma| > 1$, then the classifier cannot assign a unique class confidently

Basic Method

- Train a Classifier f
- Let $C(x, k)$ be the "conformity score" of x belonging to class k
  - $C(x, k)$ small means that x is unlikely to belong to k
  - $C(x, k)$ large means that x is likely to belong to k
- For each class k, let τ_k be a threshold (to be chosen)
- Given a test query x_q
  - Initialize $\Gamma(x) = \emptyset$
  - For each k
    - If $C\left(x_q, k\right) > \tau$ then add k to $\Gamma(x)$
  - Output $\Gamma(x)$

Conformity Scores

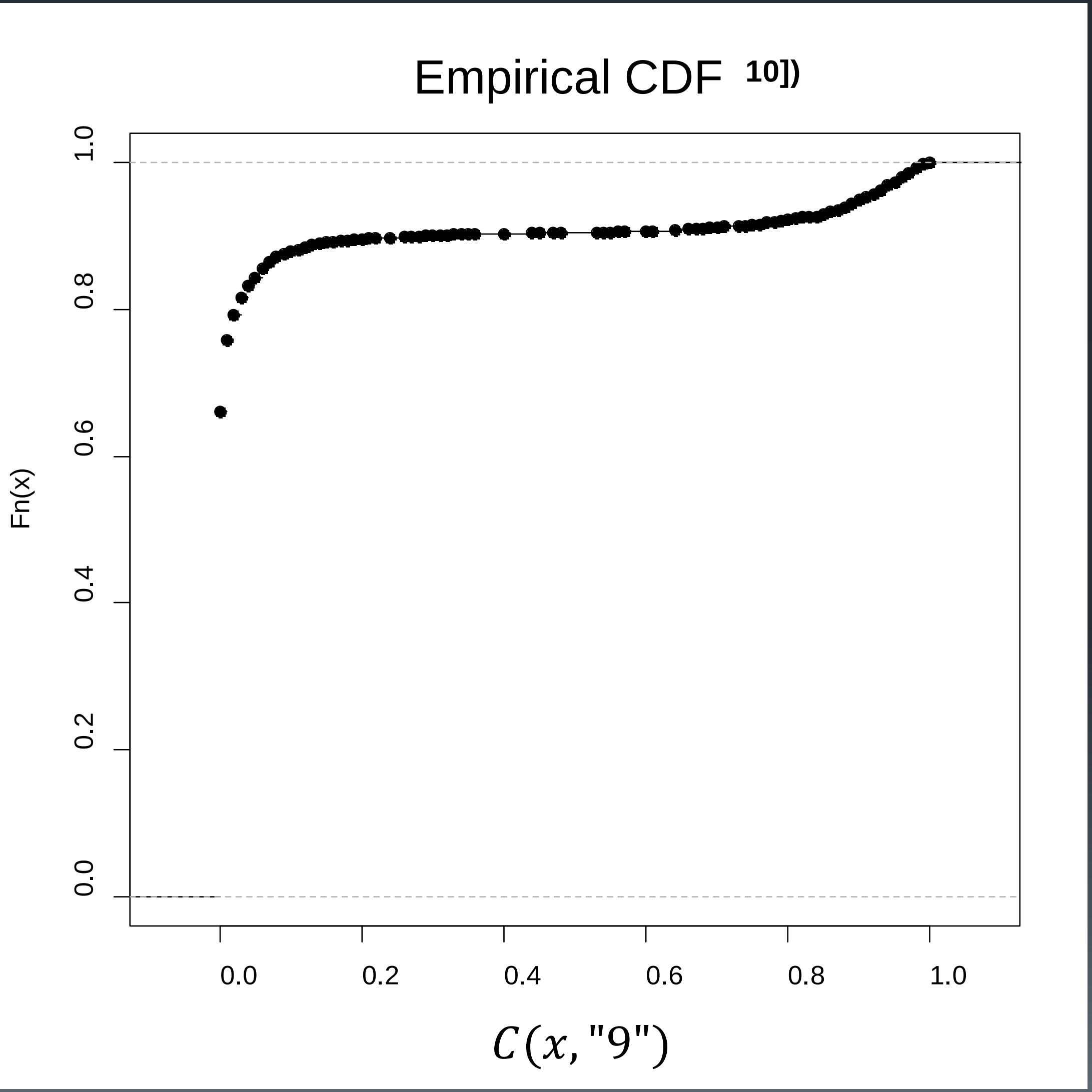
- Predicted probability $\hat{P}(y = k|x)$
- SVM Margin
- 1/distance to nearest neighbor in class k
- etc.

- Note that conformity scores do not need to be true probabilities or be calibrated in any way

Example: Pendigits + Random Forest

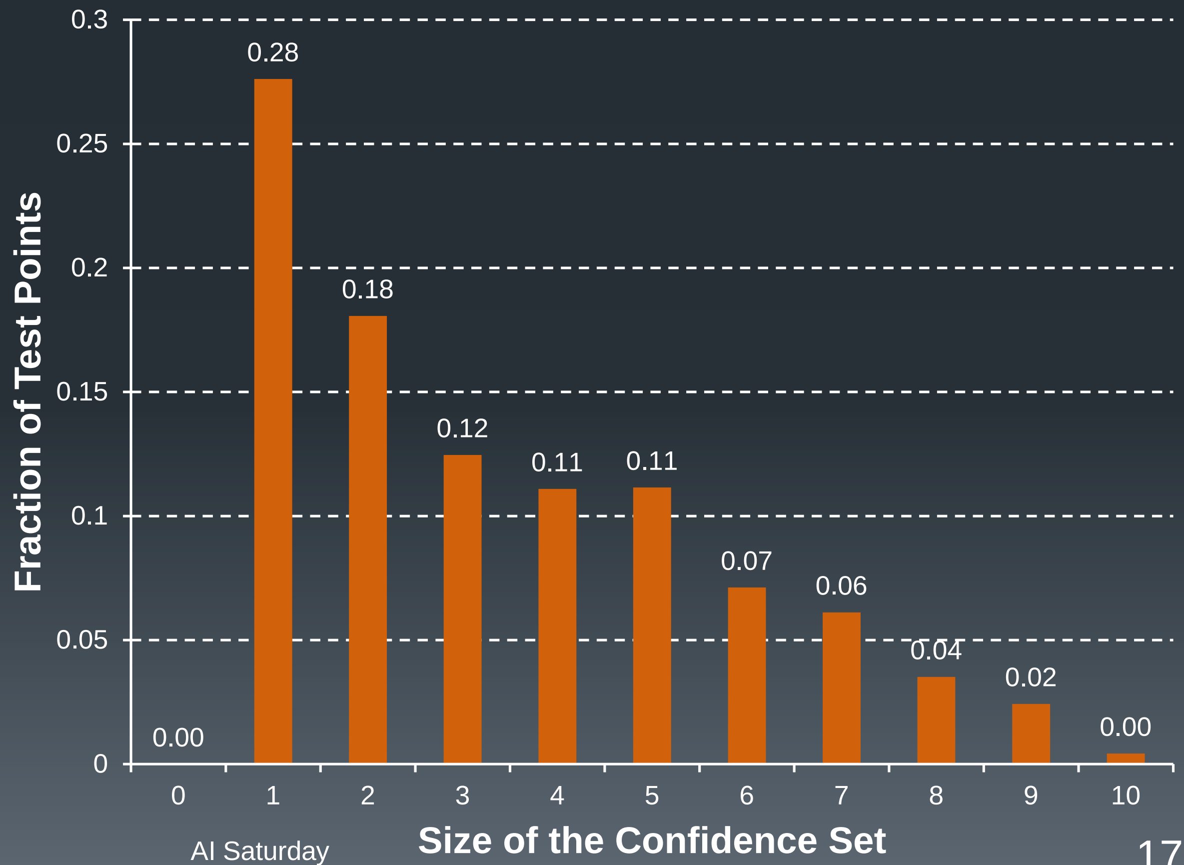
- Train a random forest on half of UCI Training Set
- Use the predicted class probability $P(y = k|x)$ as the conformity score $C(x, k)$
- Compute τ values using other half of Training Set
- Compute Γ on the Test Set

Cumulative Distribution Function for Class “9”



Pendigits Results

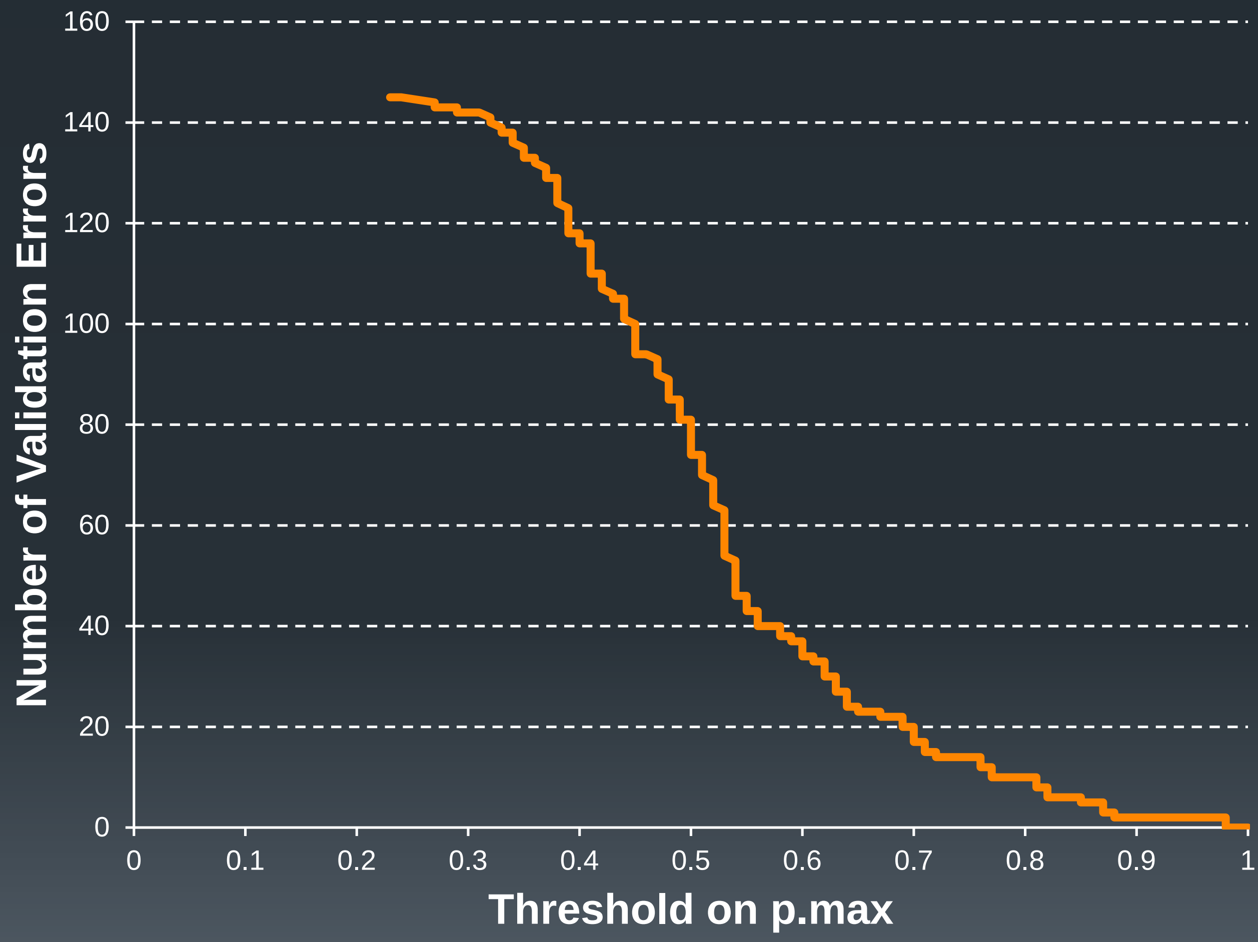
- All τ values were 0 (for $\epsilon = 0.001$)
- Probability $y \in \Gamma(x) = 0.9997$
- Abstention rate = 0.72
- Sizes of prediction sets Γ:



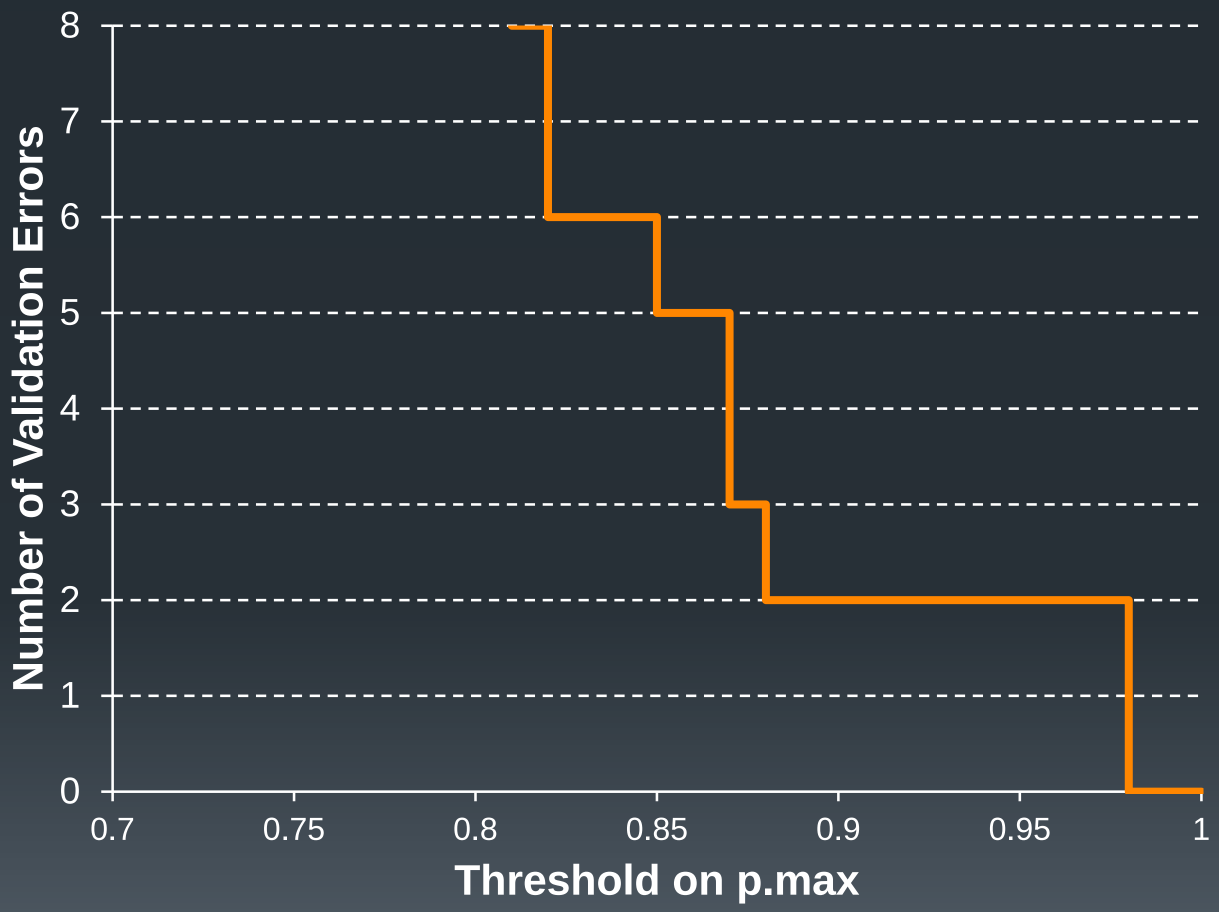
Abstention as Minimum Cost Decision Making

- A classifier with K classes and an abstention option has $K + 1$ possible actions.
- Let the cost of an incorrect prediction be 1 and the cost of an abstention be α
- Suppose we can obtain calibrated probabilities $P(y_q = k|x_q)$ from our classifier
- Let $p_{max} = \max_k P(y_q = k|x_q)$
- The probability of misclassification is $1 - p_{max}$, so the expected cost of misclassification is $1 - p_{max}$
- The cost of abstaining is α
- Therefore, we should abstain if $1 - p_{max} > \alpha$, or $p_{max} < 1 - \alpha$
- Use the validation set to choose a value of α that achieves a validation accuracy of $1 - \epsilon$

Calibrating $1 - \alpha$



Zoomed In: $1 - \alpha = 0.87$



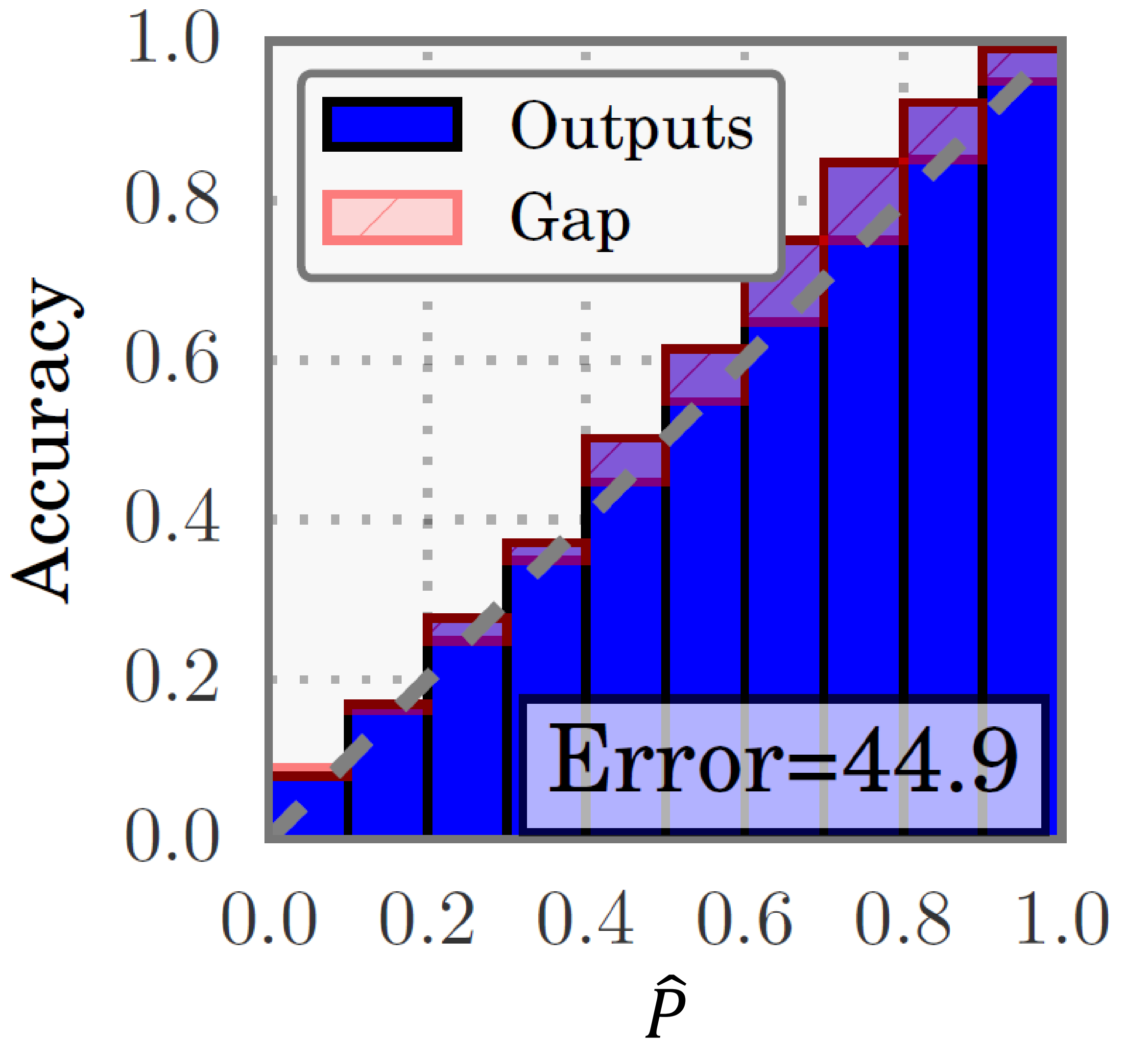
Test Set Results

- Probability of correct classification: 0.9987
- Abstention rate: 33.4%
  - [Conformal prediction was 0.72%]

Calibrated Probabilities

- What does it mean to say that a classifier gives calibrated probabilities?
- Let $X_p = \left\{x \middle| \hat{P}(\hat{y}|x) = p\right\}$
  - $\hat{y}$ is the predicted best class and
  - $\hat{P}(\hat{y}|x)$ is the predicted probability
- The classifier is well-calibrated if $P\left(\hat{y} = y \middle| x \in X_p\right) = p$
- In practice, we create bins and define X_b to be the data points that fall into bin b

Calibration Plot (LeNet on CIFAR100)

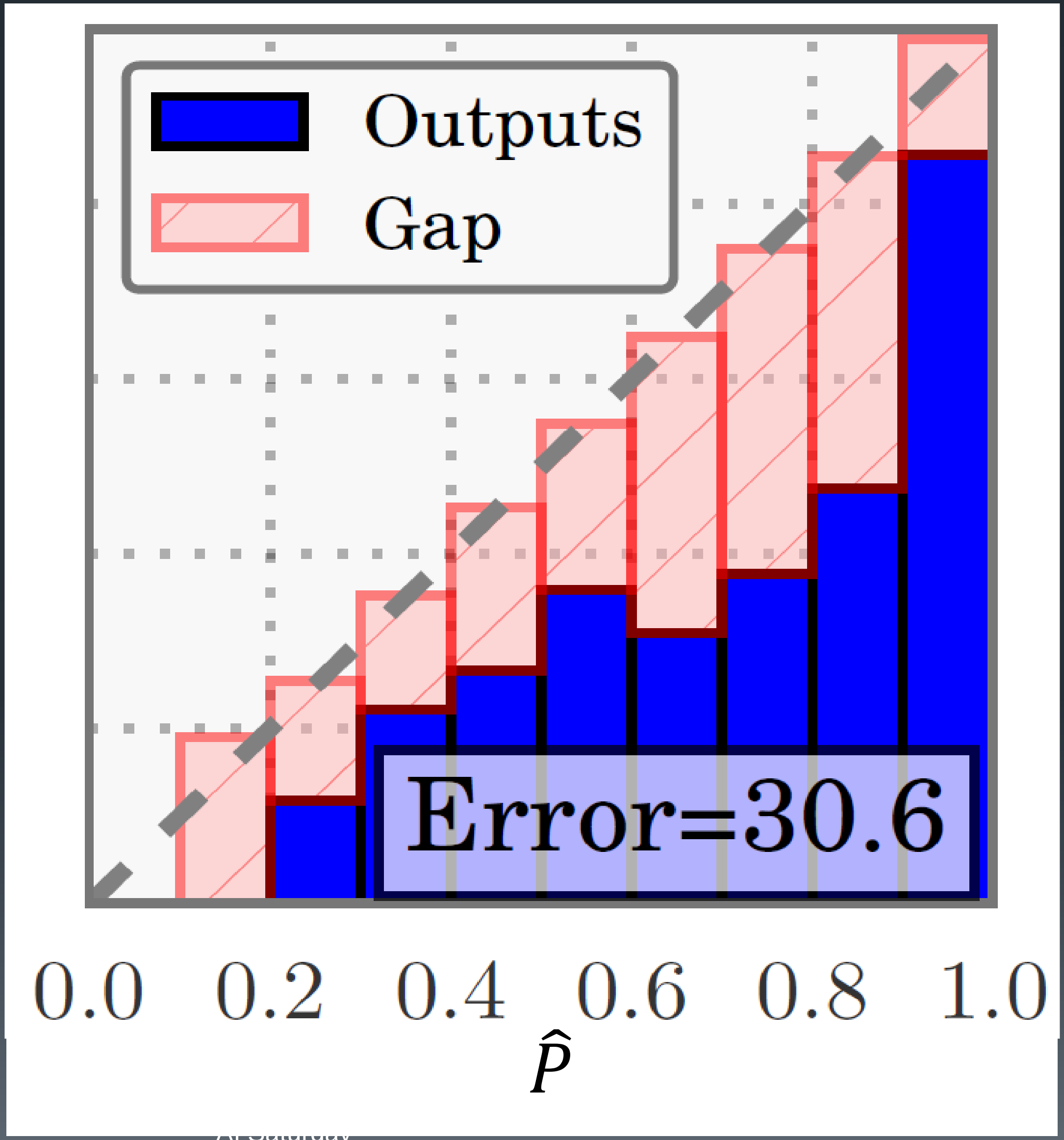


Calibration Studies

- Niculescu-Mizil & Caruana (ICML 2005)
  - Logistic regression, Random Forests, and multilayer perceptrons (1 hidden layer) give calibrated probabilities
  - Post-processing other scores (SVM margin, boosted trees, k-nn distance) is a good way to calibrate
- Guo, Pleiss, Sun, Weinberger (ICML 2017)
  - Large and wide nets (e.g., ResNet) are poorly calibrated
  - Tuning the softmax temperature is a good way to calibrate

Poor Calibration of ResNet (CIFAR 100)

- Lower error rate
- Much worse calibration
- Batch norm appears to contribute to the problem
- Increasing the SoftMax temperature fixes the problem



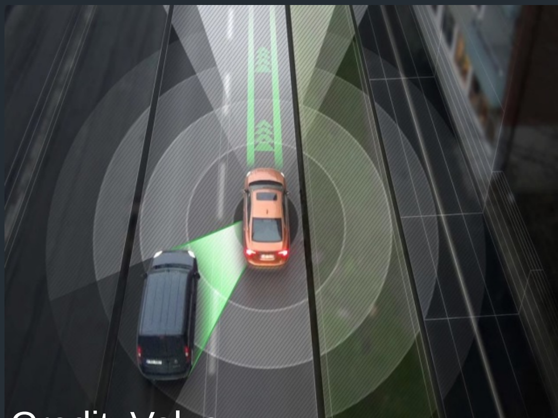
Outline

- The Need for Robust AI
- Predictions with Confidence
- Closed World Case
  - Point-wise Confidence Intervals
- Open World Case
  - Open Category Detection
  - Anomaly Detection Methods
- Dynamic World Case
  - Two-Sample Test
  - Covariate-Shift Correction

Open World Classification

- Training set contains classes 1, …, K
- Test set contains queries for those classes, AND queries for “alien” classes K+1, …, L
- How can we detect those aliens?

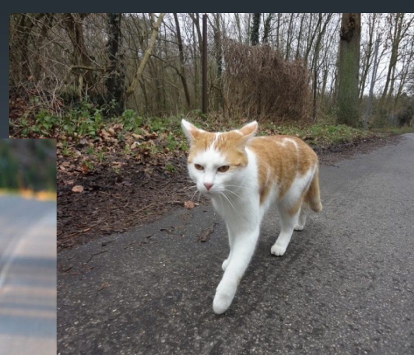
Example: Training a Self-Driving Car



Credit: Volvo

BBC
Technology

Volvo's driverless cars 'confused' by kangaroos



Example 2: Automated Counting of Freshwater Macroinvertebrates

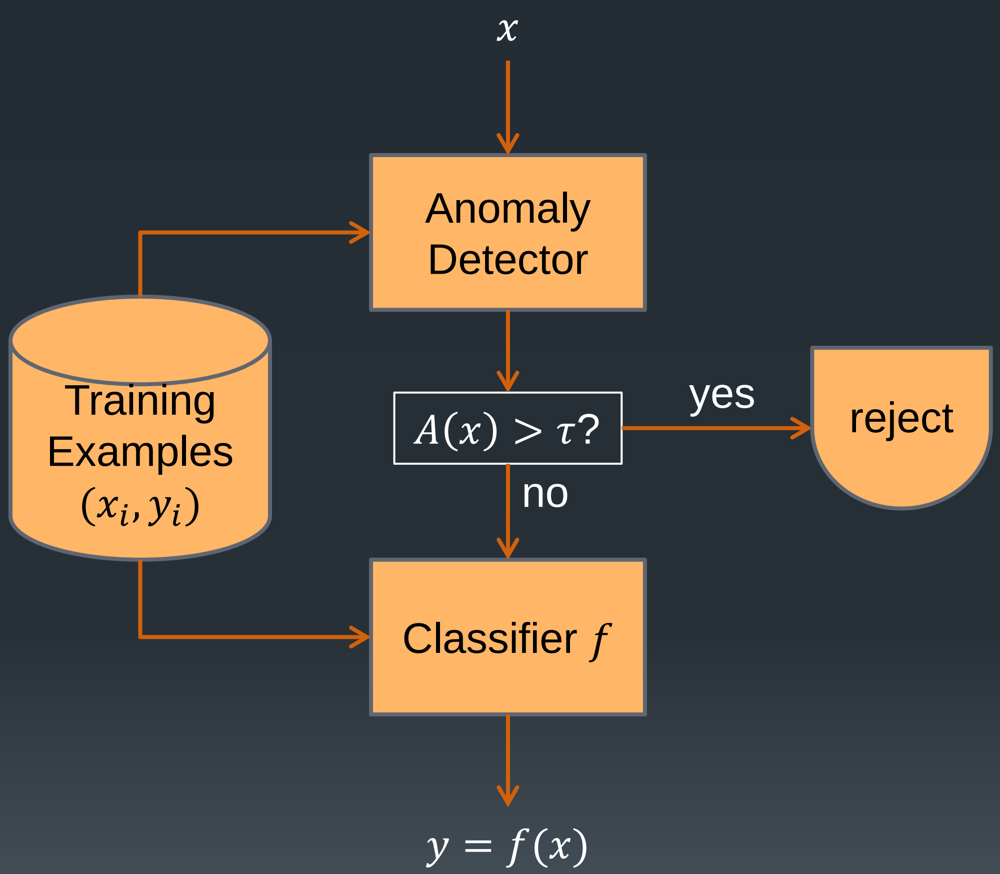
- Goal: Assess the health of freshwater streams
- Method:
  - Collect specimens via kicknet
  - Photograph in the lab
  - Classify to genus and species


www.epa.gov

Open Category Object Recognition

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

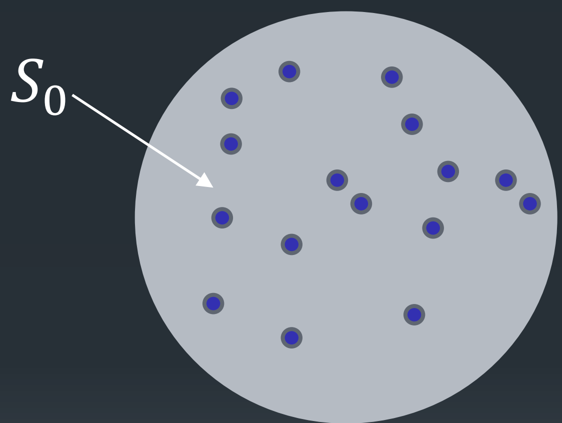
Prediction with Anomaly Detection



Training Data

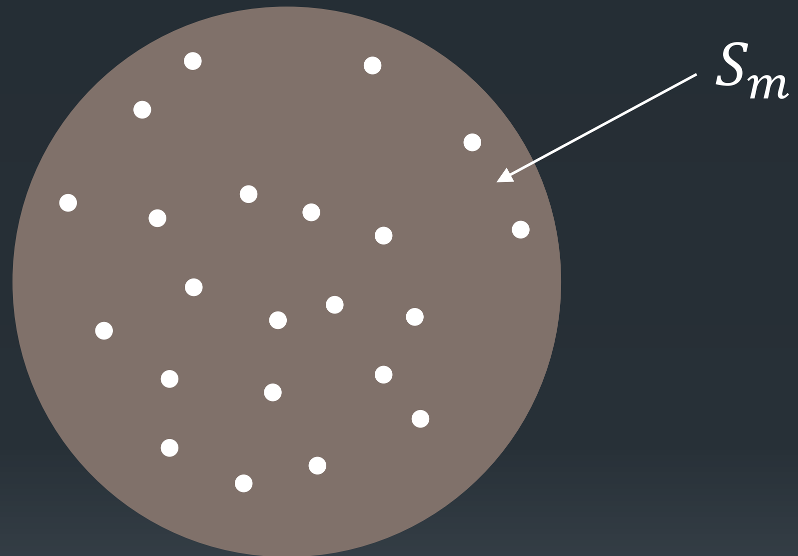
Labeled Clean Data

$P_0(x)$



Unlabeled Contaminated Data

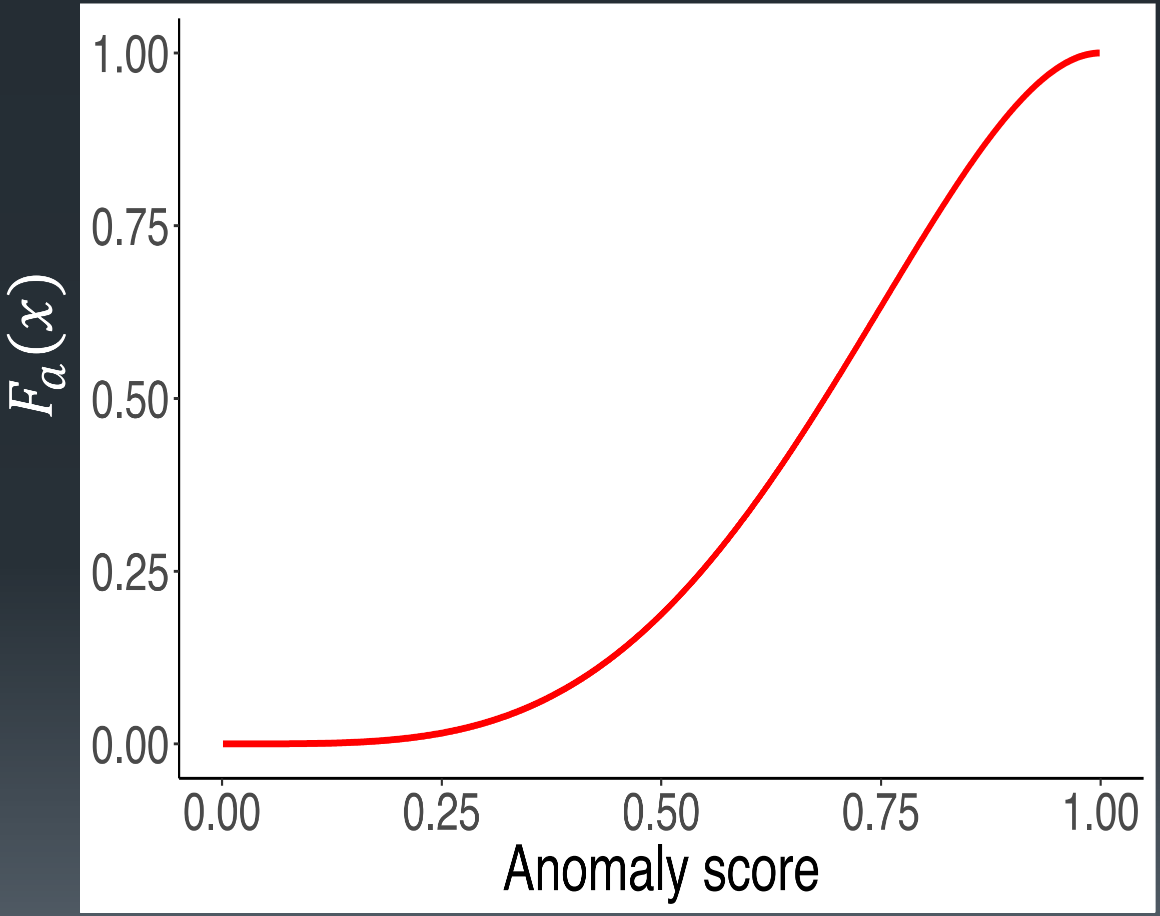
$P_m(x)$



Proportion of Aliens = α

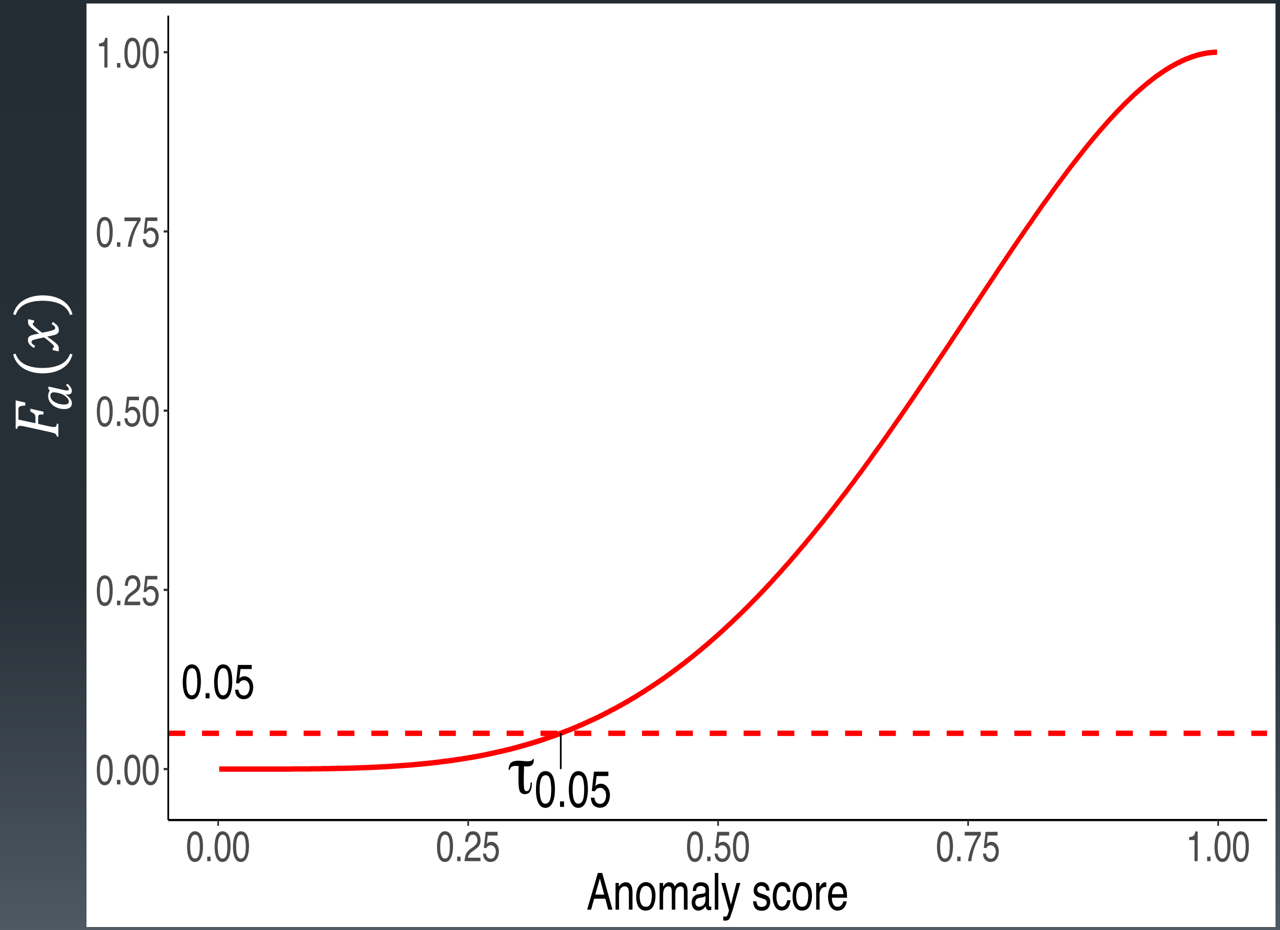
$$P_m(x) = (1-\alpha)P_0(x) + \alpha P_a(x)$$

We wish we had F_a, The Cumulative CDF of Alien Anomaly Scores



Want to have recall $= 1 - q$

Given F_a
Easy to choose τ for target quantile q



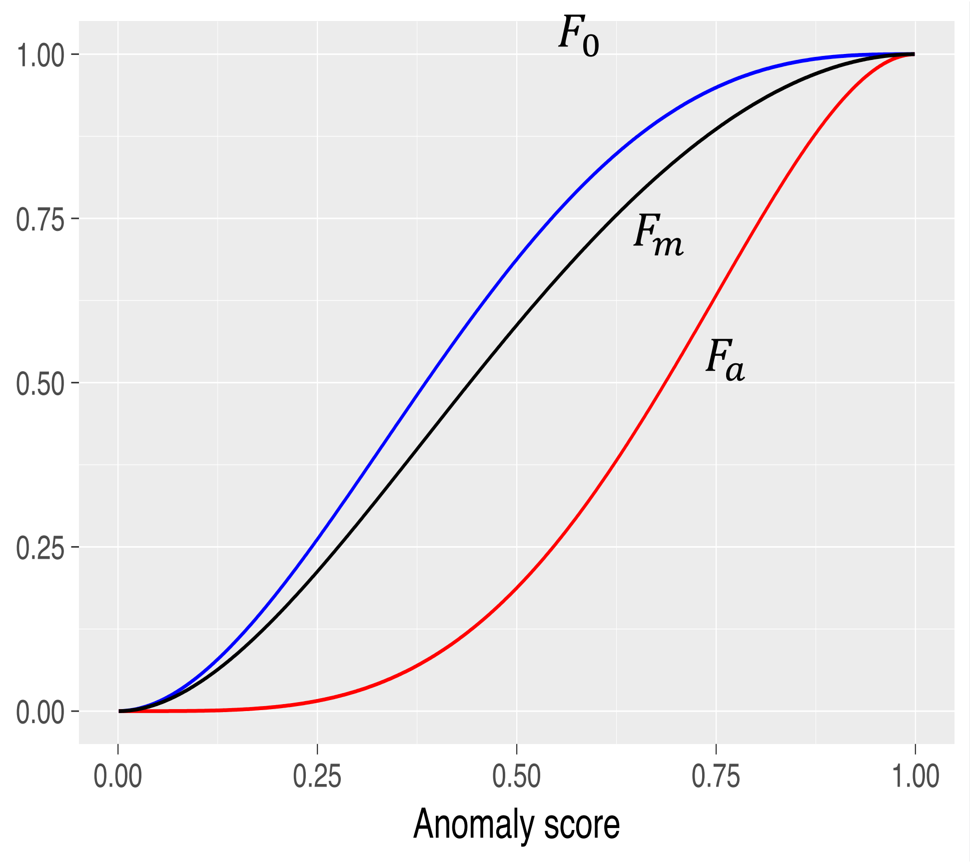
$q = 0.05$

$$P_m = (1 - \alpha)P_0 + \alpha P_a$$

implies that

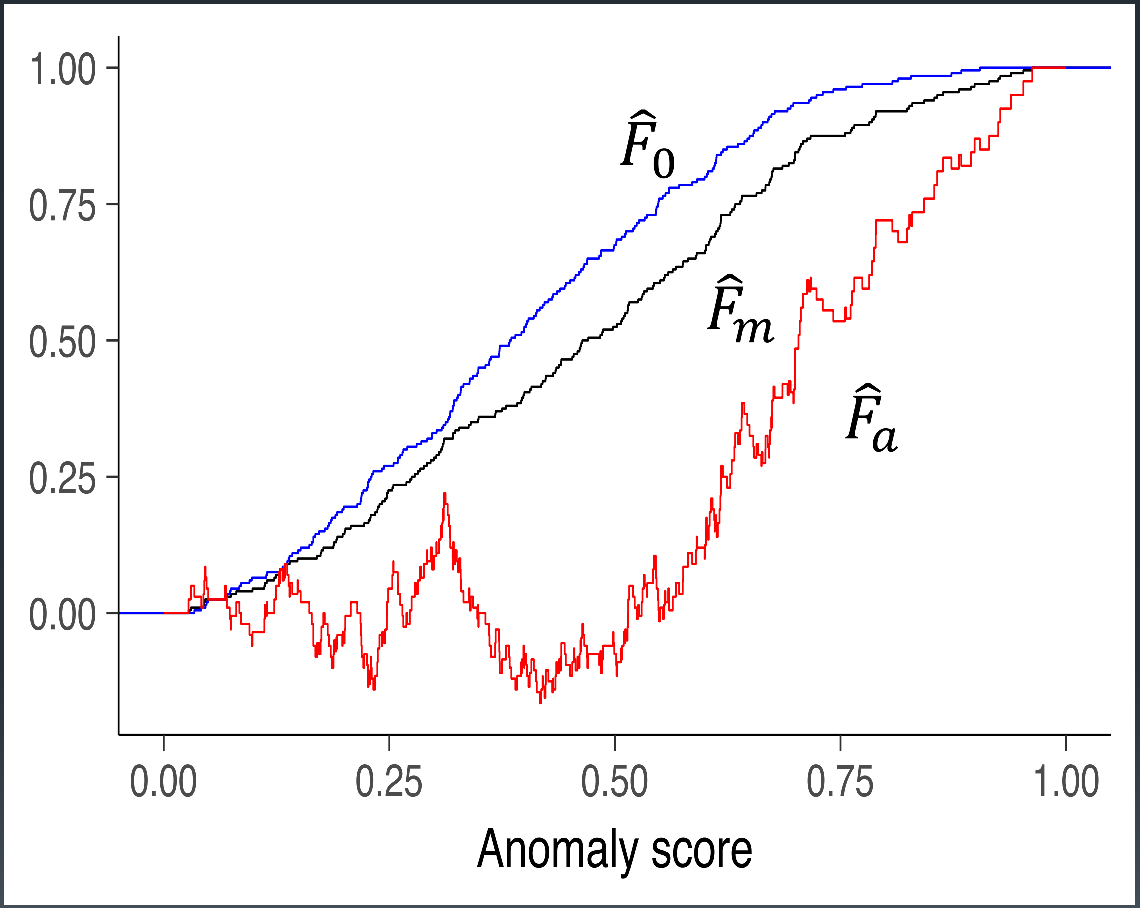
$$F_m(x) = (1 - \alpha)F_0(x) + \alpha F_a(x)$$

CDFs of Nominal, Mixture, and Alien Anomaly Scores



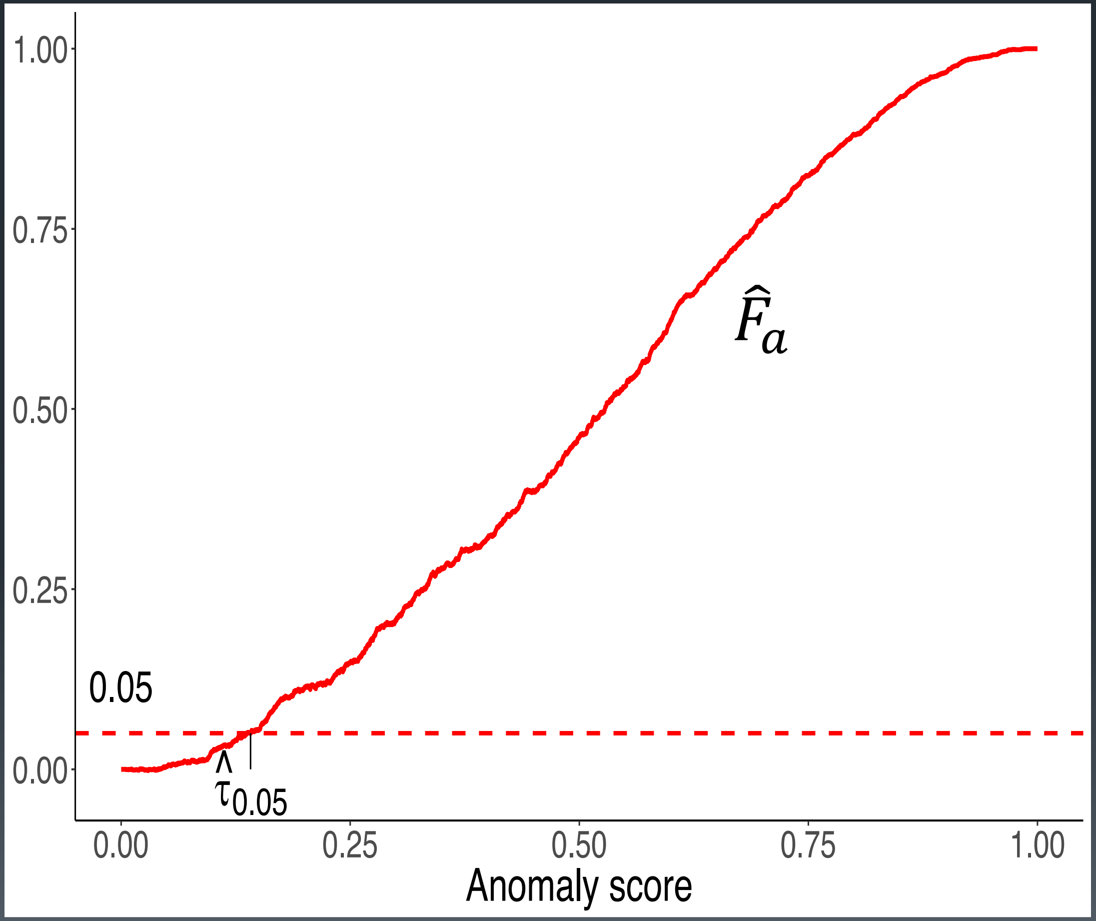
$$F_a(x) = \frac{F_m(x) - (1-\alpha)F_0(x)}{\alpha}$$

What We Have Are Empirical CDFs



$$\widehat{F}_a(x) = \frac{\widehat{F}_m(x) - (1-\alpha)\widehat{F}_0(x)}{\alpha}$$

We Use the Empirical Estimate $\hat{\tau}_{0.05}$



EstimateTau($\boldsymbol{S_0, S_m, q, \alpha}$)

1: Anomaly scores of S_0: $x_1, x_2, \cdots, x_n$

2: Anomaly scores of S_m: $y_1, y_2, \cdots, y_n$

3: Compute empirical CDFs $\hat{F}_0$ and $\hat{F}_m$.

4: Calculate $\hat{F}_a$ using

$$\hat{F}_a(x) = \frac{\hat{F}_m(x) - (1-\alpha)\hat{F}_0(x)}{\alpha}.$$

5: Output detection threshold

$$\hat{\tau}_q = \max\{u \in S : \hat{F}_a(u) \le q\},$$

where $S = \{x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_n\}$.

Theoretical Guarantee

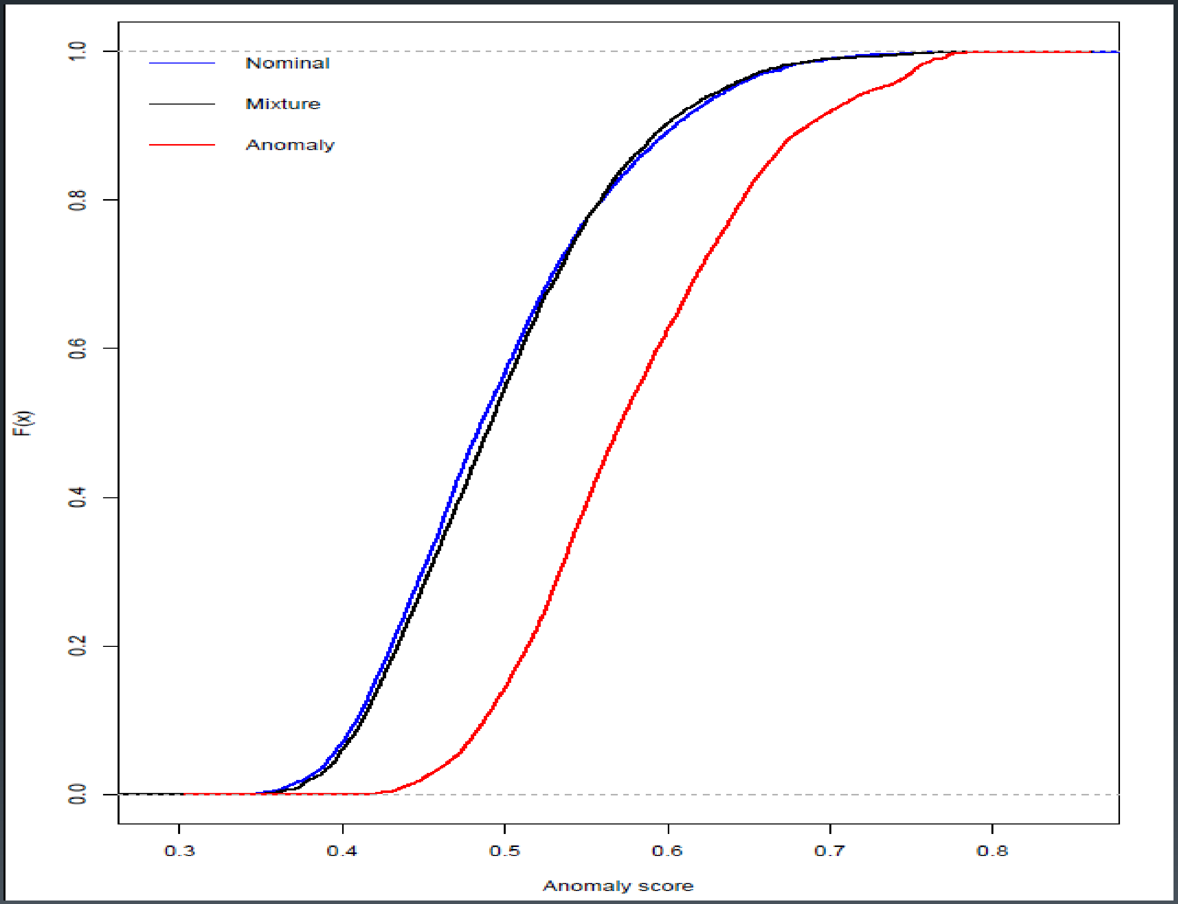
[Liu, Garrepalli, Fern, Dietterich, ICML 2018]

- Theorem: If

$$n > \frac{1}{2} \ln \frac{2}{1 - \sqrt{1 - \delta}} \left(\frac{1}{\epsilon}\right)^2 \left(\frac{2 - \alpha}{\alpha}\right)^2$$

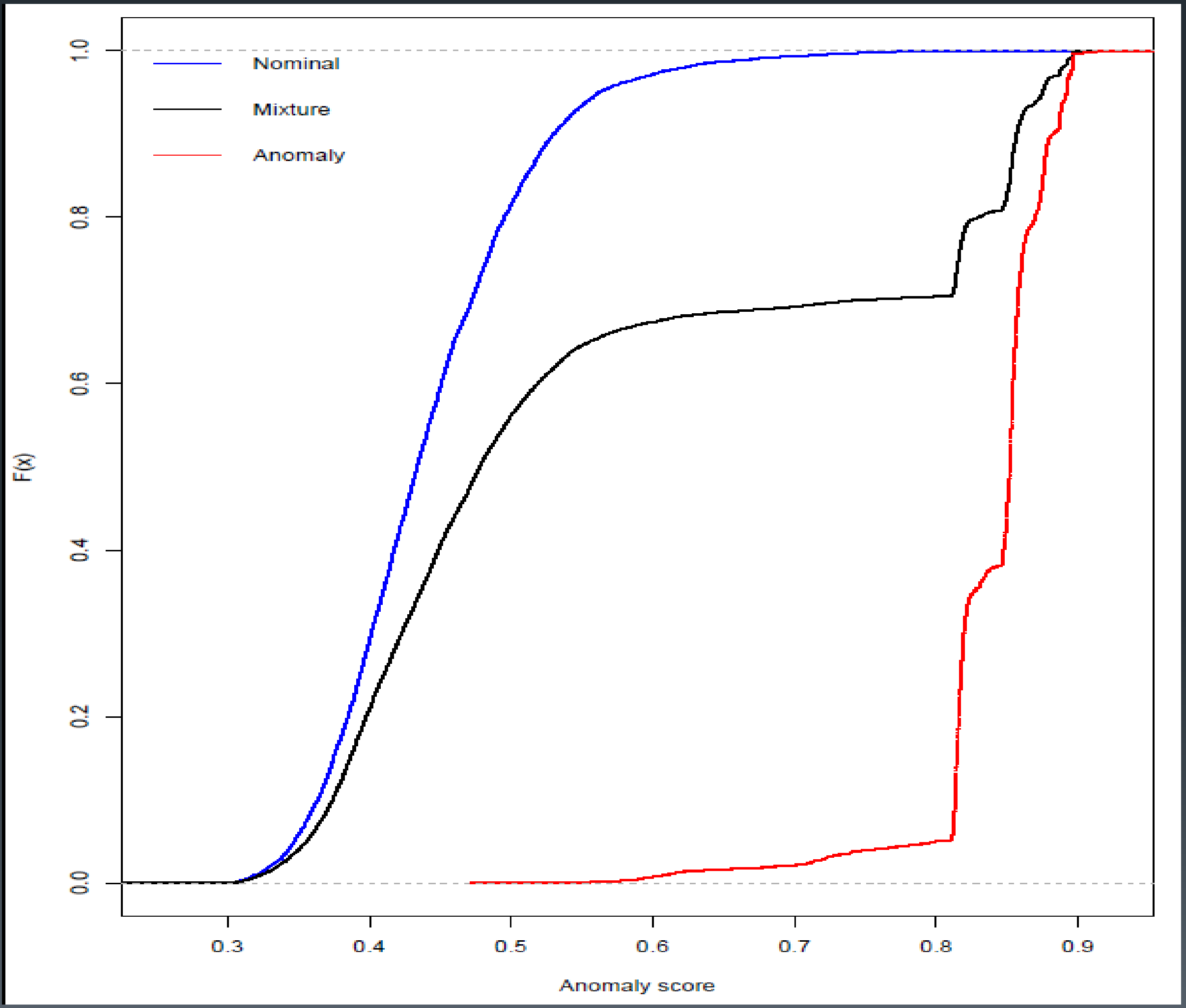
then with probability $1 - \delta$ the alien detection rate will be at least $1 - (q + \epsilon)$

Letter Recognition Dataset $\alpha = 0.1$

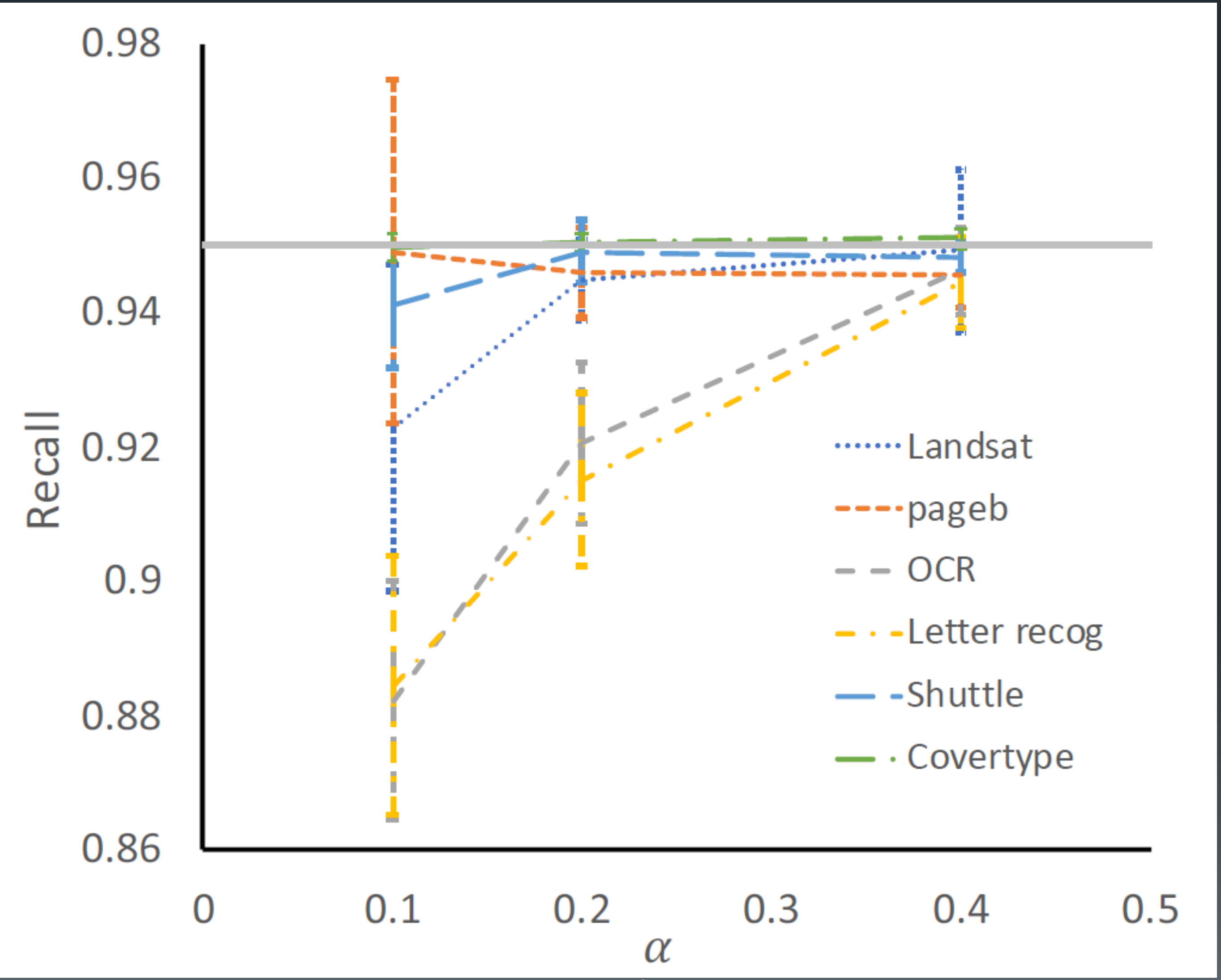


Anomaly Detector: Isolation Forest (1000 trees)

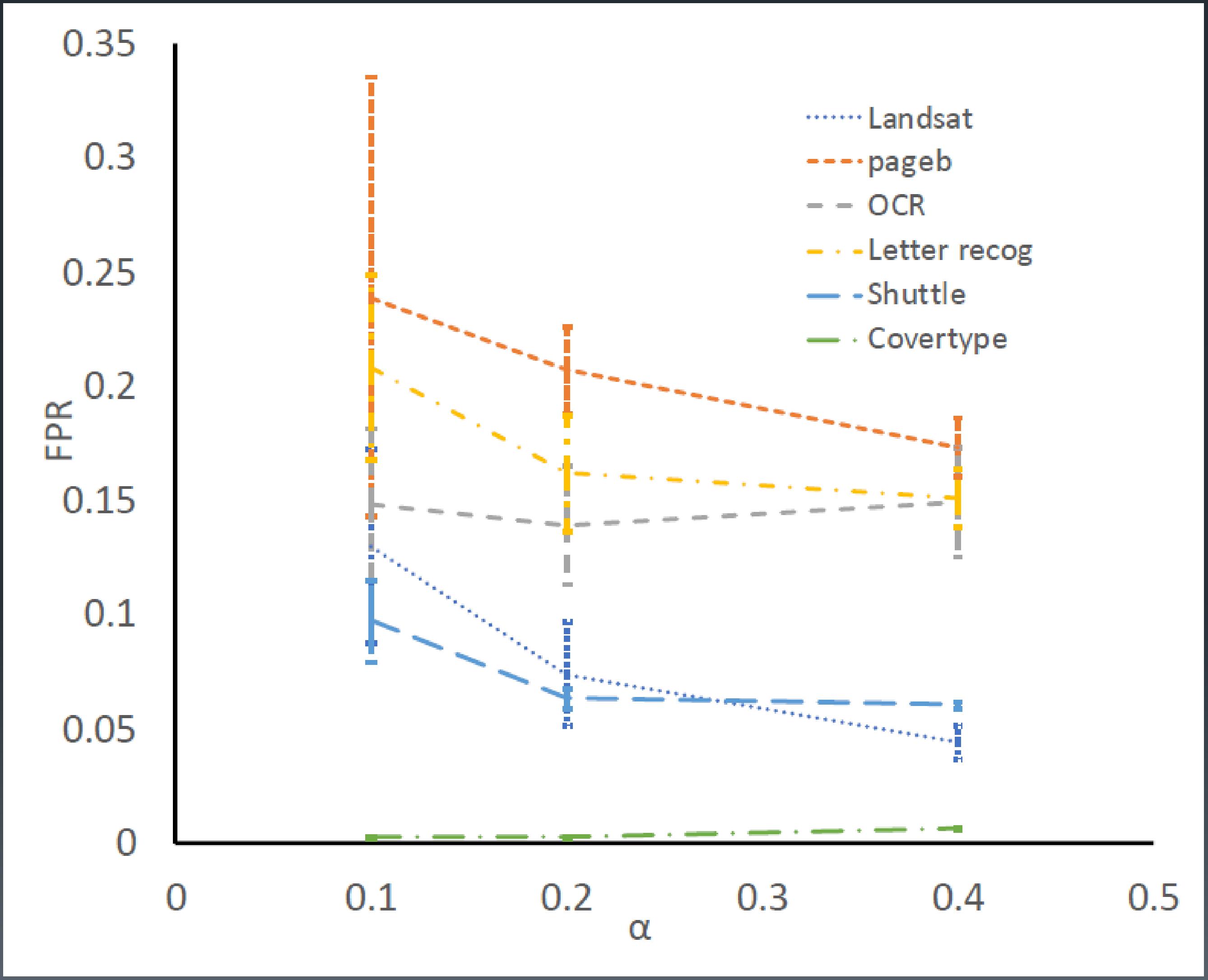
Shuttle $\alpha = 0.4$



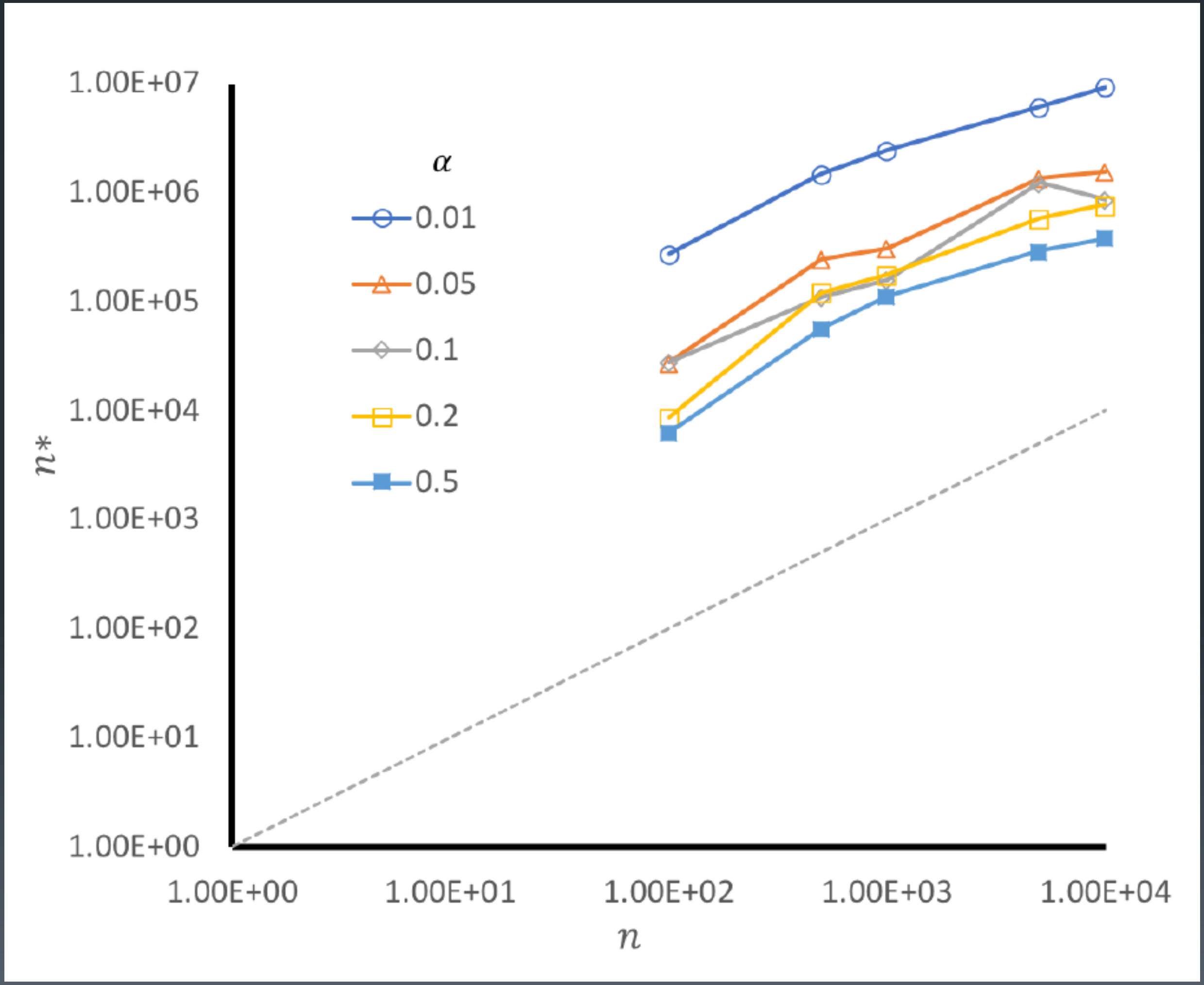
Recall (% of Aliens Detected)



False Alarm Rate (Abstention rate)



Tightness of Sample Size Bound



Conclusions

- The quality of the anomaly detector determines how well-separated F_0 and F_a are
  - If they are well-separated, then τ can detect the aliens without rejecting the nominals
- When α is small and q is small, S_m must be very large
  - We need enough data to estimate the q quantile of F_a
  - $\alpha = 0.01$ and $q = 0.01$ means that if $N_m = 10000$, there is expected to be only one alien point less than q. To get 100 such points, we need 1 million unlabeled examples
- The theoretical guarantee is conservative by 100x – 10000x
  - Can further theoretical work tighten the bound?

Further Notes

- This method assumes that underlying probability distributions of the data are not changing
  - The aliens are *not* generated by an adversary who could change the distribution
  - Methods are needed to handle such adversarial cases too
- Every deployed classifier should include an anomaly detector
  - Set the threshold manually based on synthetic or real anomaly data

For More Information…

- Full paper
  - http://proceedings.mlr.press/v80/liu18e.html
  - Liu, Garrepalli, Dietterich, Fern, Hendrycks (2018) Open Category Detection with PAC Guarantees
- Code and jupyter notebook
  - https://github.com/liusi2019/ocd

What is Anomaly Detection?

- Data $x_1, \ldots, x_N$, each $x_i \in \Re^d$
- Mixture of “nominal” points and “anomaly” points
- Anomaly points are generated by a different generative process than the nominal points
- Goal: Learn a function A such that
  - $A(x)$ is large for anomaly points
  - $A(x)$ is small for nominal points
- Metrics:
  - Error rate, area under ROC curve, precision (of anomaly detection) in top K predictions

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

Well-Defined Anomaly Distribution Assumption

- WDAD: the anomalies are drawn from a well-defined probability distribution
  - example: repeated instances of known machine failures

- The WDAD assumption is often risky
  - adversarial situations (fraud, insider threats, cyber security)
  - diverse set of potential causes (novel device failure modes)
  - user’s notion of “anomaly” changes with time (e.g., anomaly == “interesting point”)

Strategies for Unsupervised Anomaly Detection

- Let α be the fraction of training points that are anomalies
- Case 1: α is large (e.g., > 5%)
  - Fit a 2-component mixture model
    - Requires WDAD assumption
    - Mixture components must be identifiable
    - Mixture components cannot have large overlap in high density regions
- Case 2: α is small (e.g., 1%, 0.1%, 0.01%, 0.001%)
  - Anomaly detection via Outlier detection
    - Does not require WDAD assumption
    - Will fail if anomalies are not outliers (e.g., overlap with nominal density; tightly clustered anomaly density)
    - Will fail if nominal distribution has heavy tails

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

19 Selected Data Sets

Steel Plates Faults
Gas Sensor Array Drift
Image Segmentation
Landsat Satellite
Letter Recognition
OptDigits
Page Blocks
Shuttle
Magic Gamma
Skin

Waveform
Yeast
Abalone
Communities and Crime
Concrete Compressive Strength
Wine
Year Prediction
Spambase
Particle

Systematic Variation of Relevant Aspects

- Point difficulty: How deeply are the anomaly points buried in the nominals?
  - Fit supervised classifier (kernel logistic regression)
  - Point difficulty: $P(\hat{y} = "nominal" | x)$ for anomaly points
- Relative frequency:
  - sample from the anomaly points to achieve target values of α
- Clusteredness:
  - greedy algorithm selects points to create clusters or to create widely separated points
- Irrelevant features
  - create new features by random permutation of existing feature values
- Result: 25,685 Benchmark Datasets

Metrics

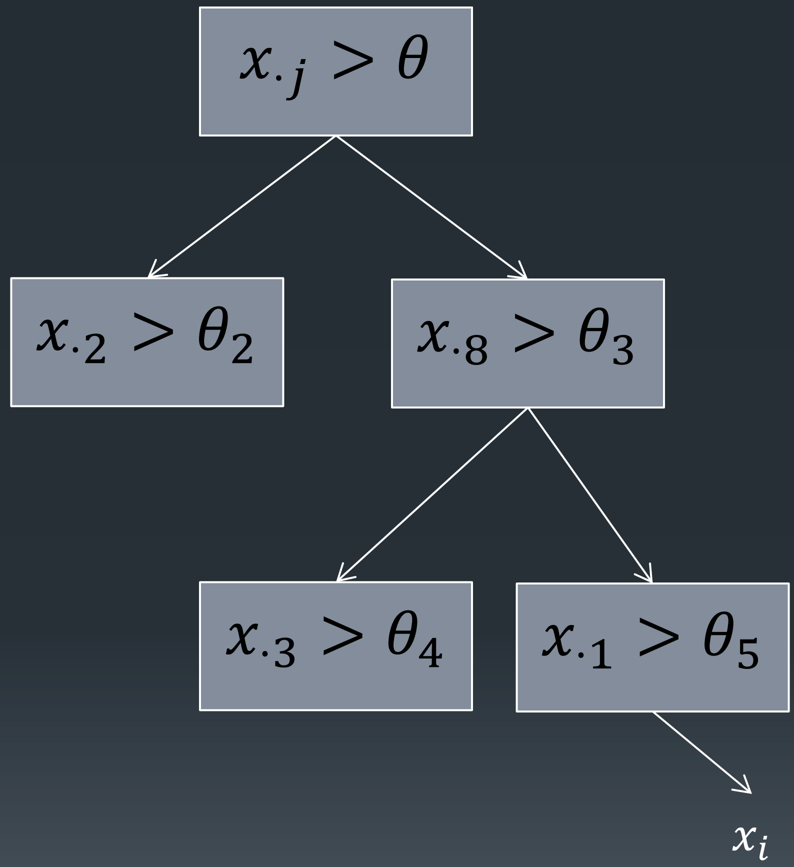
- AUC (Area Under ROC Curve)
  - ranking loss: probability that a randomly-chosen anomaly point is ranked above a randomly-chosen nominal point
  - transformed value: $\log \frac{AUC}{1-AUC}$
- AP (Average Precision)
  - area under the precision-recall curve
  - average of the precision computed at each ranked anomaly point
  - transformed value: $\log \frac{AP}{\mathbb{E}[AP]} = \log LIFT$

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)

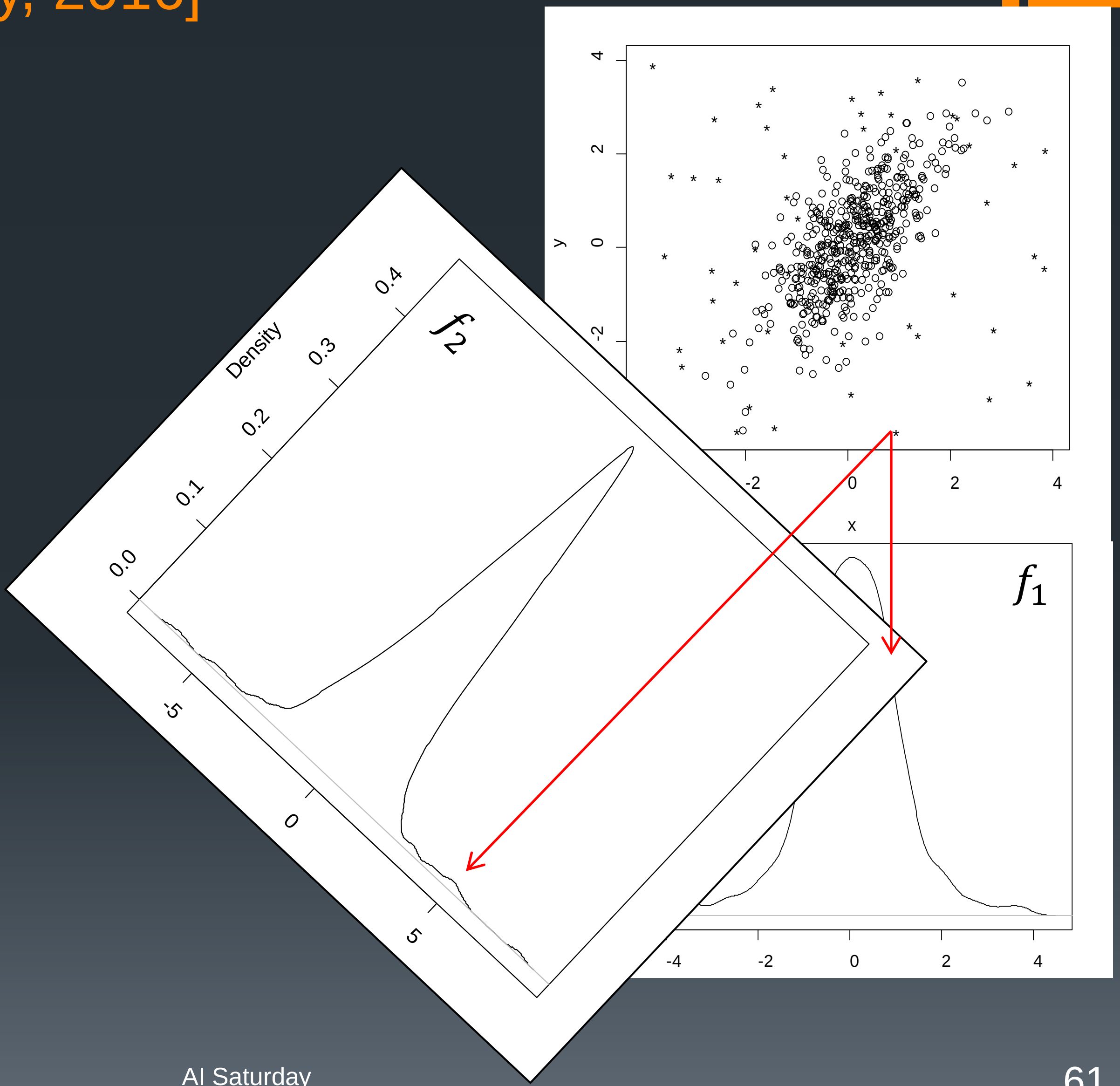
Isolation Forest [Liu, Ting, Zhou, 2011]

- 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
  - $score(x_i) = 2^{-\left(\frac{\bar{d}(x_i)}{r(x_i)}\right)}$
    - $r(x_i)$ is the expected depth



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

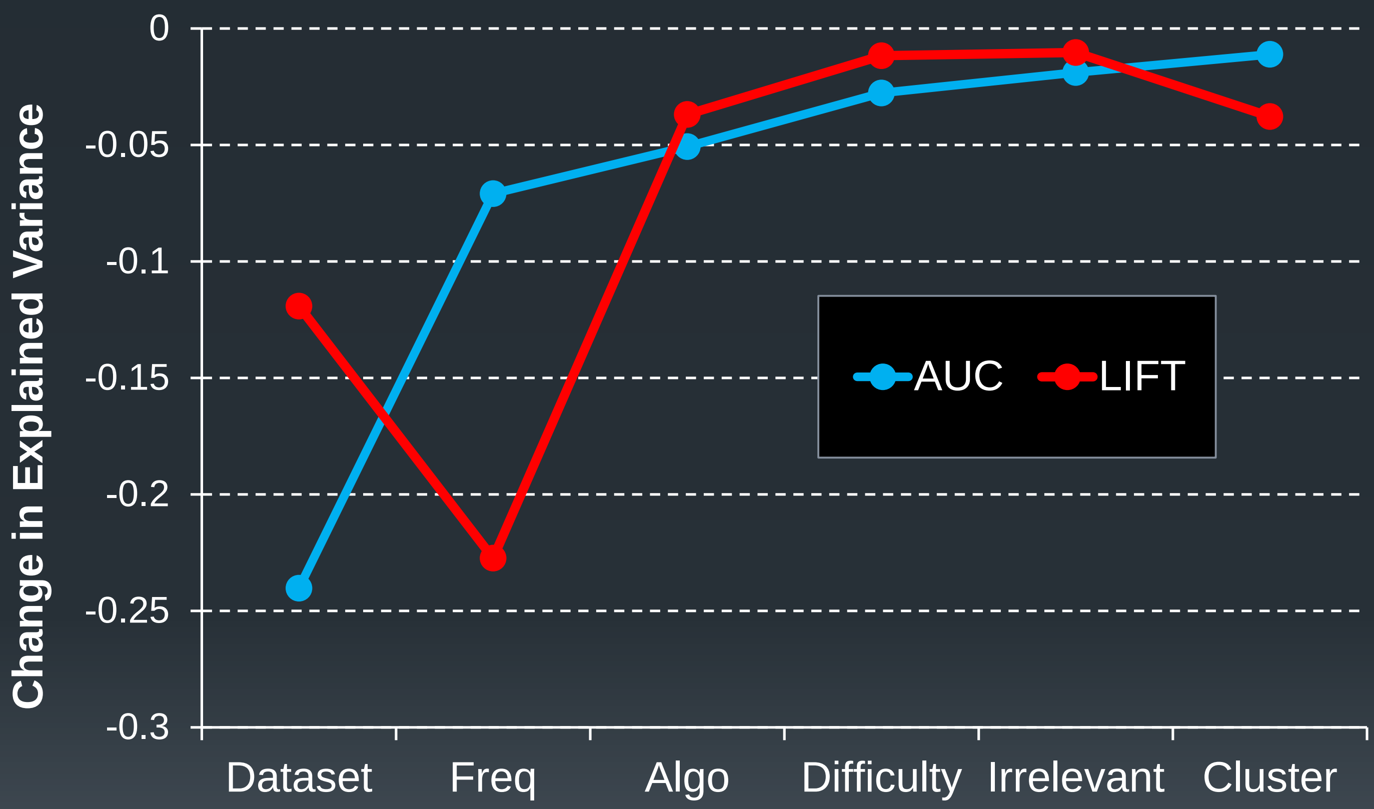
- $\Pi_1, \ldots, \Pi_M$ set of M sparse random projections
- $f_1, \ldots, f_M$ corresponding 1-dimensional density estimators
- $S(x) = \frac{1}{M}\sum_m -\log f_m(x)$ average "surprise"



Statistical Analysis

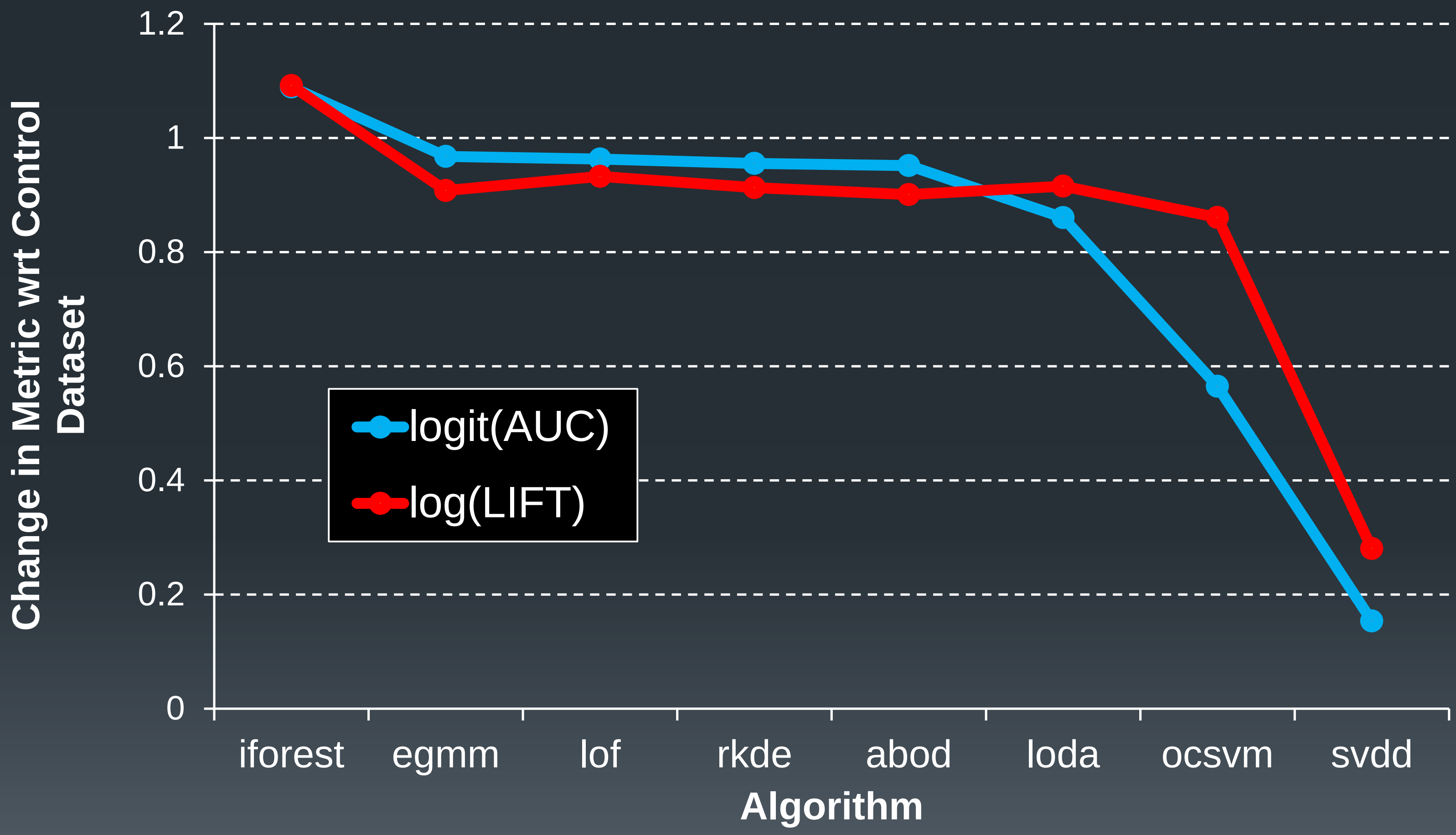
- Linear ANOVA
  - $metric \sim rf + pd + cl + ir + mset + algo$
    - rf: relative frequency
    - pd: point difficulty
    - cl: normalized clusteredness
    - ir: irrelevant features
    - mset: “Mother” set
    - algo: anomaly detection algorithm
- Validate the effect of each factor
- Assess the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

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

For Further Study

- Oregon State iForest implementation
  - https://github.com/tadeze/osu_iforest
  - R and python bindings; C++ implementation

- iForest python implementation
  - sklearn.ensemble.IsolationForest

Outline

- The Need for Robust AI
- Predictions with Confidence
- Closed World Case
  - Point-wise Confidence Intervals
- Open World Case
  - Open Category Detection
  - Anomaly Detection Methods
- Dynamic World Case
  - Two-Sample Test
  - Covariate-Shift Correction

Dynamic Worlds

- World changes between train and test
- Detecting changes
- Compensating for changes
  - Covariate Shift
  - Domain Adaptation

Detecting Changes in the Data Distribution

- Standard assumption: $(x, y) \sim P(x, y) = P(x)P(y|x)$
- Covariate Shift: $P(x)$ changes but $P(y|x)$ is unchanged
- Data Set Shift: $P(x, y)$ changes

Detecting Data Set Shift: The Problem

- Given
  - Data set 1: $S = x_1, \ldots, x_N$ drawn iid from P
  - Data set 2: $S' = \ x'_1, \ldots, x'_{N'}$ drawn iid from P'
- Question:
  - Do S and S' come from the same distribution? (i.e., does $P = P'$?)

Data Set Shift Method 1

- Train a classifier to predict whether points come from S or S'
- If the classifier does significantly better than random, then there is dataset shift
- Procedure
  - Divide S into S_{train} and S_{test}
  - Divide S' into S'_{train} and S'_{test}
  - Define a class label
    - $y = 1$ for data from S'
    - $y = 0$ for data from S
  - Train classifier f on $S_{train} \cup S'_{train}$
  - Test f on $S_{test} \cup S'_{test}$

Data Set Shift Method 2

- Apply a 2-Sample Test to check whether S and S' come from different distributions
- Kernel 2-Sample Test
  - Define a similarity kernel $k(x, x')$ that is 1 if $x = x'$ and decreases to 0 as x and x' are more different
  - Standard kernel: $\exp -\|x - x'\|^2/\sigma^2$
  - Usually standardize all features $x^j := (x^j - \bar{x}^j)/s^j$ where $\bar{x}^j$ is the sample mean and s^j is the sample standard deviation for feature j
- $MMD = \mathbb{E}_{x,x' \in S}[k(x, x')] - 2E_{x \in S, x' \in S'}[k(x, x')] + \mathbb{E}_{x,x' \in S'}[k(x, x')]$
  - Can compute a cutoff value under the null hypothesis that $P = P'$ and reject in favor of the alternative that $P \neq P'$.

Details

- Paper
  - A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, A. Smola. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, **13**: 723-773.
- Python implementation in the shogun toolbox
  - http://shogun-toolbox.org/notebook/latest/mmd_two_sample_testing.html

Covariate Shift Correction

- Let $\mu(x) = \frac{P'(x)}{P(x)}$. This is the *density ratio*
- For each training example $(x_i, y_i) \in S$, assign a weight $w_i = \mu(x_i)$
- Fit a classifier to this weighted data (most learning algorithms can handle weighted data)
- Rationale:
  - If we sampled data points $x \sim \mu(x)P(x)$ this is equivalent to sampling from $P'(x)$ because

$$\mu(x)P(x) = \frac{P'(x)}{P(x)}P(x) = P'(x)$$

Estimating μ

- Suppose we have trained f using Method 1 and it provides calibrated probabilities
  - $f(x) = P(x \in S'|x)$
  - $1 - f(x) = P(x \in S|x)$
- then we can compute
  - $\mu(x) = \frac{f(x)}{1-f(x)}$

Logistic Regression

- If we use logistic regression to learn f for Method 1, the logistic regression model is

$$\log \frac{f(x)}{1 - f(x)} = \sum_j \beta_j x^j + \beta_0 = \log \mu(x)$$

- Hence, $\mu(x) = \exp\left[\sum_j \beta_j x^j + \beta_0\right]$
- This also works when we apply logistic rescaling to SVMs, boosted trees, etc.

Relative Density Ratio Estimation

- Often the density ratio estimates can be very large, which leads to unstable function fitting
- Yamada et al. (2011) propose a more robust, smoother estimator:
- $\mu_\alpha(x) = \frac{P'(x)}{\alpha P'(x) + (1-\alpha)P(x)}$
- This ensures that the density ratio is never larger than $1/\alpha$
- They show experimental that $\alpha = 0.5$ is a good choice

Comparison of Plain vs. Relative Density Ratio Estimation

Summary

- Closed World Accuracy
  - Conformal Prediction (strong theoretical guarantees)
  - Rejection Threshold with Calibrated Class Probabilities
- Open World Accuracy
  - Theoretical method is very conservative
  - Setting the detection threshold requires guesswork (or labeled anomaly data)
- Change Detection and Correction
  - Change detection by classification and by 2-sample test
  - Correction by instance weighting

Acknowledgements

- Partially supported by
  - DARPA Contract W911NF-11-C-0088
  - DARPA Contract FA8650-15-C-7557
  - NSF Grant 1514550
  - FLI program FLI-RFP-AI1, grant number 2015-145014
  - Gift from Huawei, Inc.