Safe Behavior in Open Worlds: AI Methods for Learning and Acting in the Presence of Unknown Unknowns

Alan Fern
Tom Dietterich
Jeffrey Juozapaitis
Risheek Garrepalli

Oregon State University

Project Goals

- Novel classes in supervised learning
  - Training data from classes $\mathcal{Y}\downarrow train = \{1,...,M\}$
  - Test data from classes $\{1, ...,M,M+1,...,M+N\}$
  - **Detect** when test query does not belong to $\mathcal{Y}\downarrow train$
- Novel states and behaviors in imitation learning
  - Let π be a policy learned from $\mathrm{T}\downarrow train$, a set of training trajectories in an MDP
  - When following π, **detect** when the current state is novel
- Goal: PAC Guarantees on Detection of Unknown Unknowns

Methods

- Extend Conformal Prediction to handle Unknown Unknowns in supervised learning
  - Compute non-conformity measures using anomaly detection algorithms
- Extend Conformal Prediction to imitation learning
  - Via reduction to supervised learning

Conformal Prediction

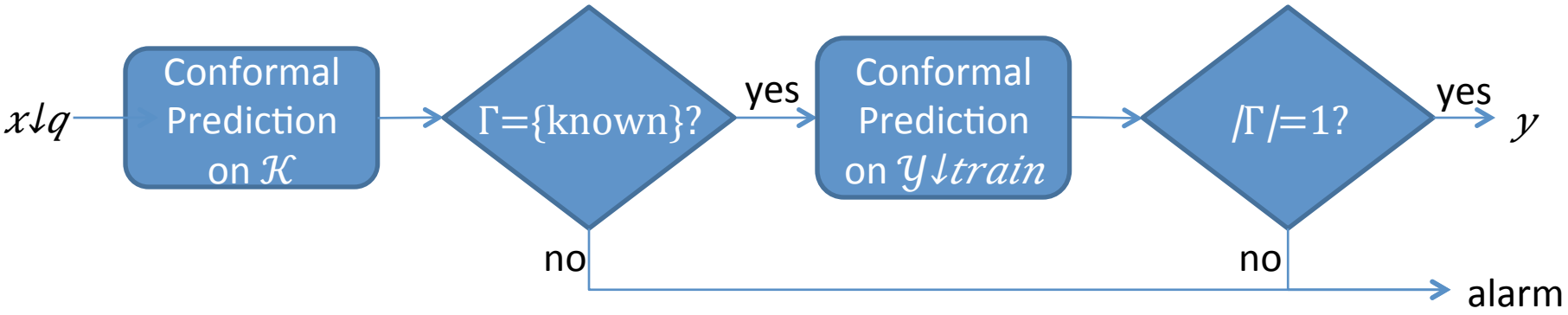
(Vovk, Gammerman & Shafer, 2005)

- Notation
  - R training set $\{(x_i, y_i)\}_{i=1}^{n}$
  - f learned classifier
  - x_q a query for which the (hidden) true class is y_q
  - $y_q = f(x_q)$ predicted class
  - $A(R, x_q, y_q)$ non-conformity measure. How surprising would it be to label x_q as y_q?
- Conformal predictor outputs a prediction set
  - $\Gamma(x_q, \epsilon; R) \subseteq \mathcal{Y}_{train}$ such that with probability $1-\epsilon$ $y_q \in \Gamma(x_q, \epsilon; R)$

Typical Conformal Prediction Algorithm

- Divide training data into subtraining and validation sets
- For each class label, compute the empirical distribution of non-conformity scores on the validation set
- Let $\tau\downarrow y$ be the $1-\epsilon$ quantile of the non-conformity scores for class label y
- $\Gamma(x,\epsilon;R)=\{y:A(R,x\downarrow q,y)\leq\tau\downarrow y\}$
- This is a discrete confidence interval

Key Idea: Series Composition



- Employ anomaly detection algorithms to compute the non-conformity score $A(R, x\downarrow q, y)$, where $y \in \mathcal{K} = \{\text{known}, \text{unknown}\}$
- Find a theoretically justified way of setting $\tau\downarrow y$
- If $\Gamma\downarrow\mathcal{K} = \{\text{known}\}$, then perform standard conformal prediction; else raise an alarm

Year 1 Questions

- How well does conformal prediction behave when $x\downarrow q \notin Y\downarrow train$?
- What is the best general-purpose anomaly detection algorithm?

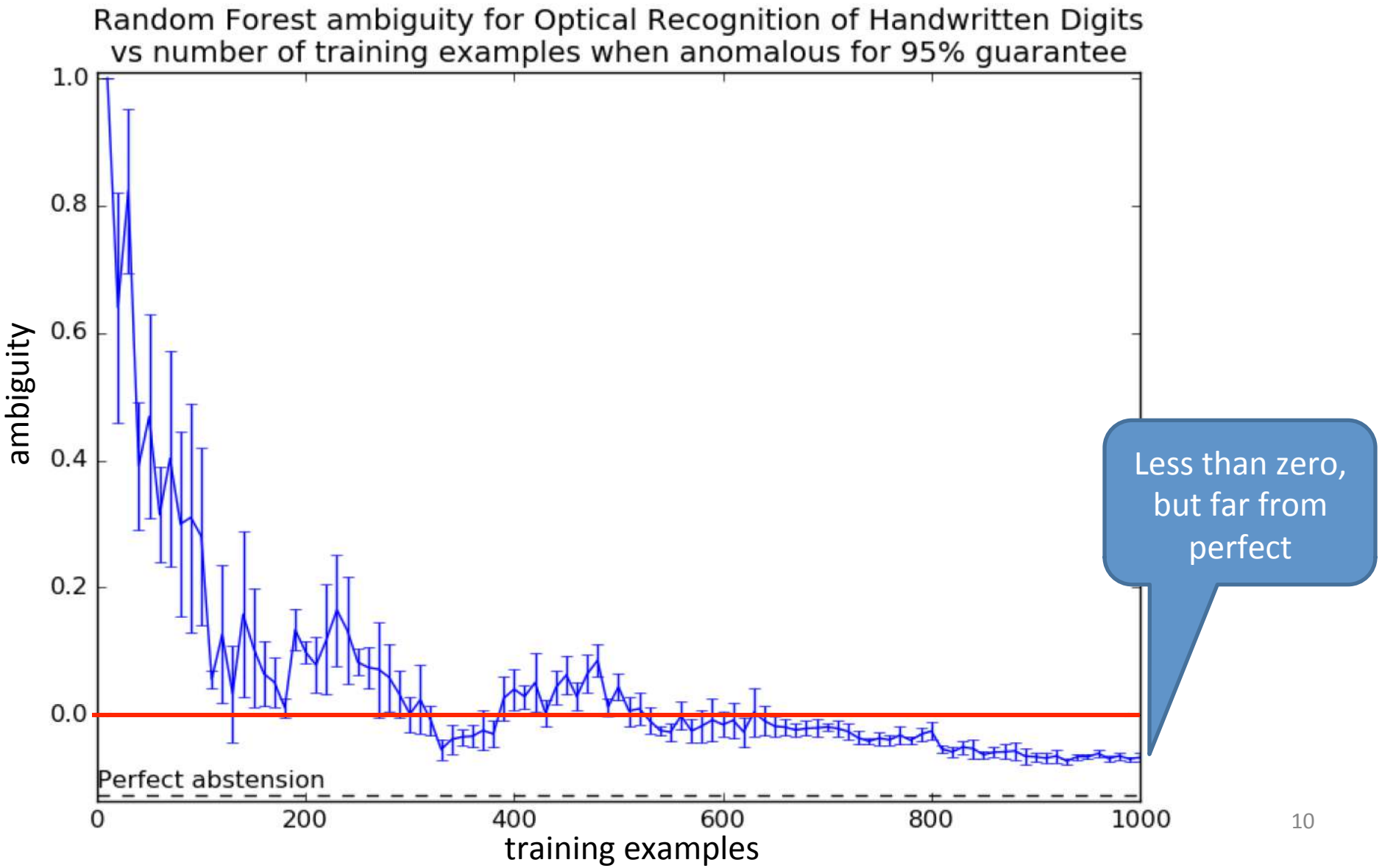
Conformal Prediction with Unknown Unknowns

- Ideal behavior: $\Gamma(x\downarrow q, \epsilon; R) = \emptyset$
- Metric for conformal prediction
  - Ambiguity: $|\Gamma(x\downarrow q, \epsilon; R)| - 1 / |Y\downarrow train| - 1$
    - $= 0$ for singleton predictions
    - $= -1 / |Y\downarrow train| - 1$ when $\Gamma = \emptyset$

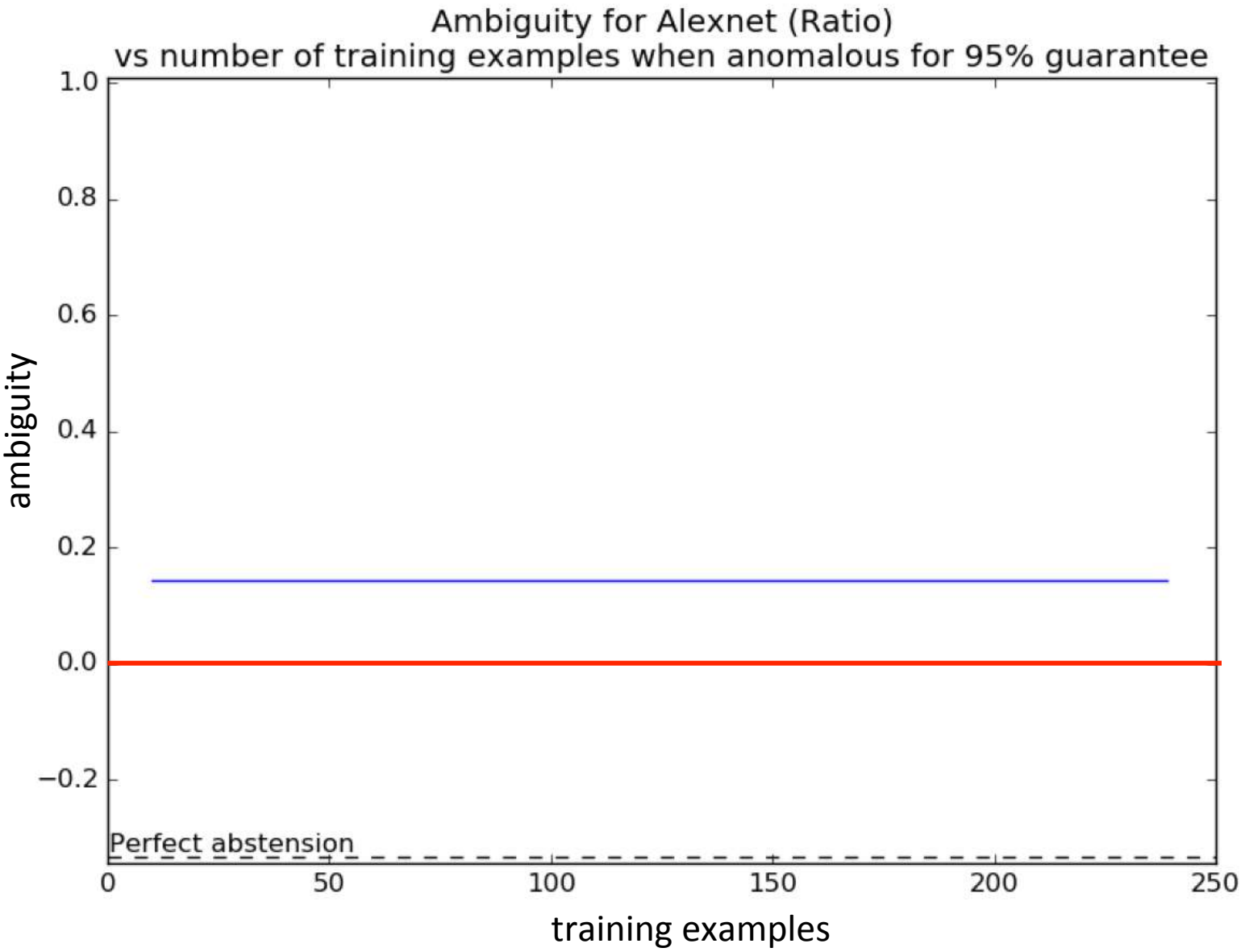
Experiments

- UCI multiclass data sets + random forest
  - novel classes are held out from training data
- AlexNet trained on {cat, dog, fictional animal, string bean}
  - novel class: mushroom

RF on Handwritten Digits



AlexNet on Mushrooms



Conclusion

- Standard conformal prediction is not able to reliably detect unknown unknowns

Anomaly Detection Algorithms Benchmarking Study

- Select UCI data sets
- Select subset of classes for "nominal" and rest for "anomalous"

Emmott, et al. arXiv 1503.01158

Steel Plates Faults
Gas Sensor Array Drift
Image Segmentation
Landsat Satellite
Letter Recognition
OptDigits
Page Blocks
Shuttle
Waveform
Yeast
Abalone
Communities and Crime
Concrete Compressive Strength
Wine
Year Prediction

Construct Data Sets to Systematically Vary Four Parameters

- **Point difficulty**: How deeply are the anomaly points buried in the nominals?
  - Fit supervised classifier (kernel logistic regression)
  - Point difficulty: $P(y = "nominal" | x)$ 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: $\log AUC/1-AUC$
- AP (Average Precision)
  - area under the precision-recall curve
  - average of the precision computed at each ranked anomaly point
  - transformed value: $\log AP/\mathbb{E}[AP] = \log LIFT$

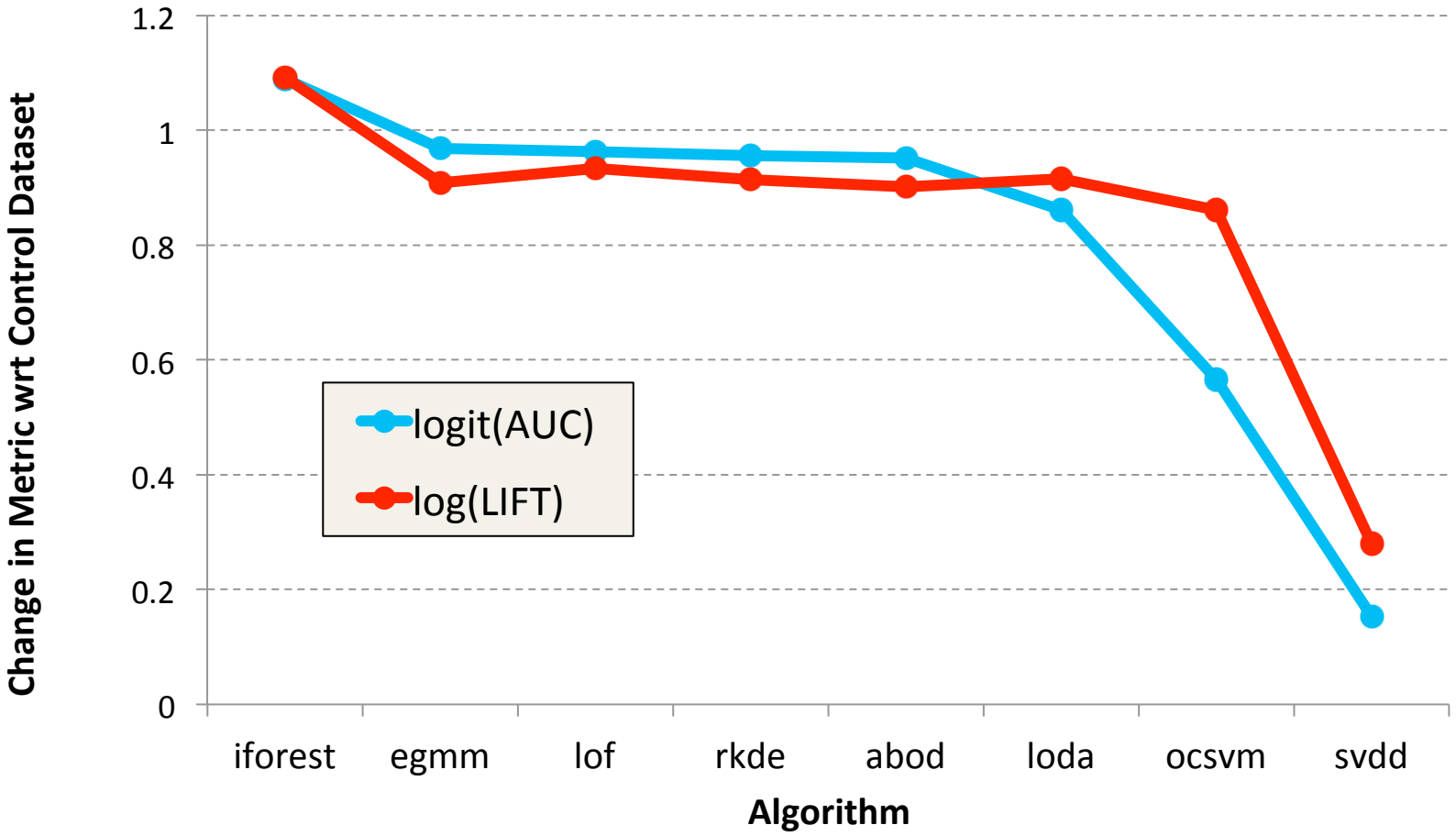
Anomaly Detection Algorithms

- **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
- Assess the *algo* effect while controlling for all other factors

Algorithm Comparison



Conclusion

- Isolation Forest is slightly better than the other algorithms
- LODA is very good and extremely efficient
- Quantile methods are surprisingly poor (esp. for AUC)

Next Steps

- Detecting novel classes via anomaly detection
- Application to imitation learning
  - detecting novel states
  - detecting surprising agent behavior

Collaboration Possibilities

- How should we respond to an alarm?
  - What "safety action" should we take?
  - How can we prove that the resulting combined system is safe?