Anomaly Detection in Machine Learning and Computer Vision

Thomas G. Dietterich, Distinguished Professor (Emeritus)
Oregon State University, Corvallis, OR USA 97331

Oregon State University

Anomaly Detection Use Cases

- **Data Cleaning**
  - Remove corrupted data from the training data
  - Example: Typos in feature values, feature values interchanged, test results from two patients combined
- **Fraud Detection, Cyber Attack, etc.**
  - At training or test time, illegal behavior creates anomalous data
- **Open Category Detection**
  - At test time, the classifier is given an instance of a novel category
  - Example: Self-driving car (trained in Europe) encounters a kangaroo (in Australia)
- **Novel Sub-category Detection**
  - At test time, the classifier is given a new kind of instance for a known category
  - Example: Chihuahua shown to a classifier trained only on Beagle and Golden Retriever
  - Example: New subtype of known disease
- **Out-of-Distribution Detection**
  - At test time, the classifier is given an instance collected in a different way
  - Example: Chest X-Ray classifier trained only on front views is shown a side view
  - Example: Self-driving car trained in clear conditions must operate during rainy conditions

Anomaly Detection

Definition of "anomaly":

- A data point that is generated by a different process than the process that is generating the "nominal" points
- Examples: sensor failures, fraud, cyber-attack, etc.

Challenges:

- Little or no labeled data
- Anomalies are rare
- Anomalies may not come from a well-defined probability distribution (especially in adversarial settings)
- Nuisance Novelty: Not all anomalies are relevant to the task or use-case
    - Irrelevant features in web site behavior or internet traffic
    - Changes in image background or context

Strategy:

- Because anomalies are rare, the main strategy for detecting them is to look for outliers: points that are far away from most of the data

Application Scenarios

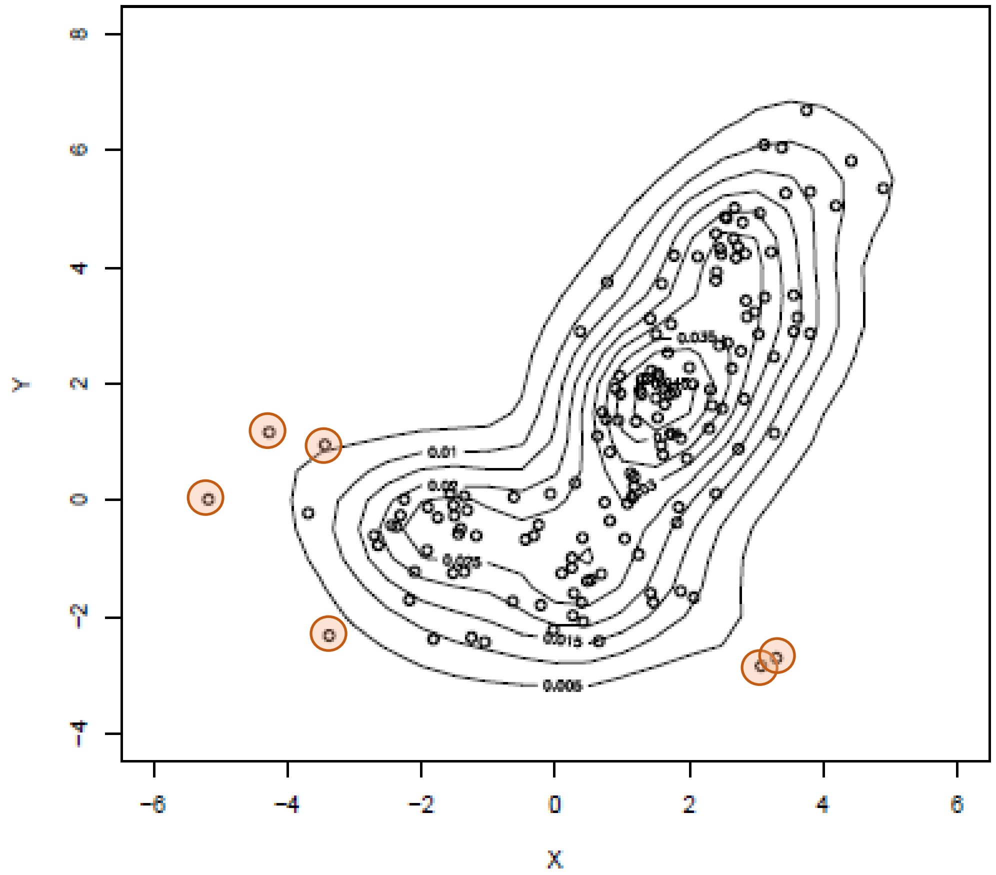
Training Data	Deployment Data	Example
Mix of nominal and anomaly	N/A	Data cleaning, fraud detection
Mix of nominal and anomaly	Mix of nominal and anomaly	Fraud detection, cyber attacks
Nominal-only (“clean”)	Mix of nominal and anomaly	Novel categories, novel cyber attacks, novel diseases

Part 1: Anomaly Detection for Feature Vector Data

- Traditional machine learning representation
- Advantages:
  - Meaningful features/attributes
  - Can design an appropriate distance or similarity measure

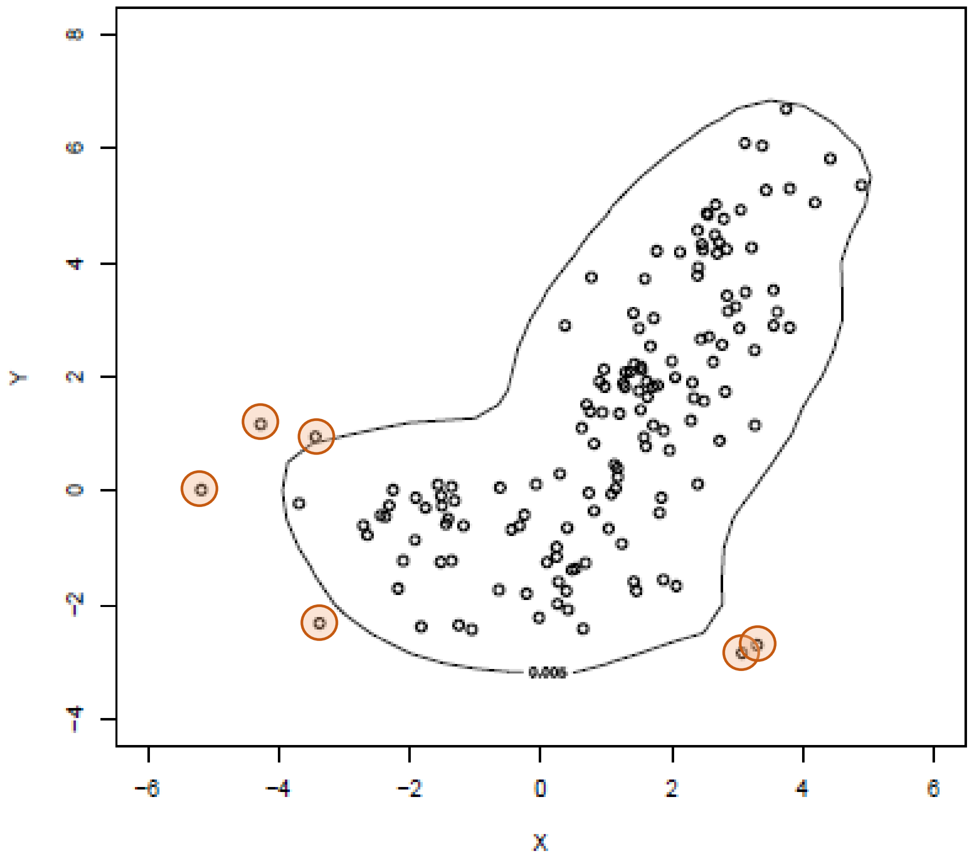
Technical Approaches

- **Density Estimation Methods**
  - Model the joint distribution $P_D(x)$ of the input data points
- **Quantile Methods**
  - Model the region of data space where $P_D(x) \geq \tau$
- **Distance-Based Methods**
  - Compute distance of new point to its k nearest neighbors
- **Projection Methods**
  - Project your data into a lower-dimensional space and then apply any of the above methods



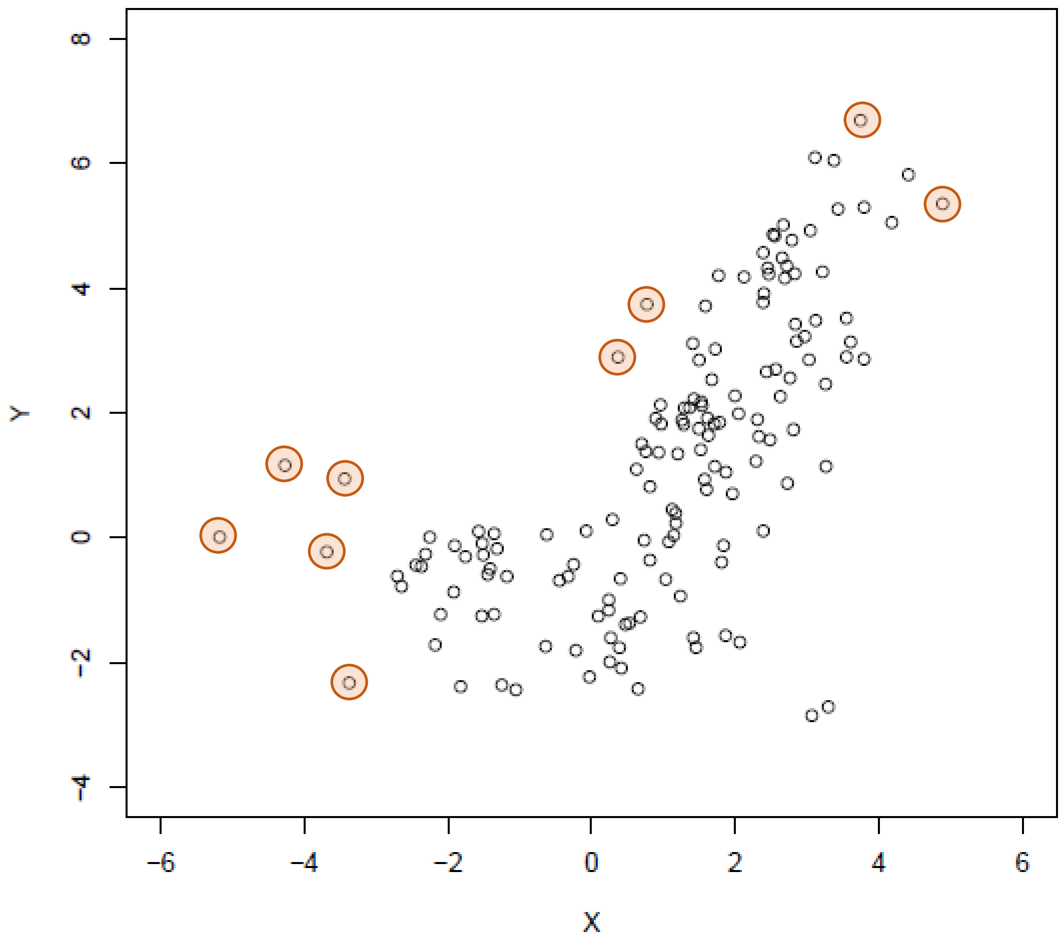
Technical Approaches

- **Density Estimation Methods**
  - Model the joint distribution $P_D(x)$ of the input data points
- **Quantile Methods**
  - Model the region of data space where $P_D(x) \geq \tau$
- **Distance-Based Methods**
  - Compute distance of new point to its k nearest neighbors
- **Projection Methods**
  - Project your data into a lower-dimensional space and then apply any of the above methods



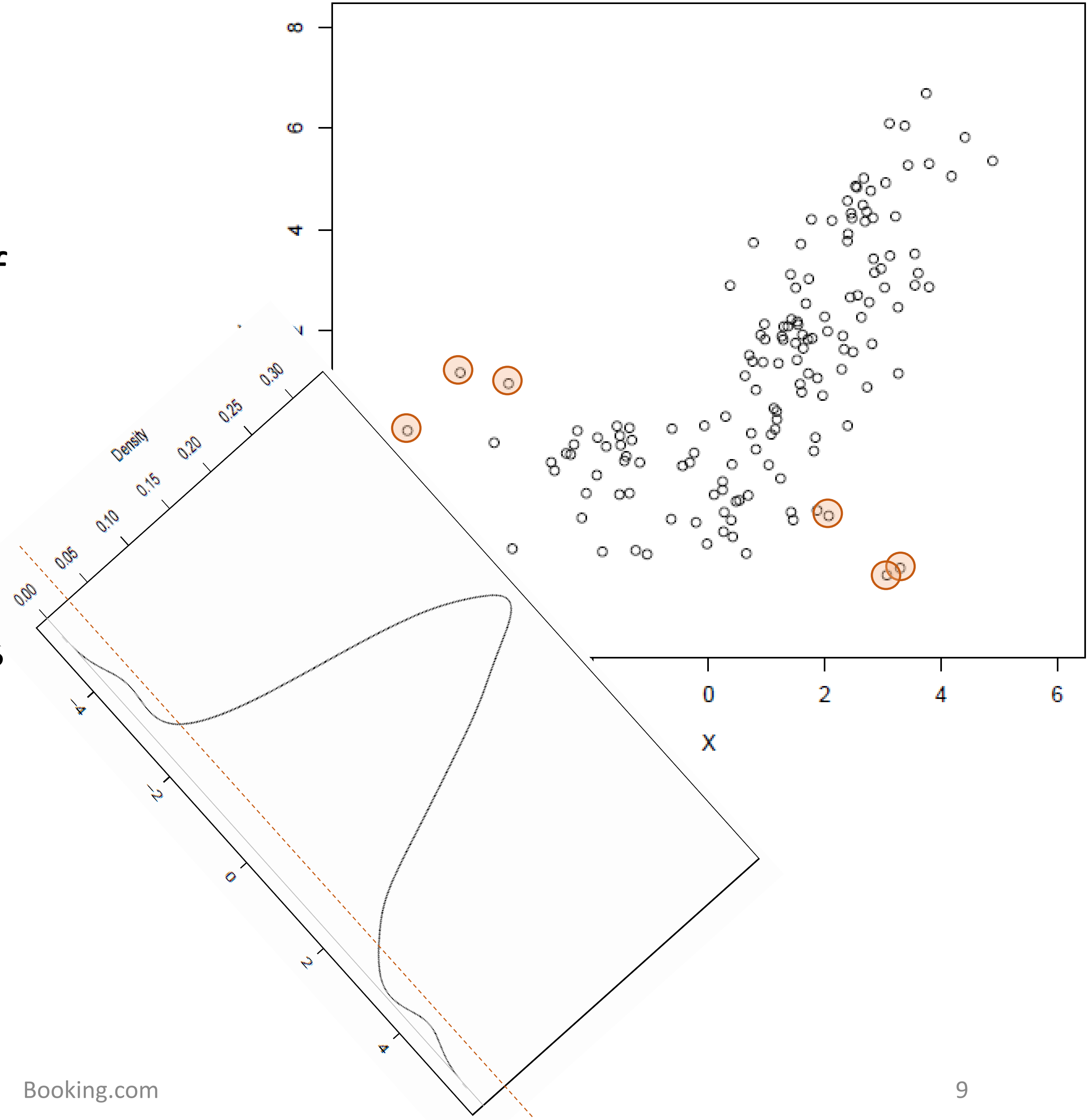
Technical Approaches

- **Density Estimation Methods**
  - Model the joint distribution $P_D(x)$ of the input data points
- **Quantile Methods**
  - Model the region of data space where $P_D(x) \geq \tau$
- **Distance-Based Methods**
  - Compute distance of new point to its k nearest neighbors
- **Projection Methods**
  - Project your data into a lower-dimensional space and then apply any of the above methods



Technical Approaches

- **Density Estimation Methods**
  - Model the joint distribution $P_D(x)$ of the input data points
- **Quantile Methods**
  - Model the region of data space where $P_D(x) \geq \tau$
- **Distance-Based Methods**
  - Compute distance of new point to its k nearest neighbors
- **Projection Methods**
  - Project your data into a lower-dimensional space and then apply any of the above methods



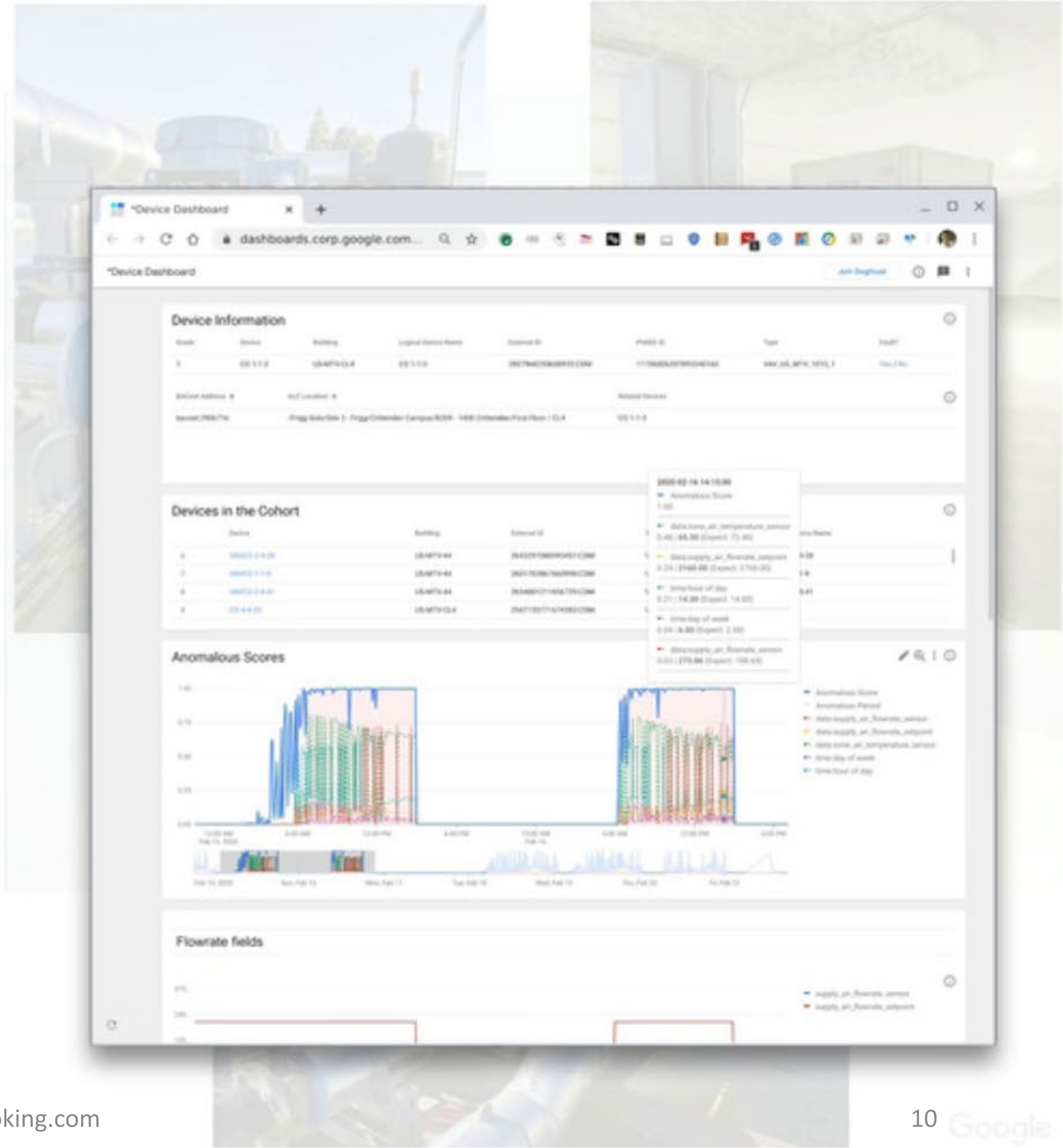
Case Study: Smart Buildings

John Sipple (ICML 2020)

Objective: *Make buildings smarter, secure and reduce energy use! Improve occupant comfort and productivity while also improving facilities' operation efficiencies.*

120 million measurements daily, generated by over **15,000 climate control devices, in 145 Google buildings**

Since going live in June 2019, FDD has created **458 facilities technician work orders, with a 44% True Positive rate**



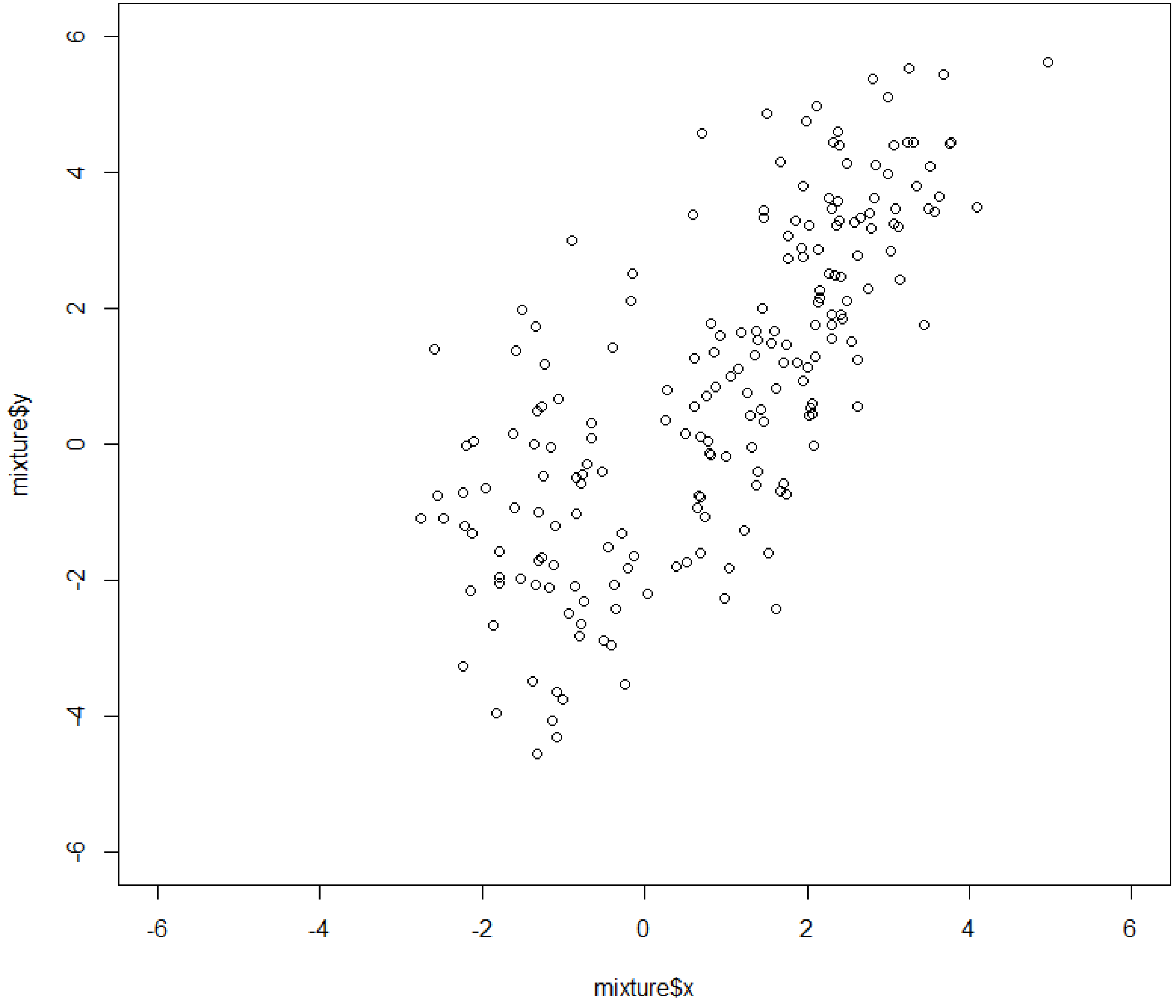
Method: Density Estimation via Noise-Contrastive Estimation

- Idea:
  - Label all points in our data set D to belong to class 0
  - Uniformly sample points from a “box” that contains D and label those points as class 1
  - Fit a flexible machine learning model $\hat{f}$ to the data
  - $\hat{f}(x) = P(y = 1|x)$ which is the probability that x is an anomaly

- History
  - “Well known statistical folklore” according to Hastie, Tibshirani & Friedman (2016) *Elements of Statistical Learning* 2nd edition
  - Pihlaja, Guttman & Hyvarinen (2010) “A Family of Computationally Efficient and Simple Estimators for Unnormalized Statistical Models”. UAI 2010

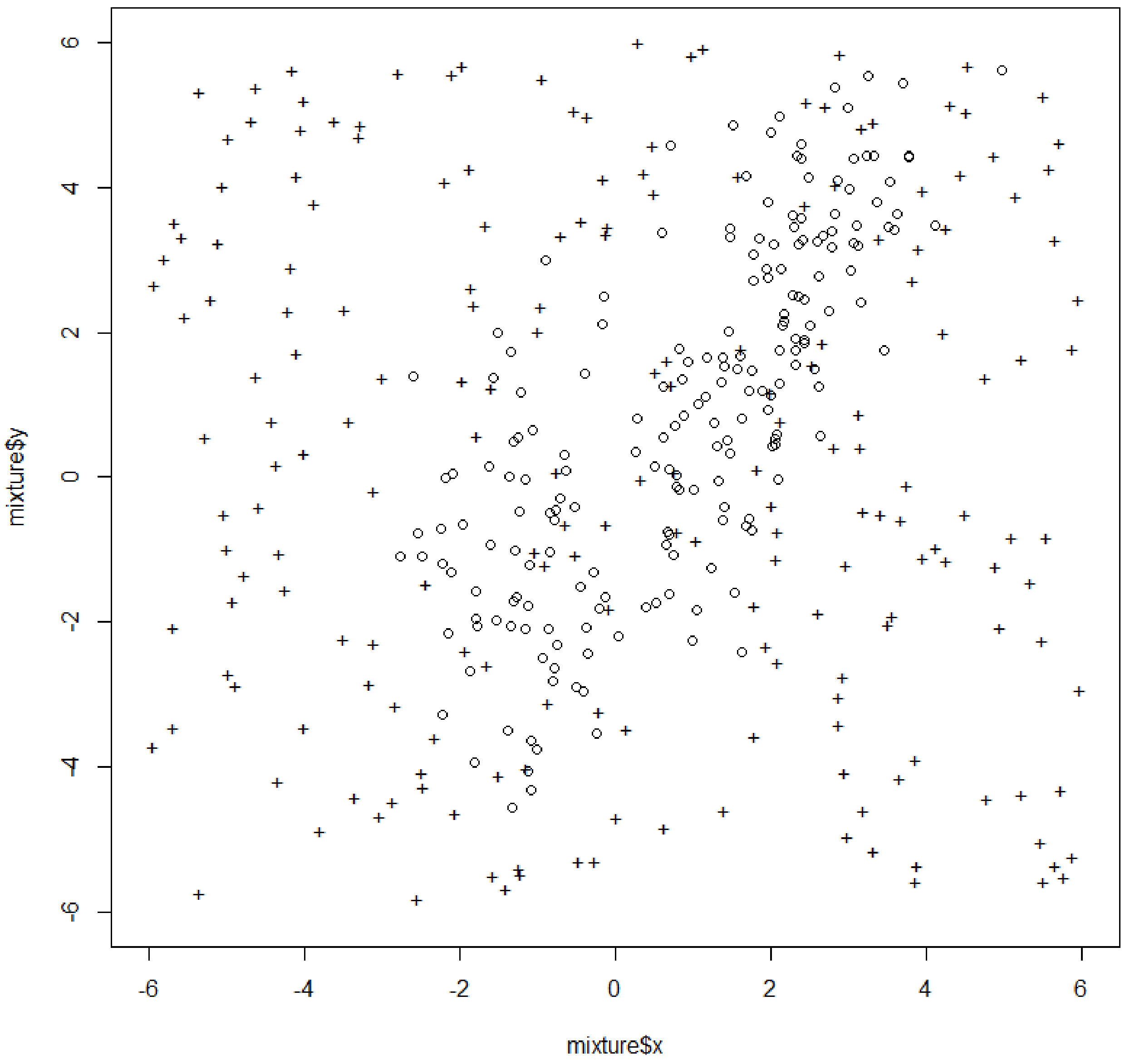
Example

- Training data D



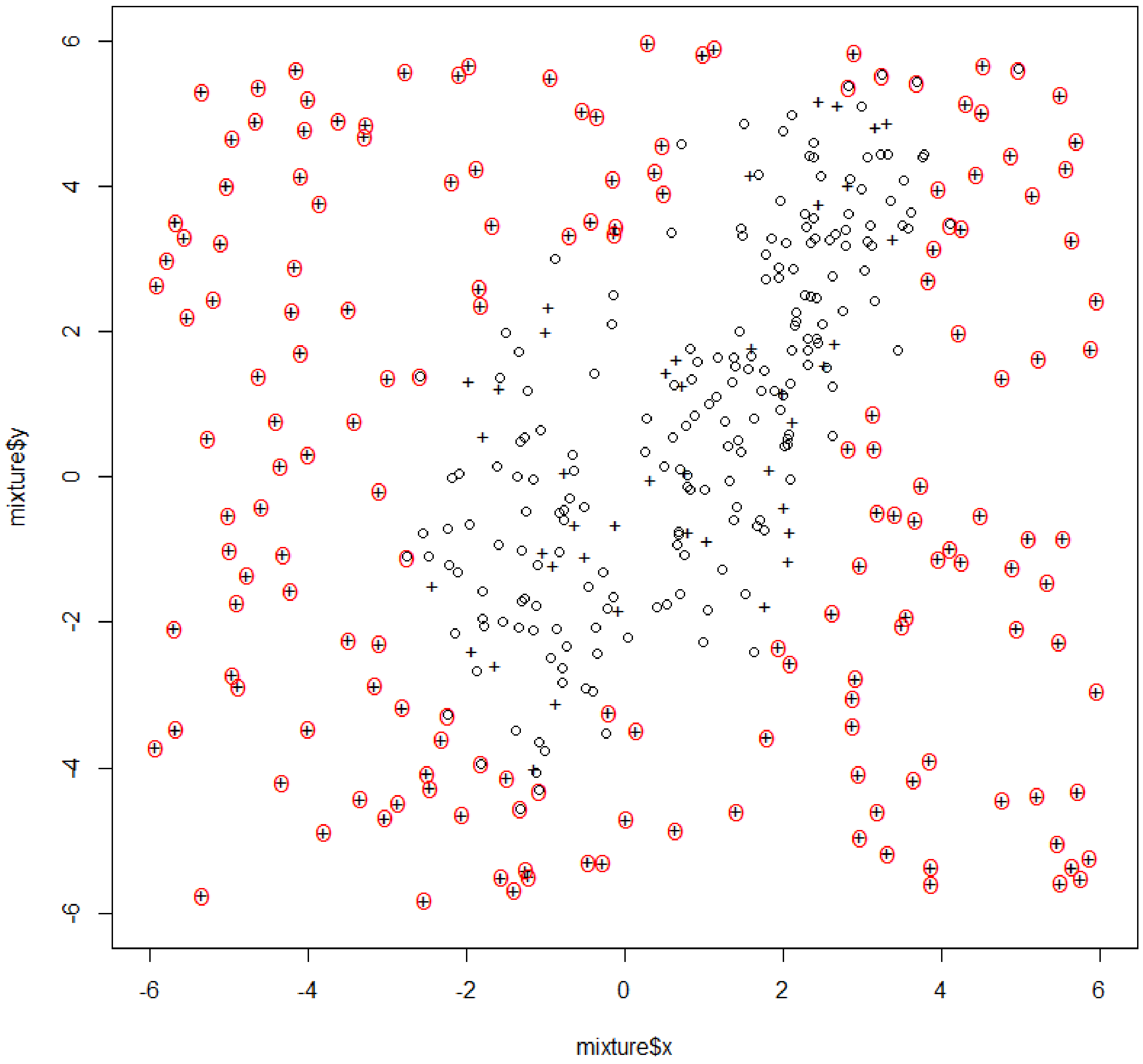
Example

- Training data D
- Random sample N



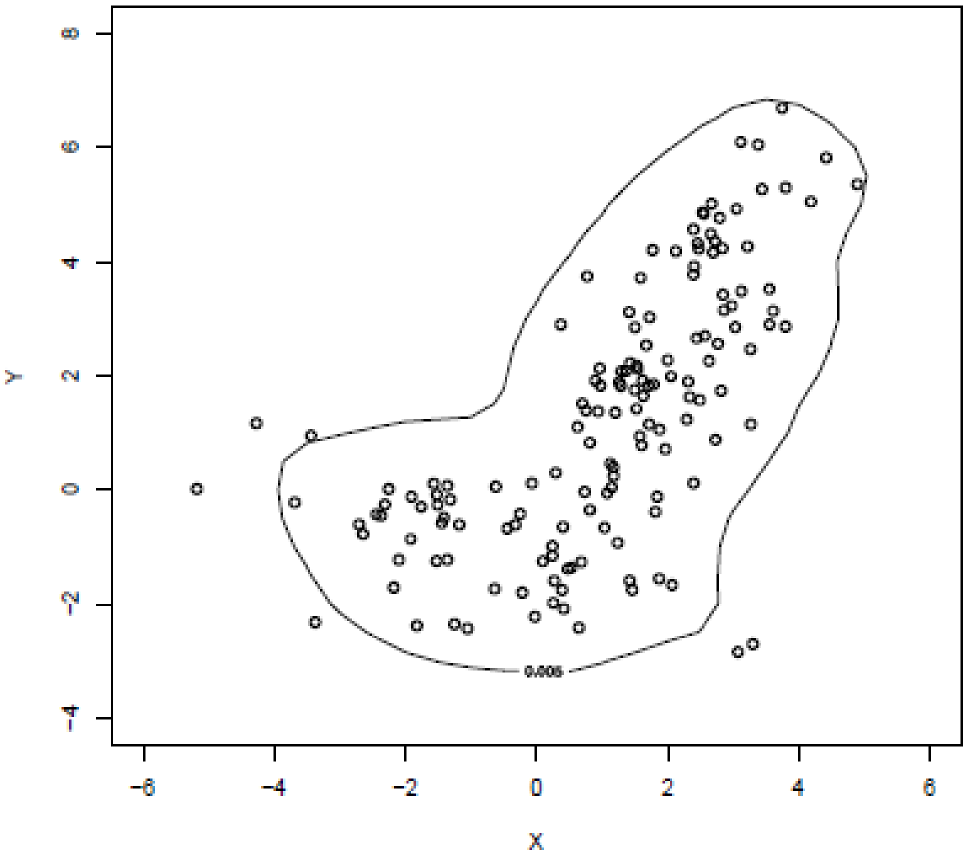
Example

- Training data D
- Random sample N
- Fit a function $\hat{f}$. I used R "gbm" method with the "logit" link function
- Points x where $\hat{f}(x) > 0.5$



Approach 2: Quantile Methods

- We don’t really need to model the whole probability distribution
- One-class Support Vector Machine (OCSVM)
- Support Vector Data Description (SVDD)



Approach 3: Distance-Based Methods

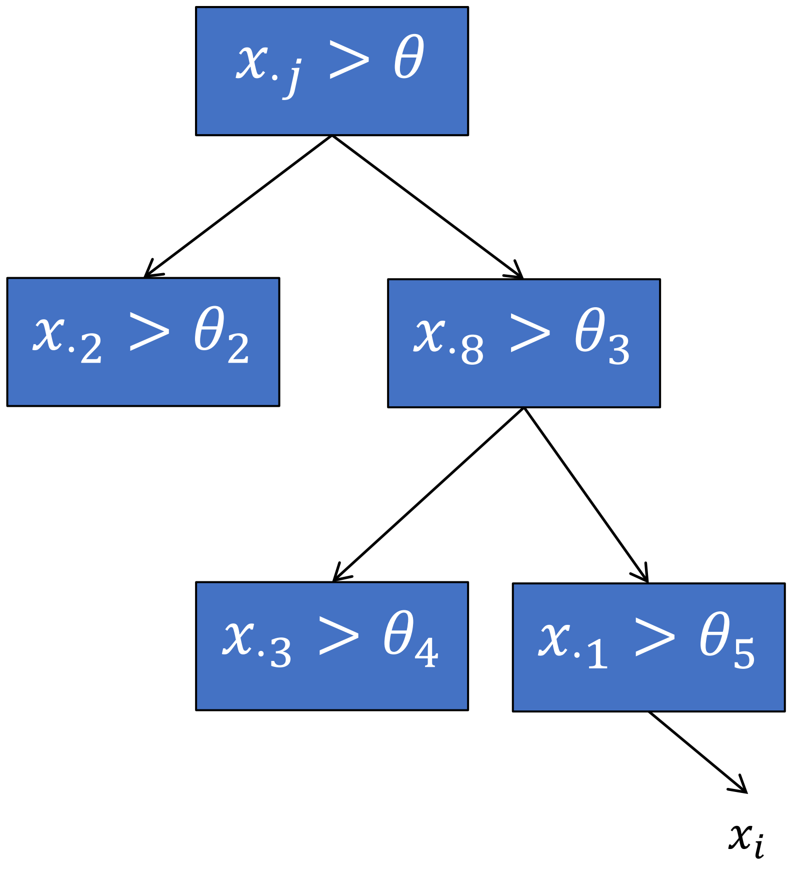
- k-nearest neighbor
- LOF: Local Outlier Factor
- ABOD: Angle-based Outlier Detector

Approach 4: Projection Methods

- Isolation Forest [Liu, Ting, Zhou, 2011]
- LODA [Pevny, 2016]

Isolation Forest [Liu, Ting, Zhou, 2011]

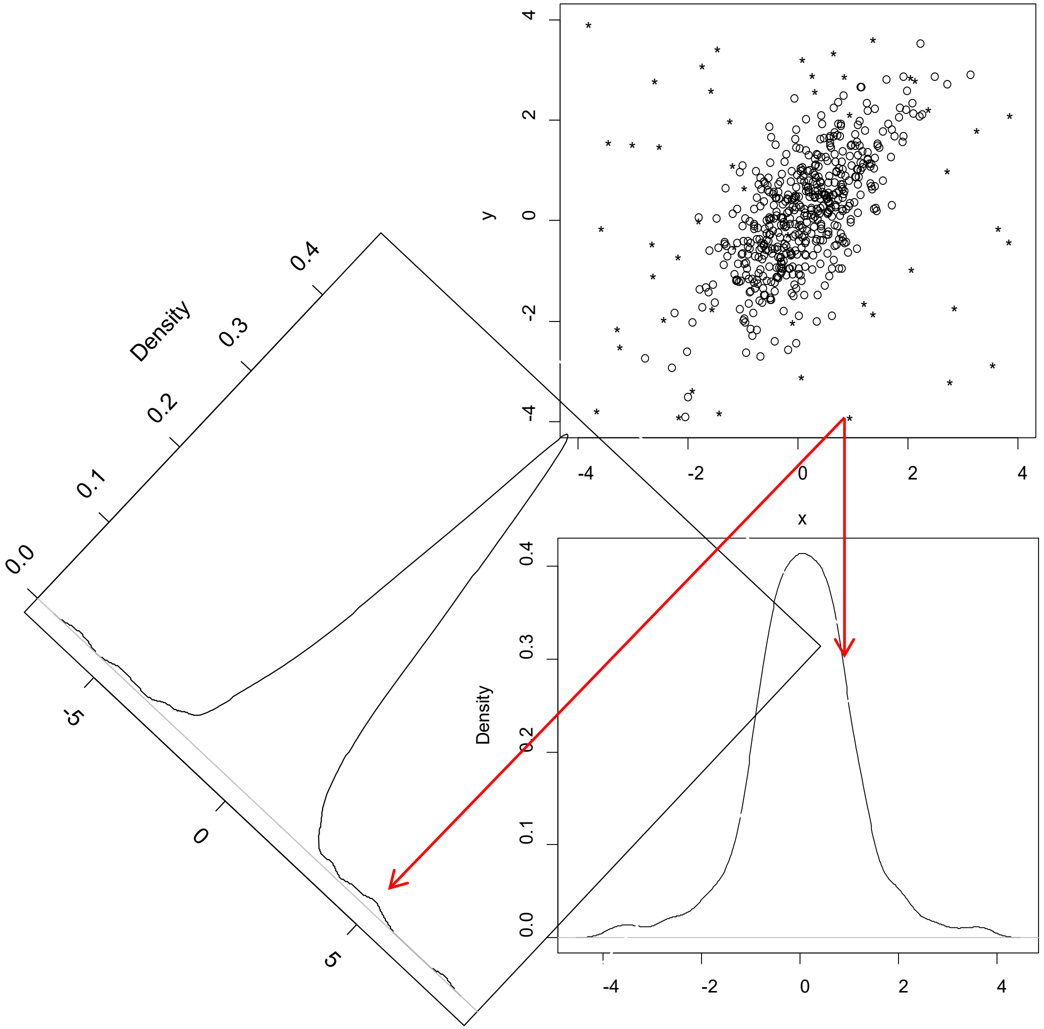
- Construct a fully random binary tree
  - choose attribute j at random
  - choose splitting threshold θ uniformly from $\left[\min\left(x_{\cdot j}\right), \max\left(x_{\cdot j}\right)\right]$
  - until every data point is in its own leaf
  - let $d(x_i)$ be the depth of point x_i
- repeat L times
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

- Generate L sparse random projections (projections onto L lines in d-dimensional space)
- Estimate the probability density for each project (easy)
- Anomaly score is the average of the anomaly scores in each projection



Benchmarking Study
[Andrew Emmott]

- Most AD papers only evaluate on a few datasets
- Often proprietary or very easy (e.g., KDD Cup 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]
[Emmott, MS Thesis. 2020]

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: 11,888 Non-trivial Benchmark Datasets

Nine Algorithms

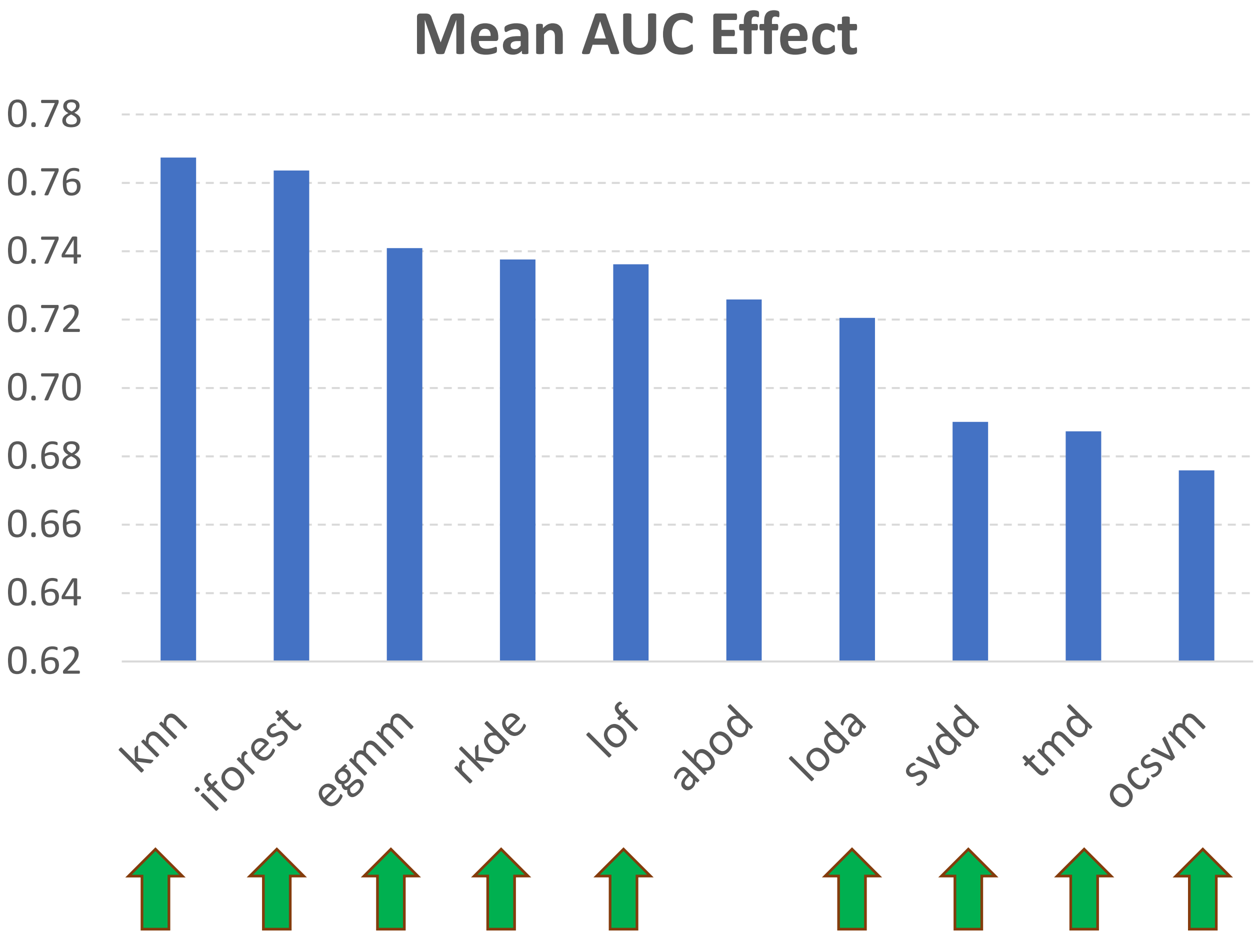
- Density-Based Approaches
  - RKDE: Robust Kernel Density Estimation (Kim & Scott, 2008)
  - EGMM: Ensemble Gaussian Mixture Model (our group)
- Quantile-Based Methods
  - OCSVM: One-class SVM (Schoelkopf, et al., 1999)
  - SVDD: Support Vector Data Description (Tax & Duin, 2004)
- Neighbor-Based Methods
  - k-NN: Mean distance to k-nearest neighbors
  - 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 of Variance

- 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
- $metric$: area under the ROC curve for the nominal vs. anomaly binary decision

Benchmarking Study Results

- 19 UCI Datasets
- 8 Leading "feature-based" algorithms
- 11,888 non-trivial benchmark datasets
- Mean AUC effect for "nominal" vs. "anomaly" decisions
  - Controlling for
    - Parent data set
    - Difficulty of individual queries
    - Fraction of anomalies
    - Irrelevant features
    - Clusteredness of anomalies
- Baseline method: Distance to nominal mean ("tmd")
- Best methods: K-nearest neighbors and Isolation Forest (projection method)
- Worst methods: Kernel-based OCSVM and SVDD



Employs a distance

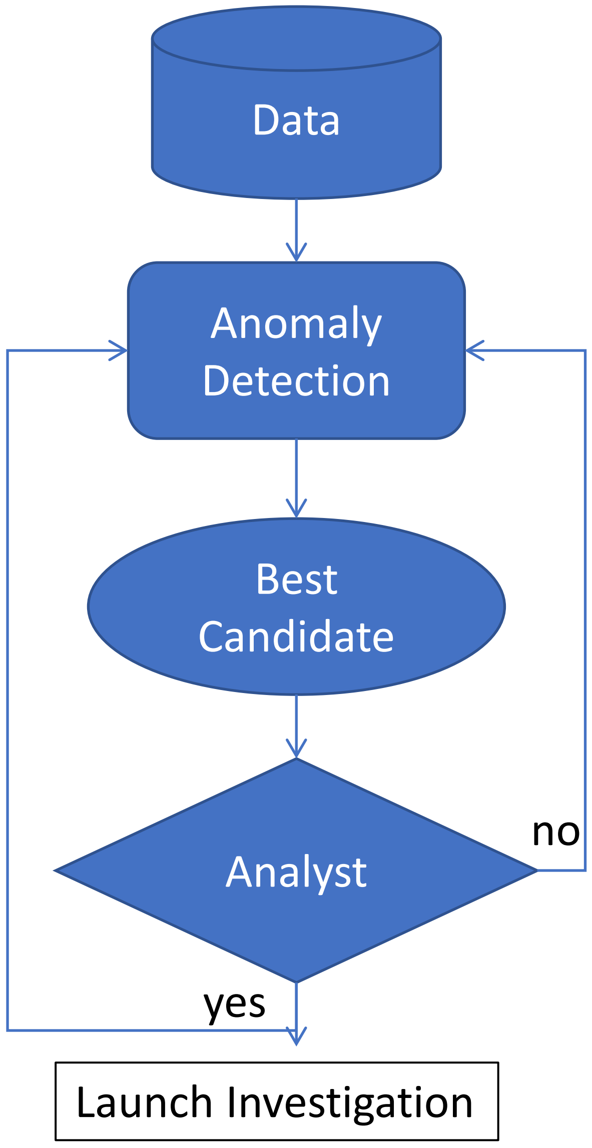
Application to Cyber Security and Fraud Detection

- In most applications, anomaly detection has a significant false alarm rate
- This means that a human needs to examine each anomaly alarm and decide whether it is a true alarm or a false alarm
- This provides us with a source of feedback for reducing false alarms

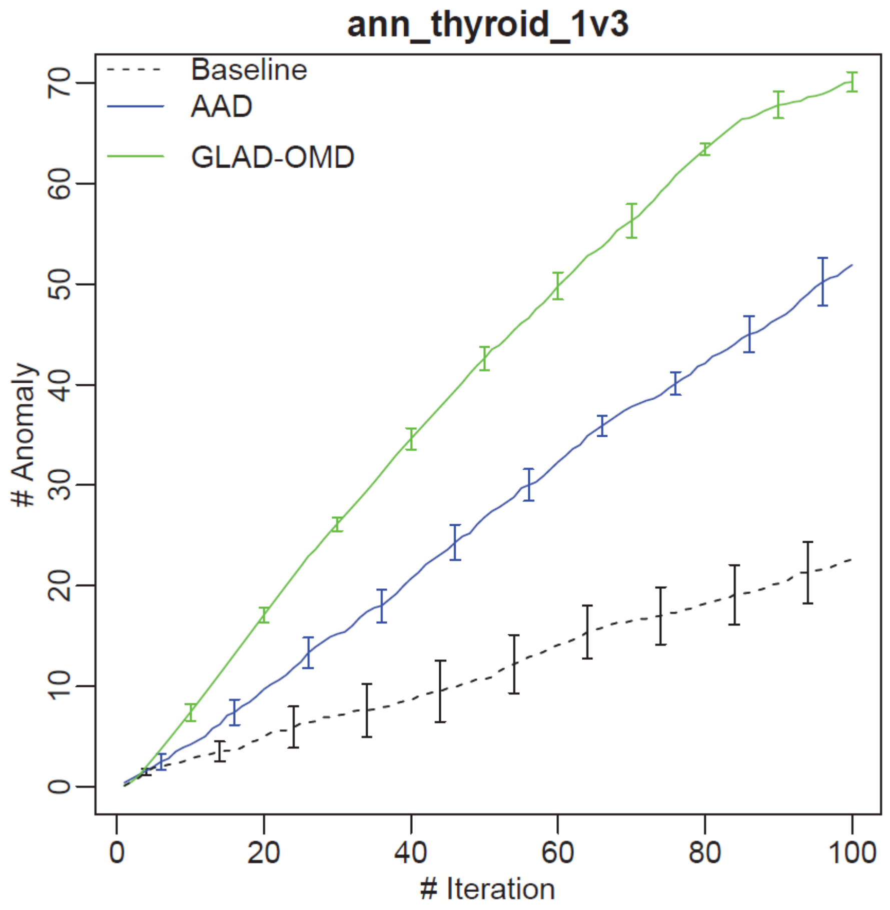
Incorporating Analyst Feedback into Anomaly Detection

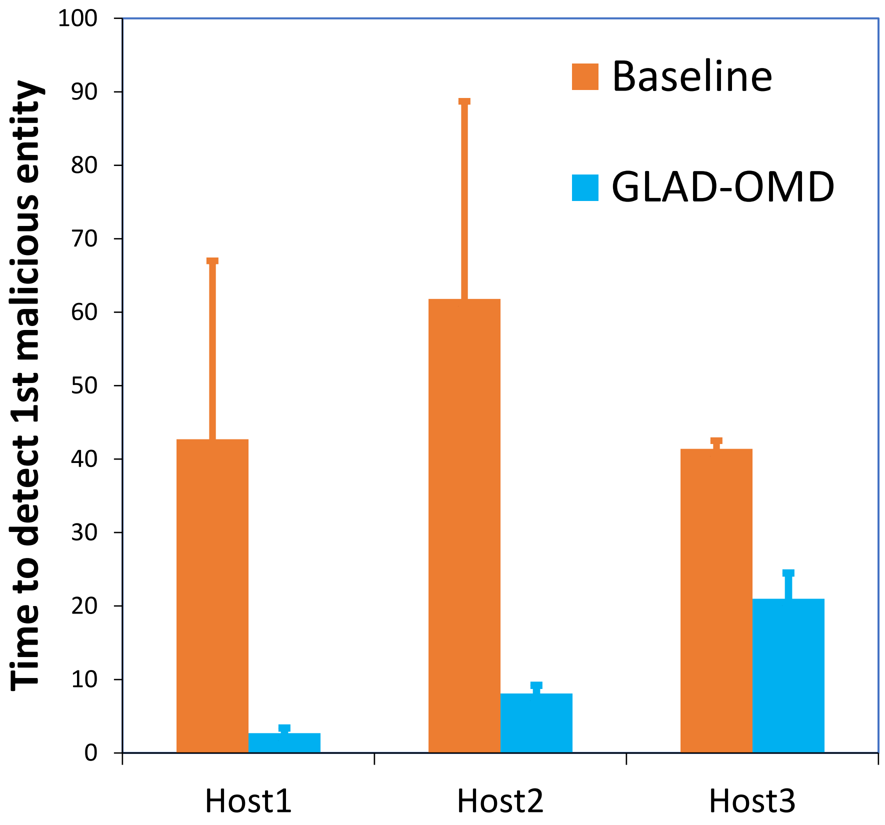
- Show top-ranked (unlabeled) candidate to the Analyst
- Analyst labels candidate
- Label is used to update the anomaly detector

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



Analyst Feedback Yields Huge Improvements in Anomaly Discovery





Part 2: Anomaly Detection in Computer Vision

- **Challenges:**
- No easy distance metrics
- Very high dimension
- High degree of nuisance novelty in natural images

- **State-of-the-art methods have difficulty deciding that SVHN house numbers are anomalies compared to CelebA!**

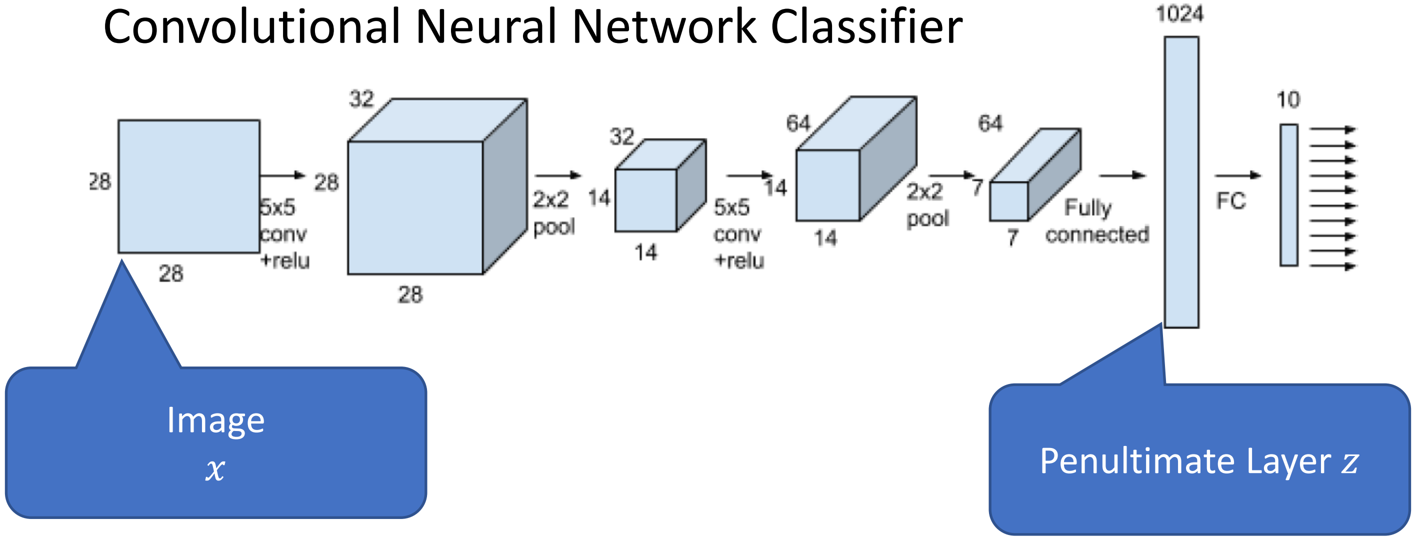
Faces from CelebA

House Number from SVHN

Central Challenge: Learned Representations

- Train a deep network to perform a classification task
- Learned representation $z = E(x)$
  - E is called the “Encoder”
- This representation is trained to separate the classes
- It loses information needed to detect outliers
- Outliers x' are often mapped close to the known classes
  - $E(x) \approx E(x')$
- No method can detect the outlier if it is not an outlier in the z space
- We have little or no control over the topology of z space (e.g., is Euclidean distance valid?)

Convolutional Neural Network Classifier



Three Main Approaches

- Method 0: Train CNN classifier
  - Extract anomaly signal from the z space
- Method 1: Modify training so that z can represent “open space”
  - Introduce “simulated anomalies” and hope the network generalizes
- Method 2: Anomaly Detection via Failure:
  - Train the network on a task so that the network will fail when given outliers

- There are many many other ad hoc methods, but these are the only approaches that have a principled justification

Method 0: Research Questions

- Q1: How well do existing anomaly scoring methods extract the anomaly information that is captured in the latent representation z?
  - Approach: Compare to an oracle anomaly detector
- Q2: How well could *any* network with this architecture perform the anomaly detection task
  - Approach: Supervised training on both nominal and anomalous classes

- Definition of anomalies: Classes not seen during training
  - “Open Category” or “Open Set” problem
  - We claim this is harder and more realistic than classic Out-Of-Distribution tasks

Methods:

- CIFAR-10: 6 “nominal” classes and 4 “anomaly” classes
- CIFAR-100: 80 “nominal” classes and 20 “anomaly” classes

- Train Classifier
  - Divide data into train (60%), validate (20%), test (20%)
  - Remove anomaly classes from the training and validation data
  - Train ResNet34; use validation set accuracy to determine stopping point
  - Compute test set anomaly scores using various metrics; measure AUC
- Oracle Anomaly Detection
  - Take all validation data and label the nominal classes as “nominal” and the anomaly classes as “anomaly”
  - Train a binary classifier that takes z as input and predicts “nominal” vs. “anomaly”
  - Compute test set anomaly scores using this classifier; measure AUC
- Oracle Representation
  - Train a binary classifier on “nominal” vs “anomaly” using data from all classes
  - Measure “nominal” vs “anomaly” AUC on the test data; measure AUC

Results

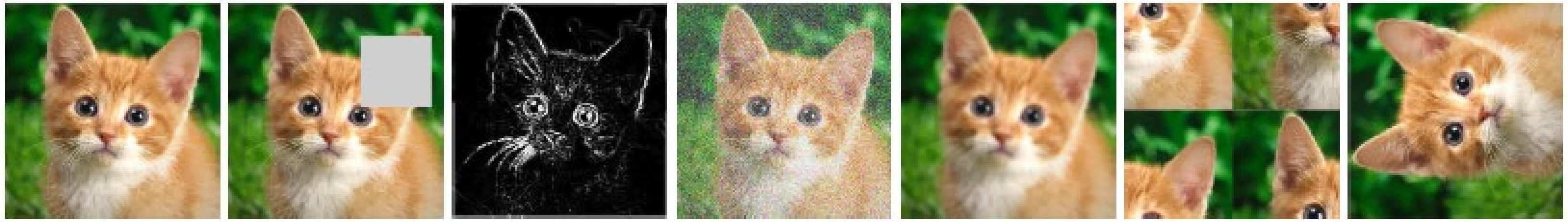
AUC	CIFAR 10	CIFAR 100
AD	0.776 ± 0.008	0.717 ± 0.008
Oracle AD	0.905 ± 0.015	0.789 ± 0.011
Oracle Classifier	0.987 ± 0.003	0.809 ± 0.011

Conclusions

- Q1: The latent space contains much more anomaly information than is extracted by current anomaly scores
  - 0.776→0.905 = 0.129; 0.717→0.789 = 0.072
- Q2: There is additional anomaly information in the images that is not represented by the latent space
  - 0.905→0.987 = 0.082; 0.789 →0.809 = 0.020

Method 1: Training to Open Up Space

- Define image of transformations
  - Transformations that preserve the class label T
    - horizontal flip, Sobel, Noise, Blur, change color map, zoom in or out
  - Transformations that change the class label S
    - permute, rotate



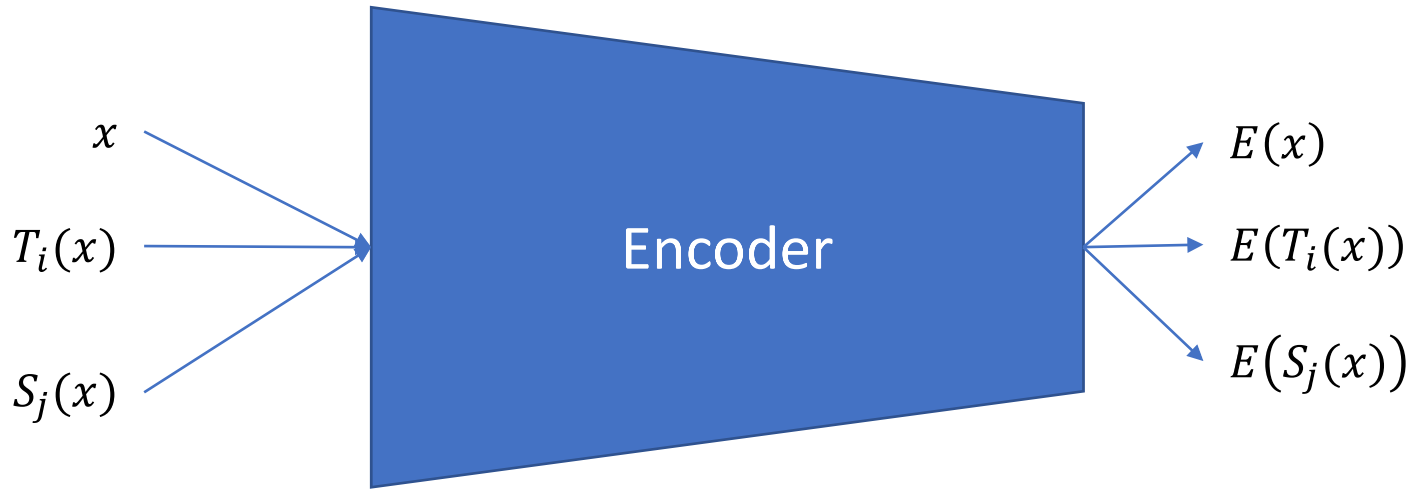
(a) Original (b) Cutout (c) Sobel (d) Noise (e) Blur (f) Perm (g) Rotate

T_1 T_2 T_3 T_4 S_1 S_2

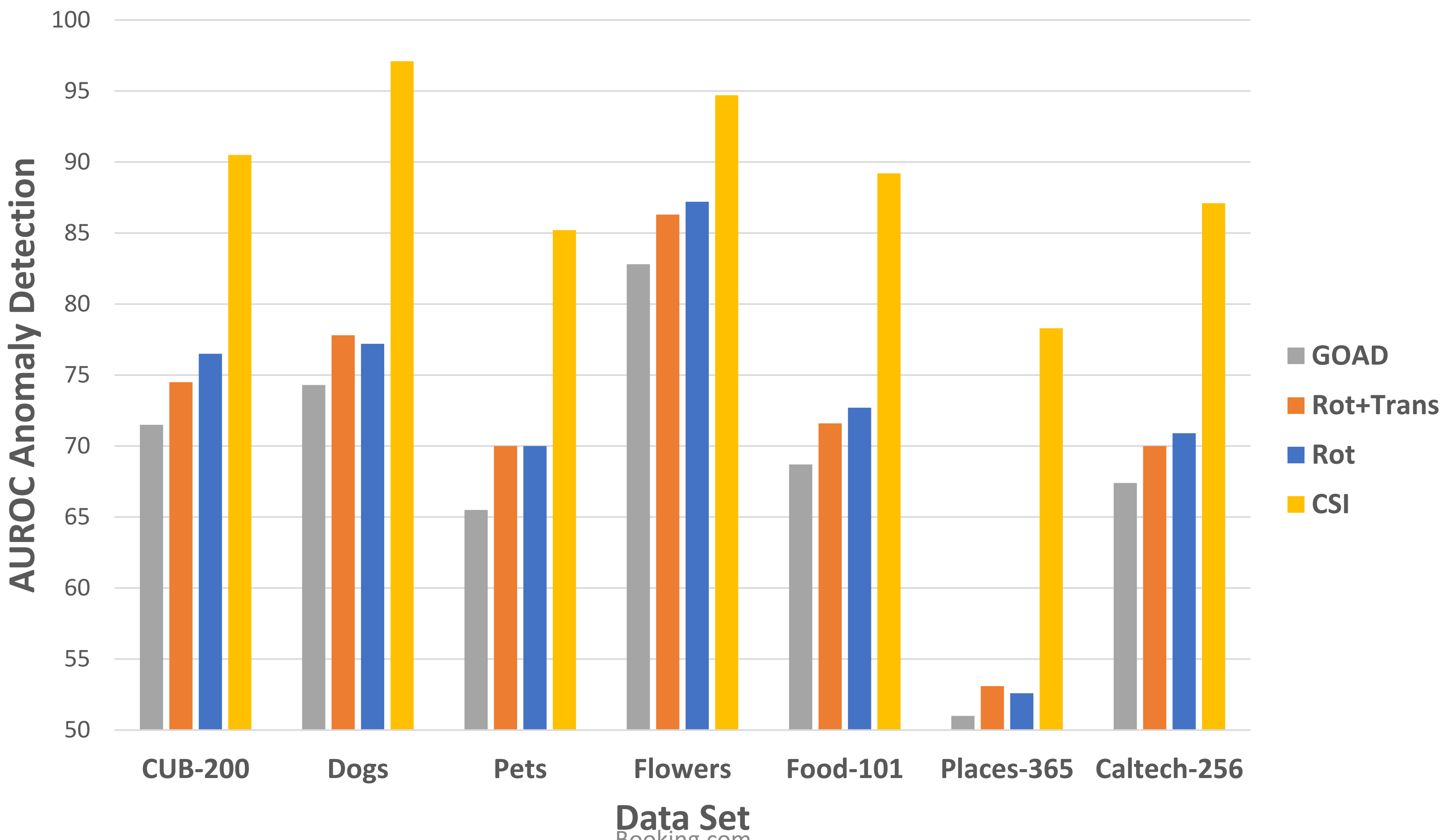
Tack, Mo, Jeong, Shin: CSI. NeurIPS 2020

Instance-Level Contrastive Learning

- Choose $T_i(x)$ and $S_j(x)$ at random from T and S
- Send x, $T_i(x)$ and $S_j(x)$ through the network to compute their encoded representations
- Update the network weights to make $E(x) \approx E(T_i(x))$ and make $E(x) \neq E\left(S_j(x)\right)$
- The S transformations simulate outliers and force the network to represent them as points far away from the inliers



Experiment: Train on ImageNet-30 (unlabeled)
Predict on mix of ImageNet-30 and an “anomaly” dataset



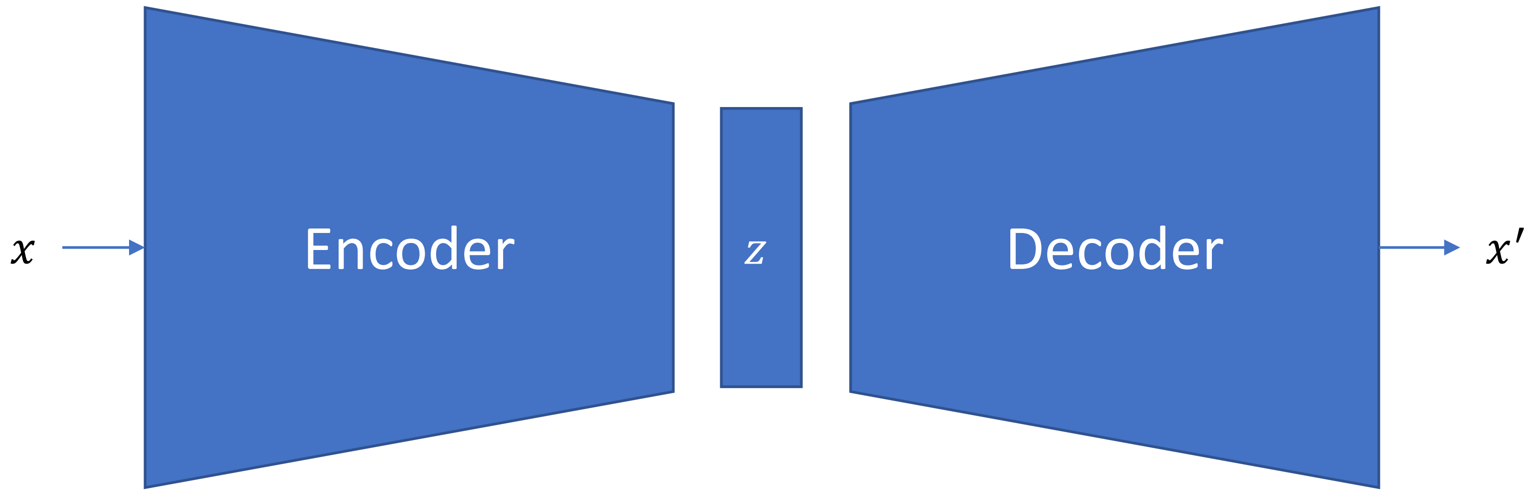
Tack, Mo, Jeong, Shin: CSI. NeurIPS 2020

Remarks

- No personal experience with this yet
- No theoretical guarantee that this will work

Method 2: Anomaly Detection via Failure

- Train network on a reconstruction task
- z is a “bottleneck” that requires the network to learn a compact code
- Train network to make $x \approx x'$ for nominals
- Hope that the reconstruction fails on anomalies
  - Make the bottleneck as small as possible
  - Other tricks (regularization for sparsity, etc.)



Results?

- For low-dimensional problems, we can replace the network with Principal Component Analysis (PCA) and measure the reconstruction error
  - This has worked well in many applications (e.g., Wagstaff, et al. 2013)
- However, experiments with deep networks have failed to achieve strong results
- Issues:
  - Hard to define how to measure similarity: $x \approx x'$
  - Networks can learn very general image compression schemes ➔ they don't fail on anomalies!

Summary

- General Anomaly Detection Methods
  - Density Estimation
  - Quantile Methods (OCSVM, SVDD)
  - Distance-Based Methods (KNN)
  - Projection Methods (Isolation Forest)
- Application to Cybersecurity and Fraud Detection
- Anomaly Detection in Computer Vision
  - Challenge: learned representations are task-specific
  - Standard CNNs retain a surprising amount of anomaly information
  - Open up “empty space” with simulated outliers
  - Solve reconstruction tasks

Concluding Remarks

- **Anomaly detection is important**
  - Critical for robust AI systems
  - Practical applications
- **Anomaly detection is difficult**
  - Moderately mature for tabular data sets
  - Fundamentally relies on some notion of distance
  - Very challenging for images where we need a notion of semantic distance
- **Research in this area is advancing rapidly with little theoretical understanding**

References

- Blázquez-García, A., Conde, A., Mori, U., & Lozano, J. A. (2020). A review on outlier/anomaly detection in time series data. *ArXiv, 2002.04236*(v1).
- Bergman & Hoshen (2020). Classification-Based Anomaly Detection for General Data. ICLR 2020
- Breunig, M. M., Kriegel, H., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. *ACM SIGMOD 2000 International Conference on Management of Data*, 1–12.
- Das, S., Wong, W. K., Dietterich, T., Fern, A., & Emmott, A. (2017). Incorporating expert feedback into active anomaly discovery. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 853–858. https://doi.org/10.1109/ICDM.2016.164
- Emmott, A., Das, S., Dietterich, T., Fern, A., & Wong, W.-K. (2016). A Meta-Analysis of the Anomaly Detection Problem. *ArXiv, 1503.01158*(v2), 1–35.
- Golan & El-Yaniv (2018). Deep Anomaly Detection Using Geometric Transformations. ArXiv: 1805.10917
- Garrepalli, R. (2020). Oracle Analysis of Representations for Deep Open Category Detection. Unpublished MS Thesis. Oregon State University.
- Hastie, Tibshirani & Friedman (2016) *Elements of Statistical Learning* 2nd edition
- Hendrycks, D., & Gimpel, K. (2017). A Baseline for Detecting Misclassified and Out-Of-Distribution Examples in Neural Networks. *ICLR 2017*.
- Kim, J., & Scott, C. D. (2012). Robust Kernel Density Estimation. *Journal of Machine Learning Research*, *13*, 2529–2565.
- Kriegel, H.-P., Schubert, M., & Zimek, A. (2008). Angle-based outlier detection in high-dimensional data. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 444–452. https://doi.org/10.1145/1401890.1401946

References (2)

- Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2012). Isolation-Based Anomaly Detection. *ACM Transactions on Knowledge Discovery from Data*, *6*(1), 1–39.
- Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., Lakshminarayanan, B., To, N., Deep, N., & Inference, A. B. (2019). Do Deep Generative Models Know What They Don't Know?
- Pevný, T. (2015). Loda: Lightweight on-line detector of anomalies. *Machine Learning*. https://doi.org/10.1007/s10994-015-5521-0
- Pihlaja, Guttman & Hyvarinen (2010) "A Family of Computationally Efficient and Simple Estimators for Unnormalized Statistical Models". UAI 2010
- Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., & Platt, J. (2000). Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, *12*(3), 582–588.
- Siddiqui, A., Fern, A., Dietterich, T. G., Wright, R., Theriault, A., & Archer, D. W. (2018). Feedback-Guided Anomaly Discovery via Online Optimization. *KDD 2018*.
- Sipple, J. (2020). Interpretable, Multidimensional, Multimodal Anomaly Detection with Negative Sampling for Detection of Device Failure. *ICML 2020*.
- Tack, J., Mo, S., Jeong, J., & Shin, J. (2020). CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances. *Advances in Neural Information Processing Systems (NeurIPS 2020)*.
- Tax, D., & Duin, R. (2004). Support vector data description. *Machine Learning*, 45–66.
- Wagstaff, K. L., Lanza, N. L., Thompson, D. R., Dietterich, T. G., & Gilmore, M. S. (2013). Guiding Scientific Discovery with Explanations using DEMUD. *AAAI 2013*.