Machine Learning Methods for Robust Artificial Intelligence Part 3: Anomaly Detection for OOD and Novel Category Detection

Thomas G. Dietterich, Oregon State University

tgd@cs.orst.edu

@tdietterich

Thank you Dan Hendrycks and Balaji Lakshminarayanan for advice and suggestions

The Class So Far

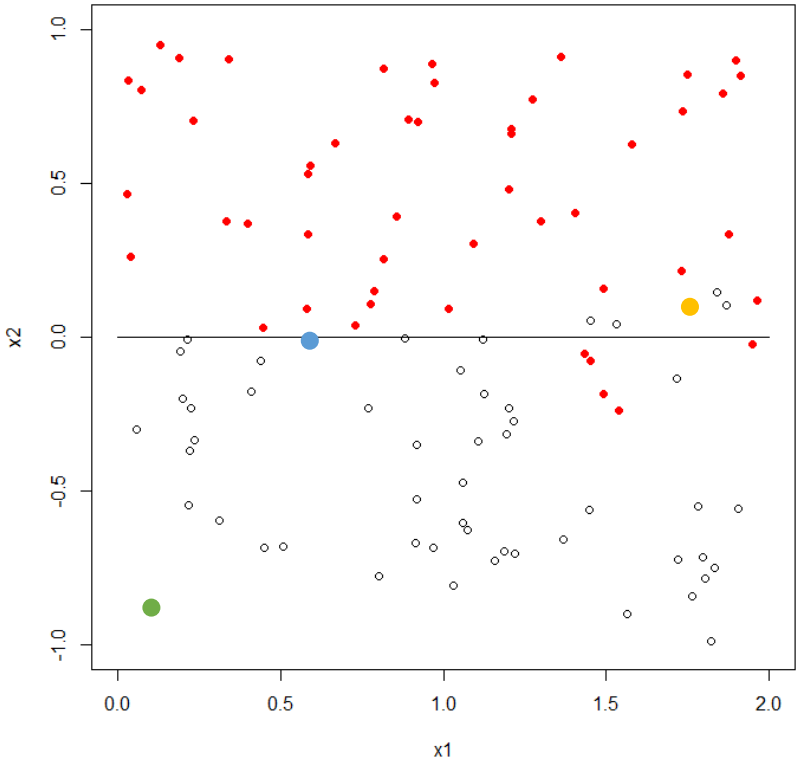
- Lecture 1: Calibrated Probabilities (Closed World)
- Lecture 2: Rejection and Prediction Sets (Closed World)
- Lecture 3: Anomaly Detection for Out-of-Distribution and Novel Category Detection (Open World)

Reminder: Threats to Competent Classifiers

- x_q is near a decision boundary (the features of x_q are ambiguous)
- x_q is in a region with high labeling noise
- x_q is in a region with little training data

  x_q belongs to a class that was not present in the training data: "novel category problem"

Today we focus on case 3 where x_q is an outlier or anomaly



Two Problem Formulations: OOD and Open Category

Out-of-Distribution Problem

- Training:
  - Data: $(x_1, y_1), \ldots, (x_N, y_N)$ drawn from D_0
  - $y_i \in \{1, \ldots, K\}$
- Testing:
  - Data: Mixture D_m of data from D_0 and D_a
  - $(x, y) \sim D_a$ belong to a **different data set**
- Goal:
  - Given a query x_q, does it belong to D_a or D_0?
  - If from D_a, REJECT as alien
  - Else classify using a classifier trained on D_0 data

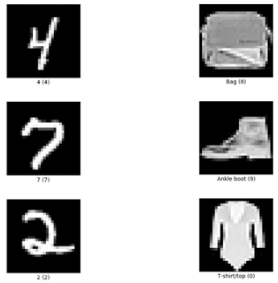
Novel Category / Open Set Problem

- Training:
  - Data: $(x_1, y_1), \ldots, (x_N, y_N)$ drawn from D_0
  - $y_i \in \{1, \ldots, K\}$
- Testing:
  - Data: Mixture D_m of data from D_0 and D_a
  - $(x, y) \sim D_a$ belong to **new classes not seen during training ("alien categories")**
- Goal:
  - Given a query x_q, does it belong to D_a or D_0?
  - If from D_a, REJECT as alien
  - Else classify using a classifier trained on D_0 data

Key Difference: Evaluation Protocol

Out-of-Distribution

- Train on data from domain A
- Test on data from a mix of domain A and domain B
- Example:
  - Train on MNIST
  - Test on MNIST + Fashion-MNIST



Novel Category

- Divide the classes of domain A into known and unknown
- Train on known classes
- Test on all classes
- Example:
  - Train on MNIST {1,2,3,4,5}
  - Test on MNIST {1,2,3,4,5,**6**,**7**,**8**,**9**,**0**}

OOD and Novel Category Metrics

- AUROC: Area under the ROC curve for the binary decision
  - OOD: Domain A vs Domain B
  - Novel Category: Known vs Unknown
- Detection rate at fixed false alarm rate. TPR@10%FAR
  - Maximize correct OOD/Novel Category detections subject to a constraint that the false alarm rate is ≤ 0.10.
- False alarm rate at fixed missed alarm rate: FAR@95%TPR
  - Detect 95% of OOD/Novel Category examples while minimizing false alarms
  - Most relevant to AI Safety and Trustworthy Systems

Comments

- OOD is usually easier than Novel Category because of global differences in image statistics
  - Different image collection methods, different subject matter
  - Novel category images are collected by the same methods and involve very similar objects
- OOD rarely corresponds to a real-world use case
  - If you've trained on MNIST for postal code recognition, you aren't likely to suddenly be given Fashion MNIST images
  - Exceptions:
    - Image collection methods can change: lighting, camera, etc.
- OOD is easier to study
  - Download pre-trained network, apply your technique, evaluate on separate data set
- Novel Category is much more relevant to real-world use cases

Motivating Example: 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
- BugID Project
  - 54 classes of interest to the EPA
  - accuracy $\approx 90\%$
  - Larios, N., Soran, B., Shapiro, L., Martínez-Muños, G., Lin, J., Dietterich, T. G. (2010). **Haar Random Forest Features and SVM Spatial Matching Kernel for Stonefly Species Identification.** *IEEE International Conference on Pattern Recognition (ICPR-2010)*.
  - Lin, J., Larios, N., Lytle, D., Moldenke, A., Paasch, R., Shapiro, L., Todorovic, S., Dietterich, T. (2011). **Fine-Grained Recognition for Arthropod Field Surveys: Three Image Collections**. *First Workshop on Fine-Grained Visual Categorization (CVPR-2011)*
  - Lytle, D. A., Martínez-Muñoz, G., Zhang, W., Larios, N., Shapiro, L., Paasch, R., Moldenke, A., Mortensen, E. A., Todorovic, S., Dietterich, T. G. (2010). **Automated processing and identification of benthic invertebrate samples.** *Journal of the North American Benthological Society*, 29(3), 867-874.

www.epa.gov

Problem: There are $\approx \mathbf{76,000}$ species of freshwater insects worldwide

- 1200 species in US
- Field samples may contain other things
  - small rocks
  - leaves
  - trash
- Simple estimate of equal error rate for novel classes vs. the 54 classes was 20% (in 2011)
  - classifier is not usable without addressing the novel class problem
- We still need to solve this problem

Anomaly Detection

- Definition: An “anomaly” is a data point generated by a process that is different than the process generating the “nominal” data
- Given:
  - Training data: $\{x_1, x_2, \dots, x_N\}$
  - Case 1: All data come from D_0 the “nominal” distribution
  - Case 2: The data come from a mixture of D_0 and D_a the “anomaly” distribution
  - Test data: $\{x_{N+1}, \dots, x_{N+M}\}$ from a mixture of D_0 and D_a
- Find:
  - The data points in the test data that belong to D_a
- Note: D_a need not be a stationary distribution, but we general assume that D_0 is stationary.

Outline

- Theoretical Approaches to Anomaly Detection
- Practical Algorithms for Hand-Crafted Features
- Deep Anomaly Detection
- Setting the Anomaly Detection Threshold

Theoretical Approaches to Anomaly Detection

- **Density Estimation Methods**
  - Surprise: $A(x_q) = -\log P_D(x_q)$
  - Model the joint distribution $P_D(x)$ of the input data points $x_1, \ldots \in D$
  - Issues:
    - Vulnerable to nuisance novelty
    - High-dimensional density estimation requires exponential amounts of training data
- **Quantile Methods**
  - Find a smooth function f such that $\{x: f(x) \geq 0\}$ contains $1 - \alpha$ of the training data
  - Anomaly score $A(x) = -f(x)$
    - Based on kernel techniques, so requires a distance metric and a choice of kernel hyperparameters; vulnerable to irrelevant features
- **Distance-Based Methods**
  - Anomaly score $A(x_q) = \min_{x \in D} \|x_q - x\|$
  - Issues:
    - Requires a good distance metric; vulnerable to irrelevant features
- **Reconstruction Methods**
  - Train an auto-encoder: $x \approx D(E(x))$, where E is the encoder and D is the decoder
  - Anomaly score
    $$A(x_q) = \left\| x_q - D\left(E(x_q)\right) \right\|$$
  - Issues:
    - Vulnerable to irrelevant features

Density Estimation

- Given a data set $\{x_1, \ldots, x_N\}$ where $x_i \in \mathbb{R}^d$
- We assume the data have been drawn iid from an unknown probability density: $x_i \sim P(x_i)$
- Goal: Estimate P
- Anomaly Score: $A(x_q) = -\log P(x_q)$
  - "surprisal" from information theory

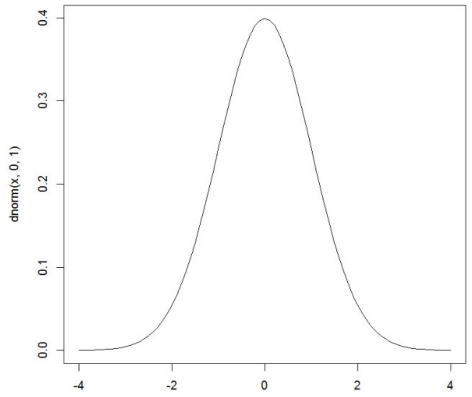
- Requirements
  - $P(x) \geq 0 \; \forall x \in \mathbb{R}^d$ must be non-negative everywhere
  - $\int_{x \in \mathbb{R}^d} P(x) dx = 1$ must integrate to 1

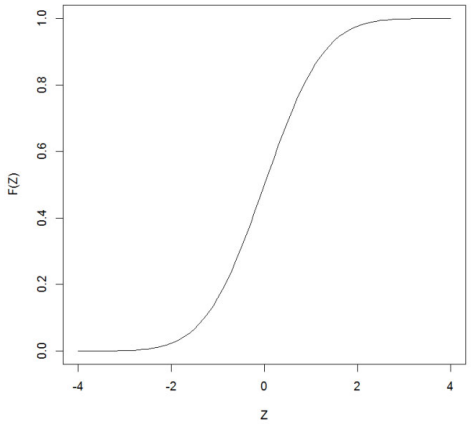
Example: The Gaussian (normal) Distribution

- Normal probability density function (pdf)

$$P(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp -\frac{1}{2}\left[\frac{x-\mu}{\sigma}\right]^2$$

- Normal cumulative distribution function (cdf)
  - $F(z;\mu,\sigma) =$ probability of the event $[-\infty, z]$
  - $F(z;\mu,\sigma) = \int_{-\infty}^{z} P(x;\mu,\sigma)dx$



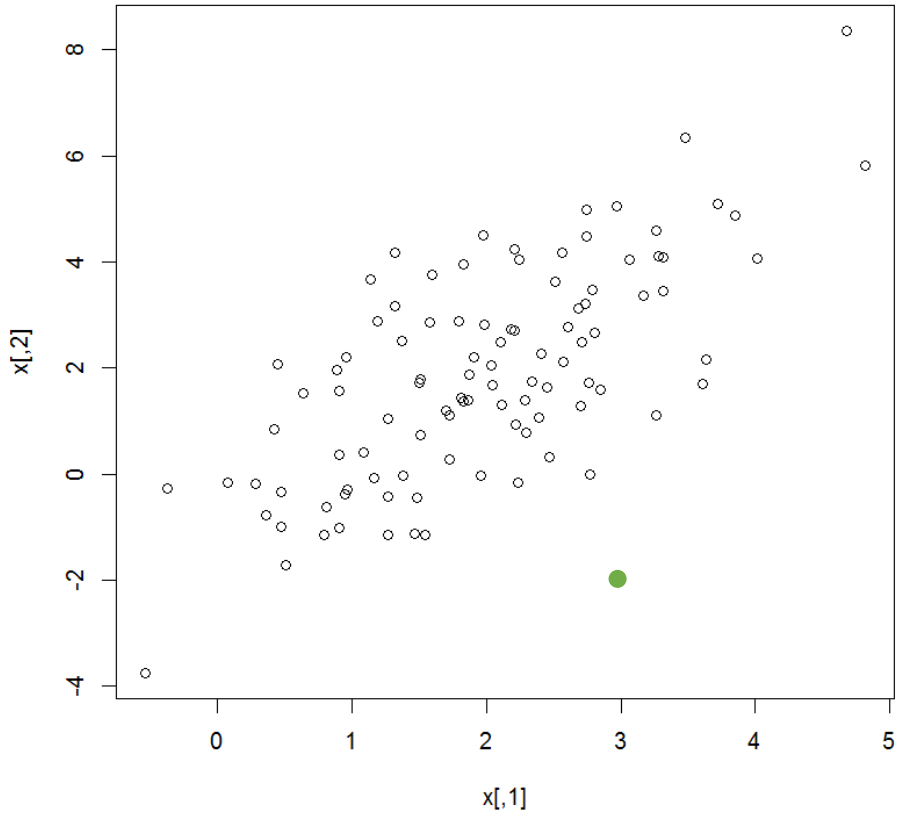


Parametric Density Estimation

- Assume $P(x) = \text{Normal}(x|\mu, \Sigma)$ is the multivariate Gaussian distribution
- $P(x) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp -\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)$
- Fit by computing the first and second moments:
  - $\hat{\mu} = \frac{1}{N}\sum_i x_i$ mean
  - $\hat{\Sigma} = \frac{1}{N}\sum_i (x_i - \hat{\mu})(x_i - \hat{\mu})^\top$ covariance matrix

Example

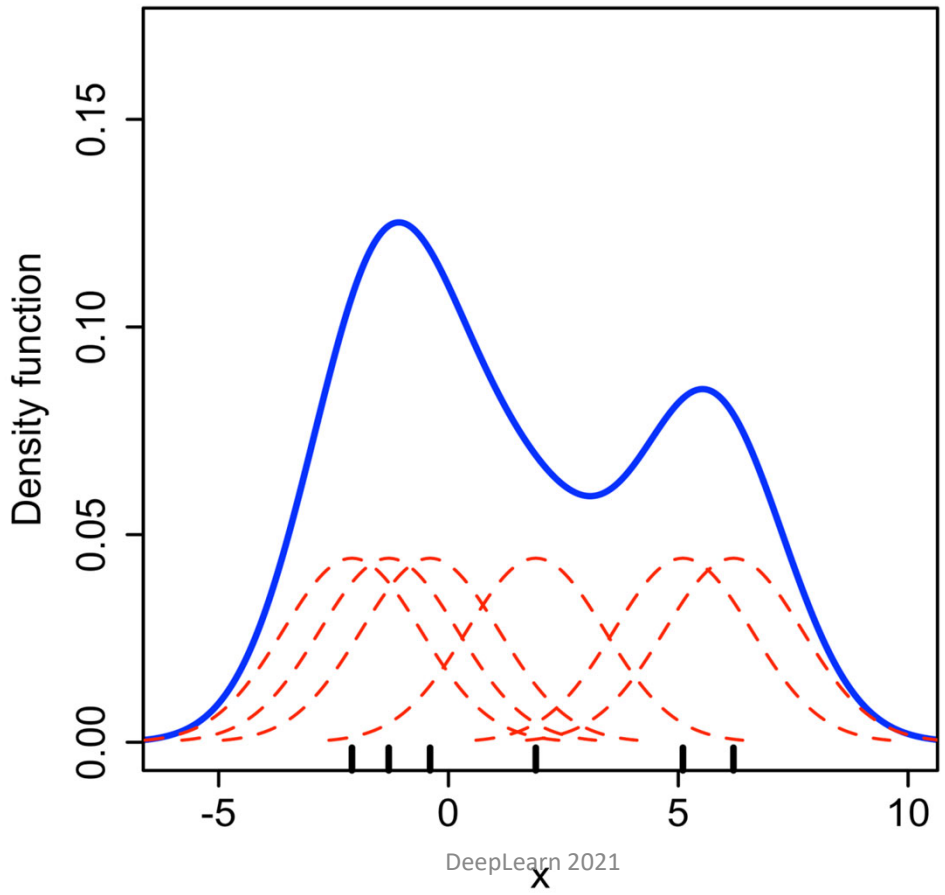
- Sample 100 points from multivariate Gaussian with $\mu = (2,2)$ and $\Sigma = \begin{bmatrix} 1 & 1.5 \\ 1.5 & 4 \end{bmatrix}$
- Estimates:
  - $\hat{\mu} = (1.968731, 1.894511)$
  - $\hat{\Sigma} = \begin{bmatrix} 1.081423 & 1.462467 \\ 1.462467 & 4.000821 \end{bmatrix}$
- Surprisal of $x_q = (3, -2)$ is 9.635
- Surprisal of $\hat{\mu} = 2.229$



Kernel Density Estimation

- Estimate the density as the sum of kernels placed at each point in the training data. The kernel must be a probability density (integrate to 1):
  - $\hat{P}(x) = \frac{1}{N}\sum_{i=1}^{N} k(\|x - x_i\|, \sigma^2)$
  - Often use a Gaussian Kernel $k(x, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{x^2}{2\sigma^2}\right]$
  - Often use a fixed scale σ^2. The scale is also called the “bandwidth”

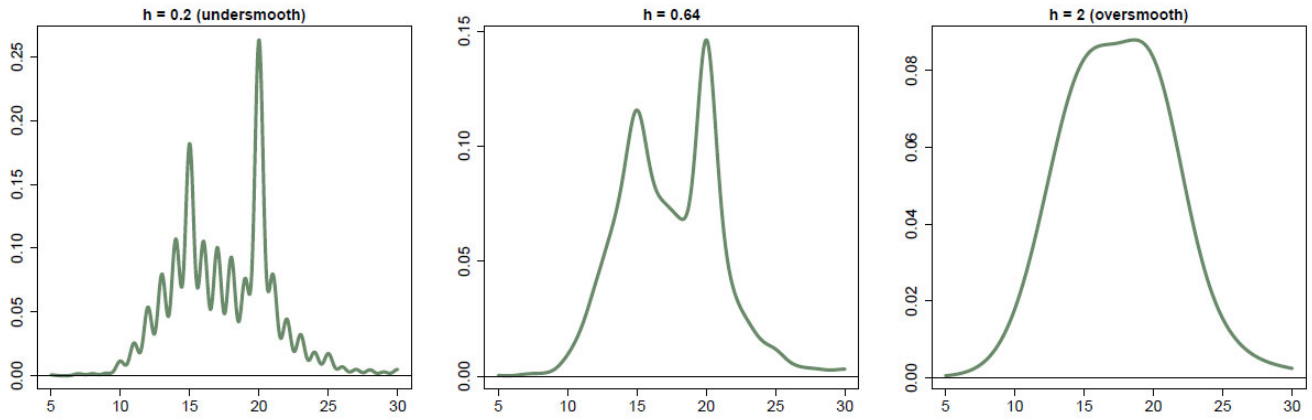
One-Dimensional Example



Source: wikipedia

Design Decisions

- Choice of Kernel: generally not super important as long as it is local
- Choice of bandwidth is very important



Challenges

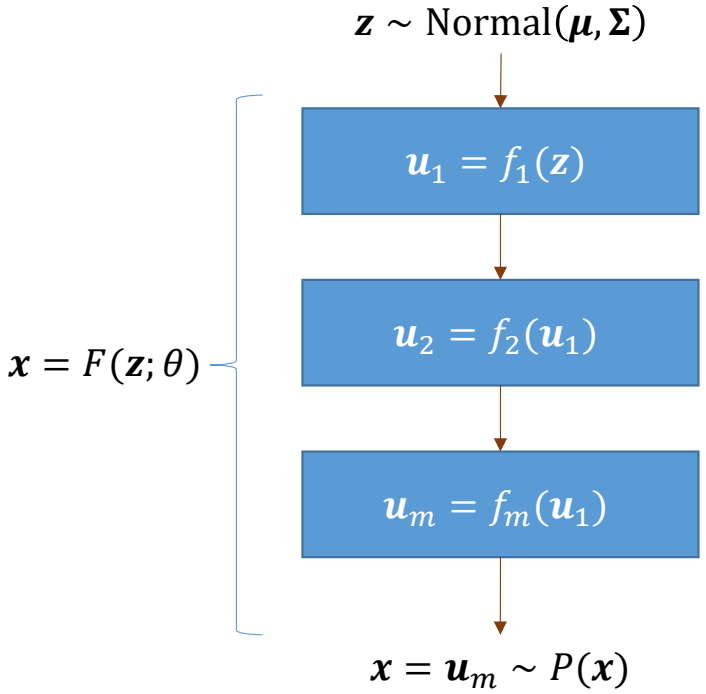
- KDE in high dimensions suffers from the “Curse of Dimensionality”
- The amount of data required to achieve a desired level of accuracy scales exponentially with the dimensionality d of the problem: $\exp \frac{d+4}{2}$

Deep Neural Networks for Density Estimation

- Idea: Transform a Normal density into a density that fits the data. Adjust the parameters θ of the model F to maximize the likelihood of the data

$$\sum_i \log P(\boldsymbol{x}_i)$$

- If F is invertible, then $P(\boldsymbol{x}) = \text{Normal}(F^{-1}(\boldsymbol{x}); \boldsymbol{\mu}, \boldsymbol{\Sigma})$

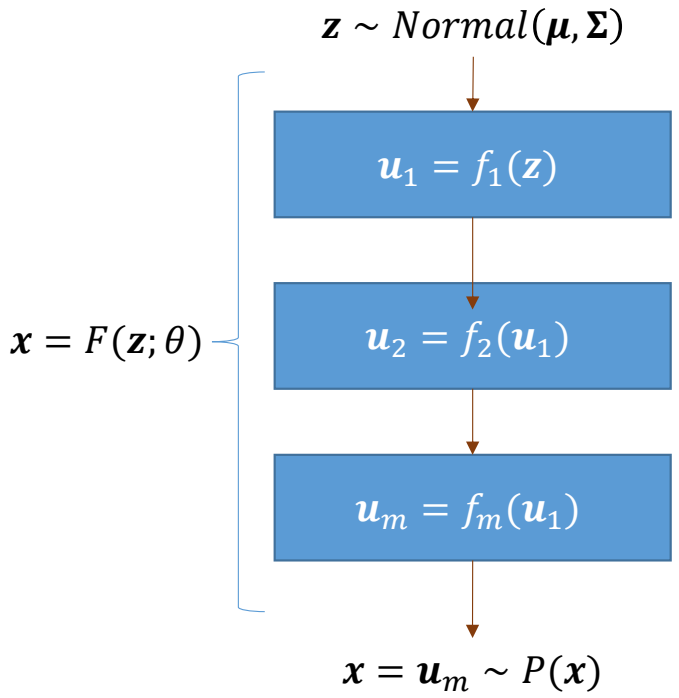


Constraint: Must Preserve Probabilities of Events

- Recall
  - Let $P(\boldsymbol{x})$ be a probability density (a measurable function that integrates to 1)
  - An *event* is a region V, and its probability mass is
  - $\int_{\boldsymbol{x}\in V} P(\boldsymbol{x})d\boldsymbol{x} = \Pr[V]$

- We need to ensure that for any region V in the input space, the corresponding integral in $\mathcal{Z}$ space gives the same answer

$$\int_{\boldsymbol{x}\in V} F(\boldsymbol{x})d\boldsymbol{x} = \int_{\boldsymbol{z}\in F^{-1}(V)} \text{Normal}(\boldsymbol{z};\boldsymbol{\mu},\boldsymbol{\Sigma})d\boldsymbol{z}$$



Change of Variables Formula

- $P(\mathbf{x}) = \text{Normal}(F^{-1}(\mathbf{x}); 0, \mathbf{I}) \left|\det \left[\frac{\partial F^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]\right|$

  where $\det \left[\frac{\partial F^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]$ is the Jacobian of F^{-1}

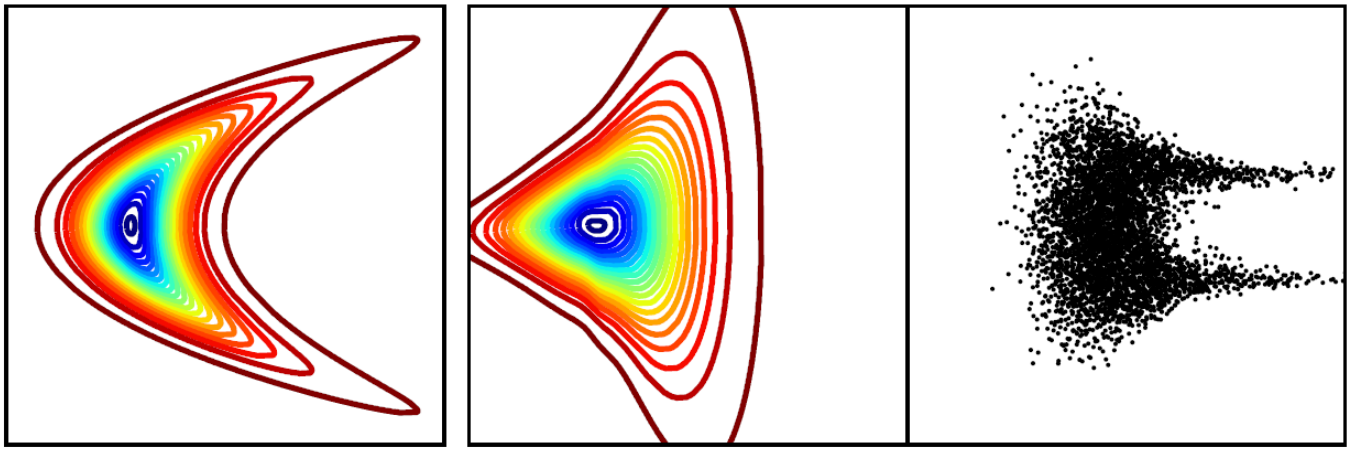
This compensates for any stretching or compression of the space

Constructing Deep Density Models

- Ensure that each f is invertible and has an easy-to-compute Jacobian
- Example: Masked Auto-Regressive Flow (Papamarkarios, et al 2017)

Stacking MAFs

- One MAF network is often not sufficient

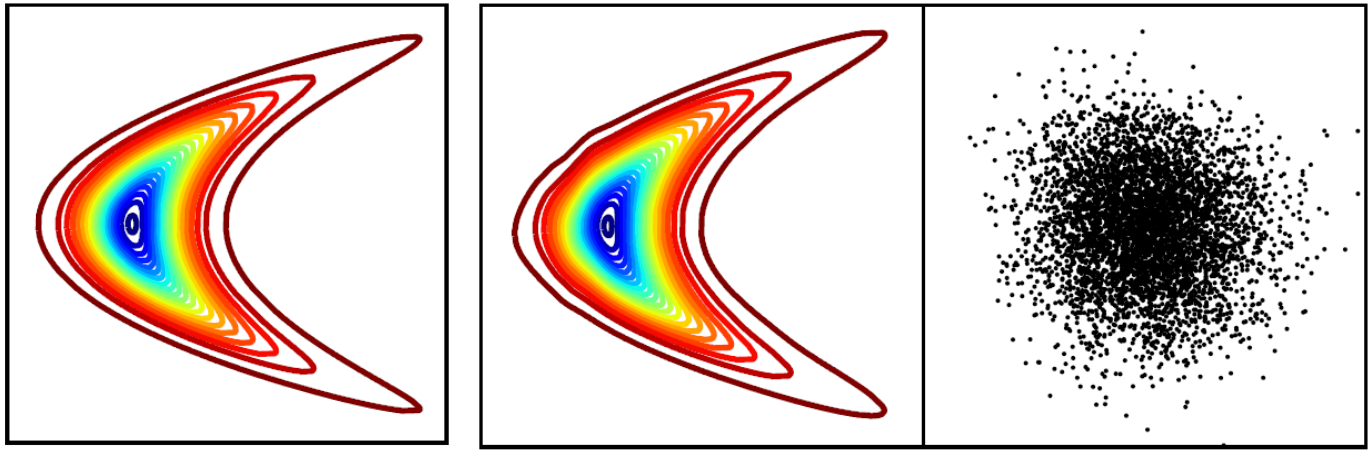


True Density

Fitted Density from single MAF network

Distribution of the $\mathbf{z}$ values

Stack MAFs until the $\mathbf{z}$ values are Normal(0,I)



True Density

Fitted Density from stack of 5 MAFs

Distribution of the $\mathbf{z}$ values

Test Set Log Likelihood

	POWER	GAS	HEPMASS	MINIBOONE	BSDS300
Gaussian	-7.74 ± 0.02	-3.58 ± 0.75	-27.93 ± 0.02	-37.24 ± 1.07	96.67 ± 0.25
MADE	-3.08 ± 0.03	3.56 ± 0.04	-20.98 ± 0.02	-15.59 ± 0.50	148.85 ± 0.28
MADE MoG	$\mathbf{0.40 \pm 0.01}$	8.47 ± 0.02	$\mathbf{-15.15 \pm 0.02}$	-12.27 ± 0.47	153.71 ± 0.28
Real NVP (5)	-0.02 ± 0.01	4.78 ± 1.80	-19.62 ± 0.02	-13.55 ± 0.49	152.97 ± 0.28
Real NVP (10)	0.17 ± 0.01	8.33 ± 0.14	-18.71 ± 0.02	-13.84 ± 0.52	153.28 ± 1.78
MAF (5)	0.14 ± 0.01	9.07 ± 0.02	-17.70 ± 0.02	$\mathbf{-11.75 \pm 0.44}$	155.69 ± 0.28
MAF (10)	0.24 ± 0.01	$\mathbf{10.08 \pm 0.02}$	-17.73 ± 0.02	-12.24 ± 0.45	154.93 ± 0.28
MAF MoG (5)	0.30 ± 0.01	9.59 ± 0.02	-17.39 ± 0.02	$\mathbf{-11.68 \pm 0.44}$	$\mathbf{156.36 \pm 0.28}$

Priyank, Kobyzev, Yu & Brubaker (ICML 2020): Use a Student t distribution instead of a Gaussian. This allows you to generate distributions with heavy tails, which Gaussians cannot do

Potential Bug
(Le Lan & Dinh, 2020)

- We want to use $-\log P(x)$ as our anomaly score $A(x)$
- Formally, $-\log P(x)$ applies only to the *probability mass of an event* $\Pr[V]$. Under a probability density $P(X)$, the event that $X = x$ has zero probability mass
- Solution: Consider a *region* surrounding x: $V(x) = \{x' : \|x - x'\| < \rho\}$
  - $\Pr[X \in R(x)] = \int_{x \in R(x)} P(x)dx$
- When we use $-\log P(X = x)$ as an anomaly score, we are assuming that the density at x is a good approximation to $\Pr[X \in V(x)]$
- This assumption is broken in most deep density models, because the invertible flow F changes the distances between points, so a local neighborhood of x may have a bizarre non-local shape in z: $F^{-1}(V)$

Lesson

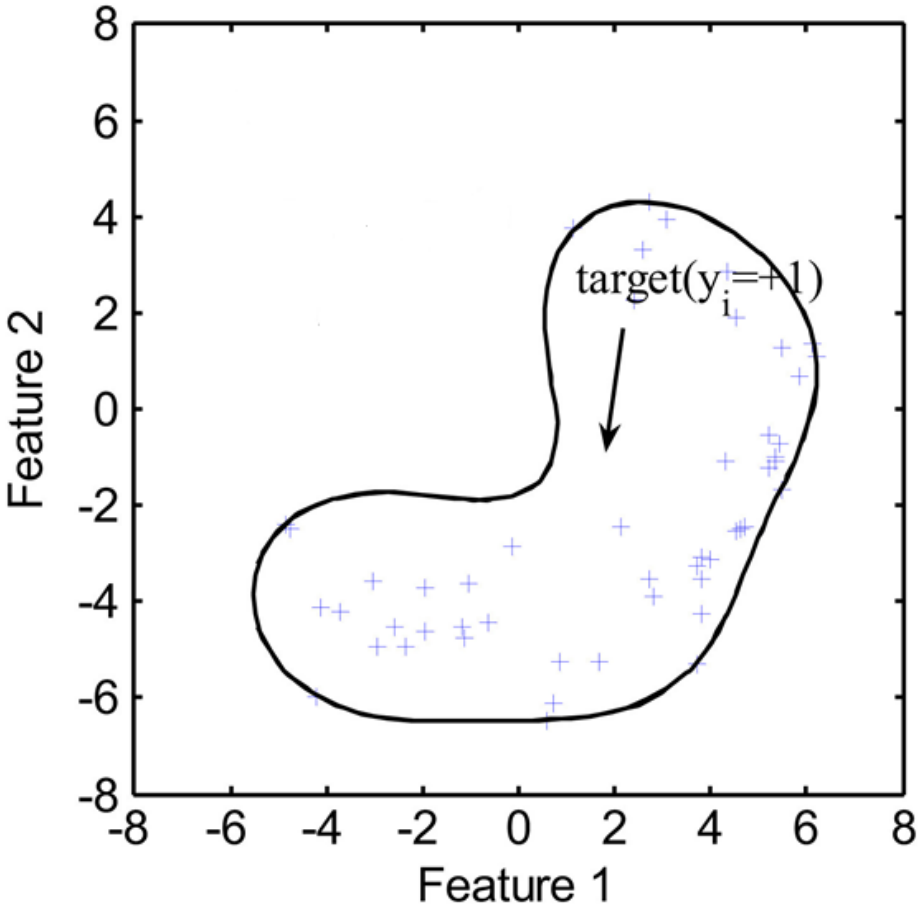
- The representational space matters
- We want to apply density estimation in a meaningful space
  - This is NOT the input image/pixel space
  - We DO NOT want to compute $\Pr[x \in V(x)]$
- We want to learn a good latent space $\mathcal{Z}$ such that images of similar contents (same objects, same class, etc.) are close together
- Then apply density estimation in that space

Approach 2: Quantile Methods

- Vapnik's principle again: We only need to estimate the "decision boundary" between nominal and anomalous
- Surround the data by a function f that captures $1 - \epsilon$ of the training data
  - One-Class Support Vector Machine (OCSVM)
    - f is a hyperplane in "kernel space"
  - Support Vector Data Description (SVDD)
    - f is a sphere is "kernel space"
- Closely related to kernel density estimation:

$$f(x) = \sum_{x_i \in SV} \alpha_i k(x, x_i) - \rho$$

where SV is the set of "support vectors". These are a carefully-selected subset of the training data points. ρ is a scalar parameter



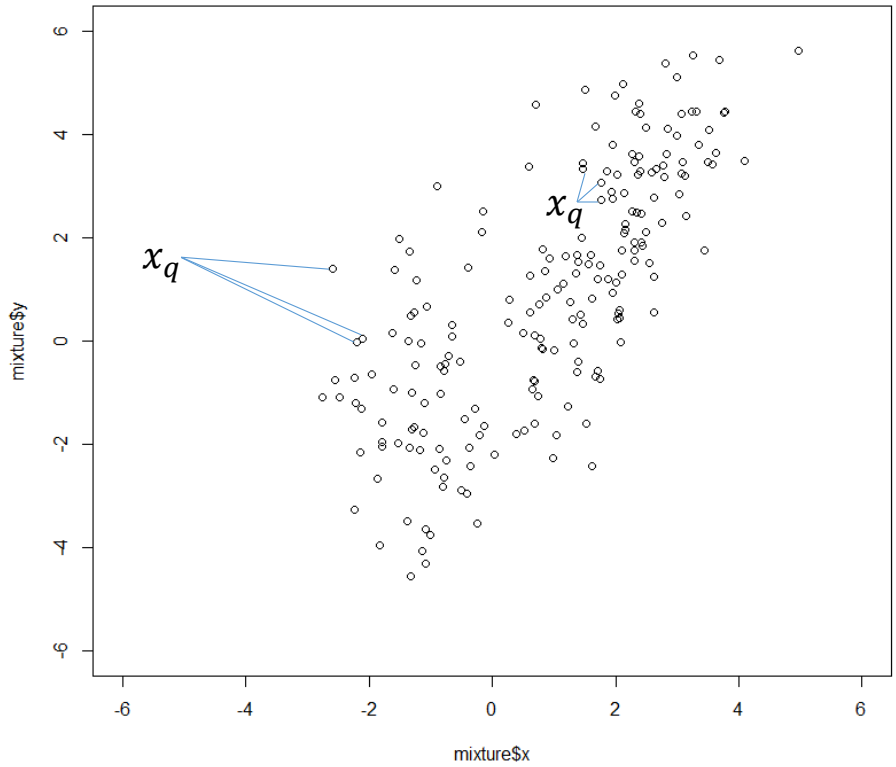
Approach 3: Distance-Based Methods

- Do we really need to estimate probability densities?
- In most applications, we just need a way of ranking the anomalies

- Define a distance $d(x_i, x_j)$
- $A(x_q) = \min_{x \in D} d(x_q, x)$

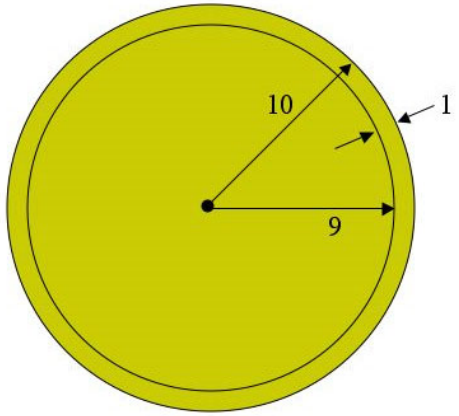
- This can be made more robust by looking at the average distance to the k-nearest points
  - "k-nn anomaly detection"

- This can be normalized by dividing by the distance of each neighbor to *their* k-nearest neighbors
  - "Local Outlier Factor (LOF)"



Challenges with Distance Metrics

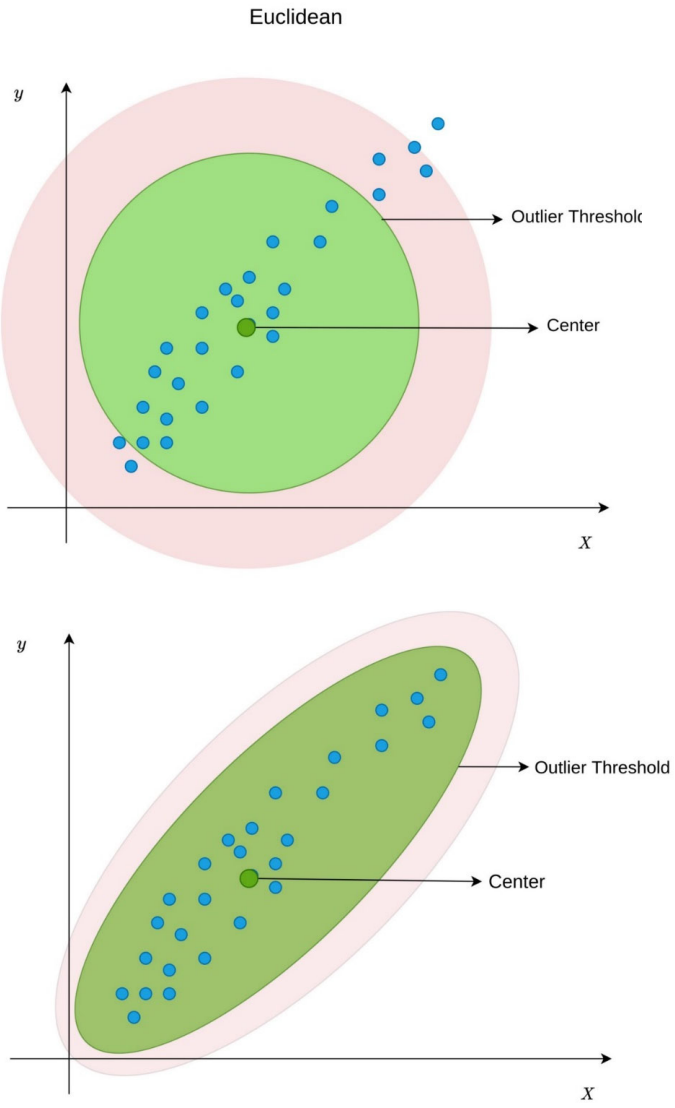
- Correlated or Redundant Features
  - If a feature appears twice, then it contributes to the distance twice, which gives it too much weight
  - If a pair of features is correlated, they have too much weight
- Distances in high dimensions are counter-intuitive
  - Suppose data points are uniformly distributed within the volume of a d-dimensional hypersphere
  - Most of the points will be very close to the surface of the sphere.
    - in $2d$, the shell at right contains 27% of the volume
    - in $100d$, the shell contains 99.9973% of the volume
  - The distances between pairs of points tends to cluster tightly (yet another version of the Central Limit Theorem)
- Therefore: Reduce dimensionality as much as possible



https://ee.stanford.edu/~hellman/playground/hyperspheres/hyper01.html

Computing Distances

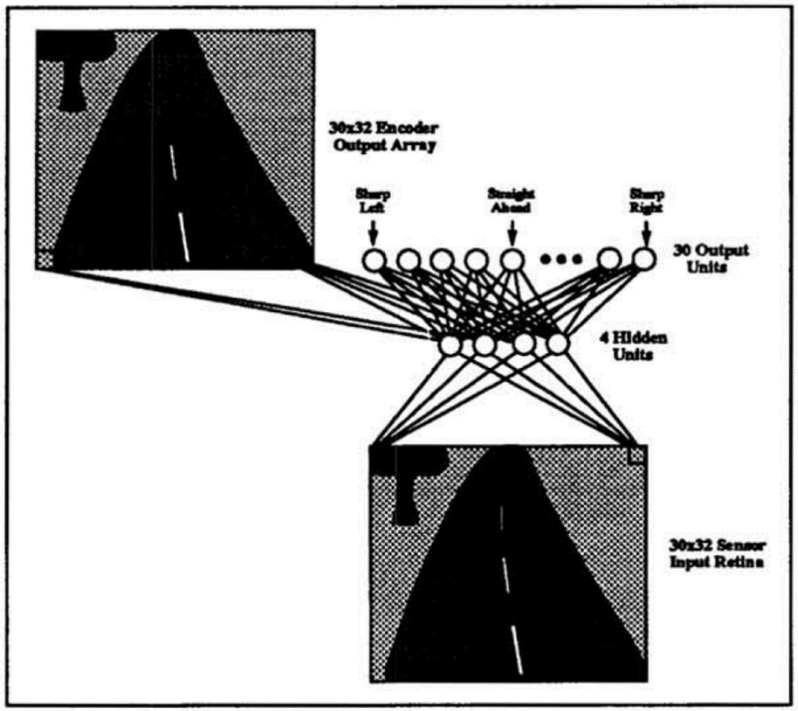
- Mahalanobis Distance
  - Fit a multi-variate Gaussian distribution to your data
    - Mean vector: $\boldsymbol{\mu}$
    - Covariance matrix: $\boldsymbol{\Sigma}$
  - Compute the Mahalanobis Distance:
    - $d_{MH}(x, x') = \sqrt{(x - \boldsymbol{\mu})^{\top} \boldsymbol{\Sigma}^{-1} (x - \boldsymbol{\mu})}$
    - This handles the correlation structure of the data
  - Points of constant MD are ellipsoids in the original space



https://towardsdatascience.com/multivariate-outlier-detection-in-python-e946cfc843b3

Approach 4: Reconstruction Methods

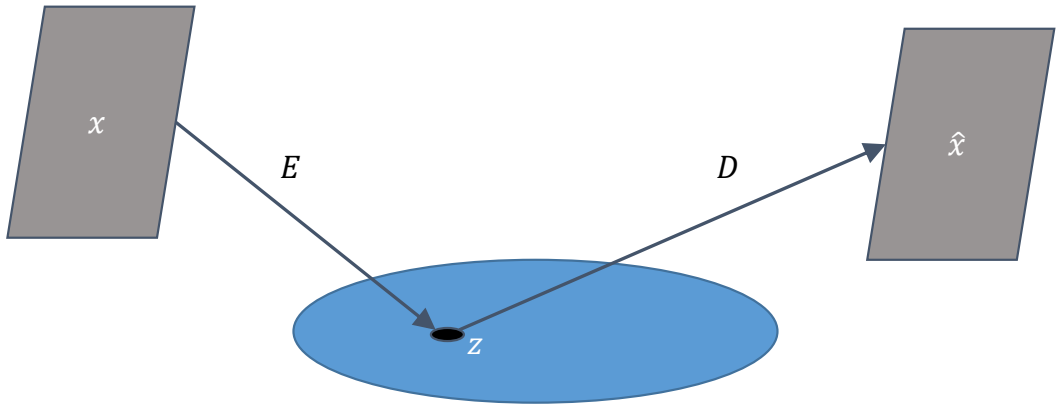
- NavLab self-driving van (Pomerleau, 1992)
  - Primary head: Predict steering angle from input image
  - Secondary head: Predict the input image ("auto-encoder")
  - $A(x_q) = \|x_q - \hat{x}_q\|$
  - If reconstruction is poor, this suggests that the steering angle should not be trusted
- Principle: Anomaly Detection through Failure
  - Define a task on which the learned system should fail for anomalies



Pomerleau, NIPS 1992

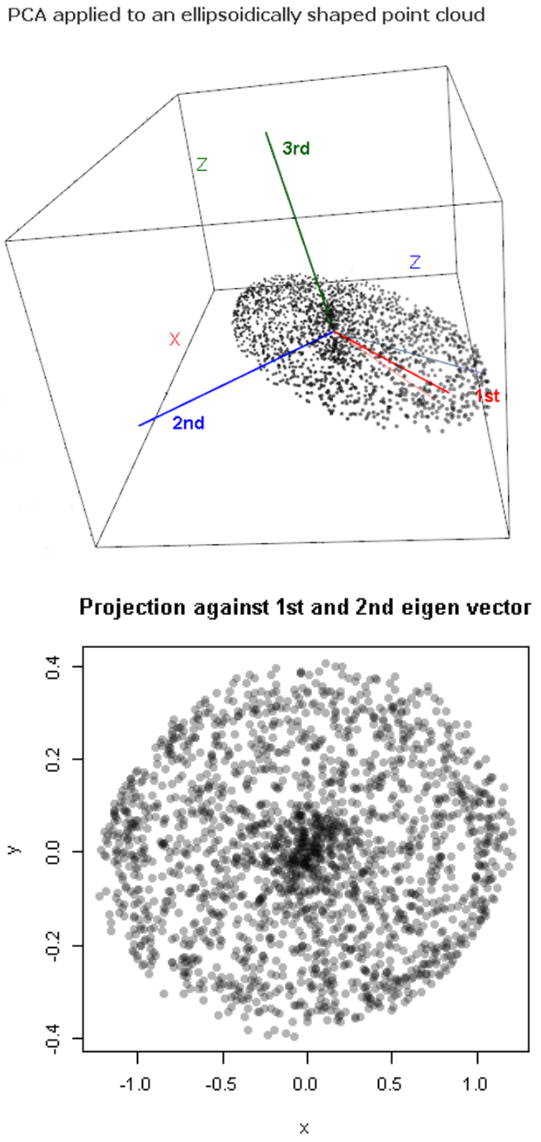
Autoencoders

- $z = E(x)$
- $\hat{x} = D(z)$



Linear Autoencoder == Principal Component Analysis

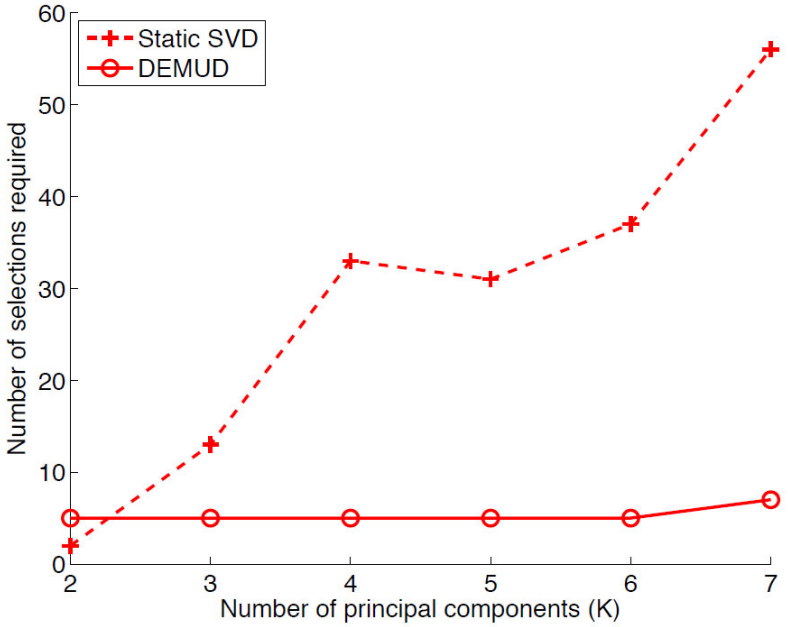
- PCA:
  - Let the input dimension be d
  - Choose a latent dimension ℓ
  - Find the $d \times \ell$ matrix W that minimizes the squared reconstruction error
  - $\min_W \sum_i \|x_i - WW^\top x_i\|^2$
    - This can be done using the Singular Value Decomposition
    - It can also be viewed as fitting a multi-variate Gaussian to the data and then keeping only the ℓ dimensions of highest variance



https://www.joyofdata.de/blog/illustration-of-principal-component-analysis-pca/

Application: Finding Unusual Chemical Spectra

- NASA Mars Science Laboratory ChemCam instrument
  - Collects 6144 spectral bands on rock samples from 7m distance using laser stimulation
  - Goal: active learning to find interesting spectra
  - DEMUD
    - Incremental PCA applied to samples one at a time
    - Fit only to the samples labeled as "uninteresting" by the user
    - Show the user the most un-uninteresting sample (sample with highest PCA reconstruction error)
  - Rapidly discovers interesting samples
  - Wagstaff, et al. (2013)



(a) Effort required to discover magnesite.

Outline

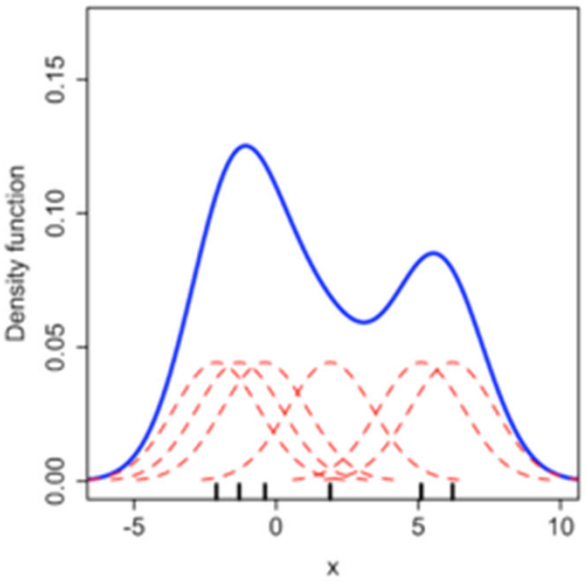
- Theoretical Approaches to Anomaly Detection
- Practical Algorithms for Hand-Crafted Features
- Deep Anomaly Detection
- Setting the Anomaly Detection Threshold

Density Estimation

- RKDE: Robust Kernel Density Estimation (Kim & Scott, 2008)
  - Selects a subset of the input points and places kernels only on those points
  - Robust to the presence of anomalies in the training data
- EGMM: Ensemble Gaussian Mixture Model (our group)
  - Fit a mixture of Gaussian mixture models
- LODA: Lightweight Online Detector of Anomalies (Pevny, 2016)
  - Fit an ensemble of histogram density estimators to sparse, random one-dimensional projections of the data

Robust Kernel Density Estimation

- Kernel Density Estimation
  - Let $k_\sigma(x, x')$ be a positive semi-definite kernel such as the Gaussian kernel or the Student-t-kernel
  - $\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} k_\sigma(x, x_i)$
- Let $\Phi(x)$ be the feature function corresponding to k_σ
  - $k_\sigma(x, x') = \langle \Phi(x), \Phi(x') \rangle$
- Then the KDE is the solution to a least squares problem in Hilbert space:
  - $\hat{p} = \min_{g \in \mathcal{H}} \sum_{i=1}^{N} \|\Phi(x_i) - g(x_i)\|_{\mathcal{H}}^2$
- We can make this more robust by replacing the square loss with a robust loss

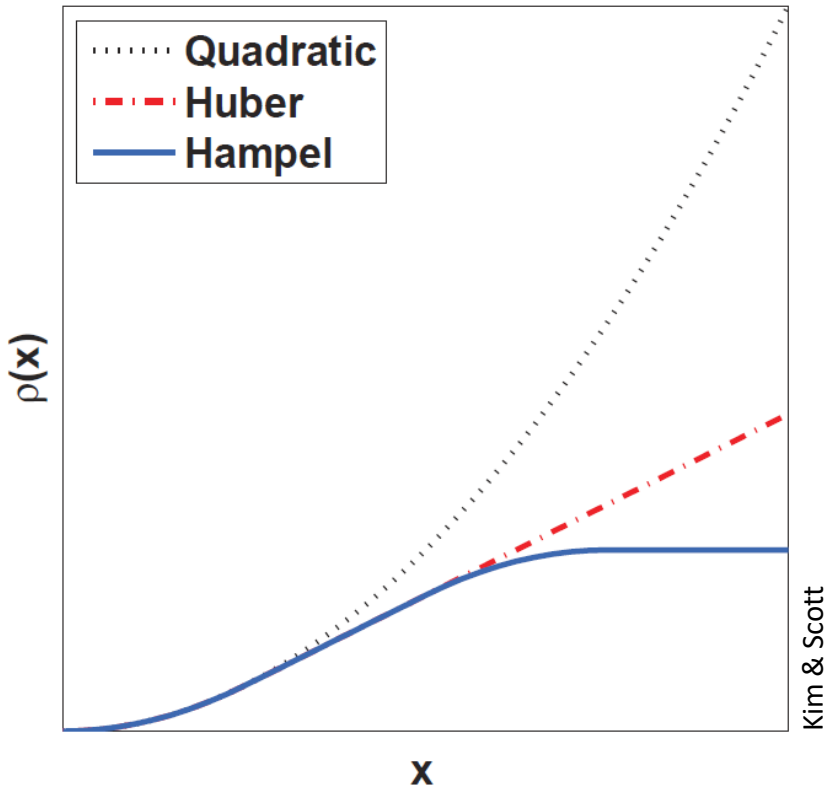


Wikipedia

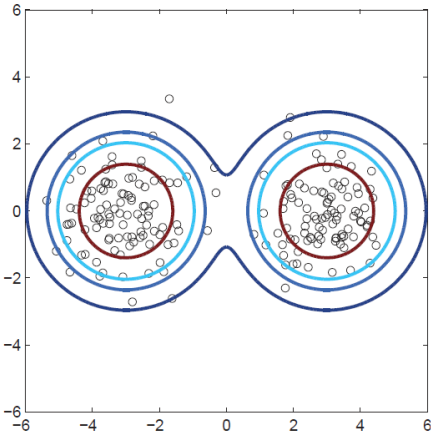
Robust Loss Functions

$$\hat{p} = \underset{g \in \mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{N} \rho(\|\Phi(x_i) - g(x_i)\|_{\mathcal{H}})$$

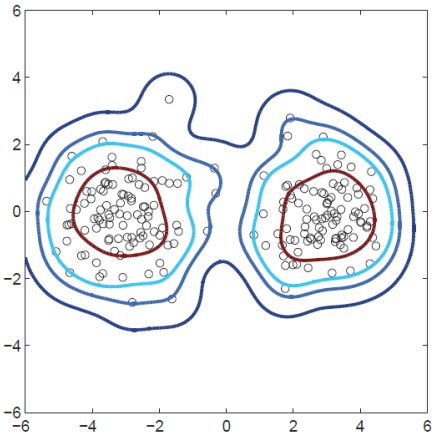
This can be solved by Iteratively Reweighted Least Squares



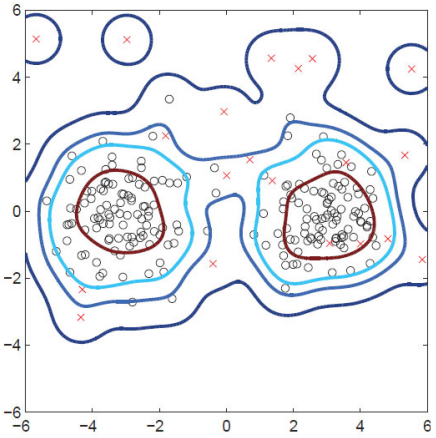
Example:
Mixture of 2 Gaussians



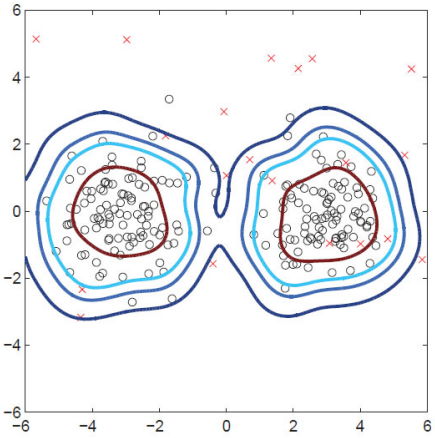
(a) True density



(b) KDE without outliers



(c) KDE with outliers



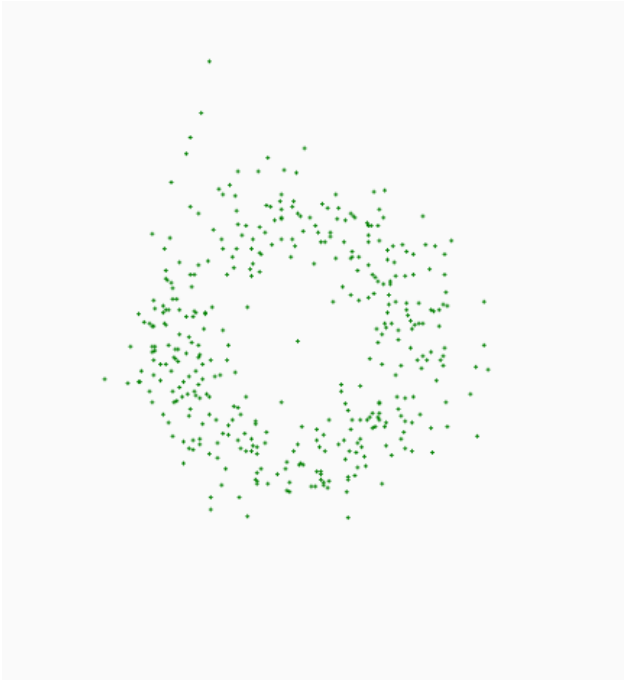
(d) RKDE with outliers

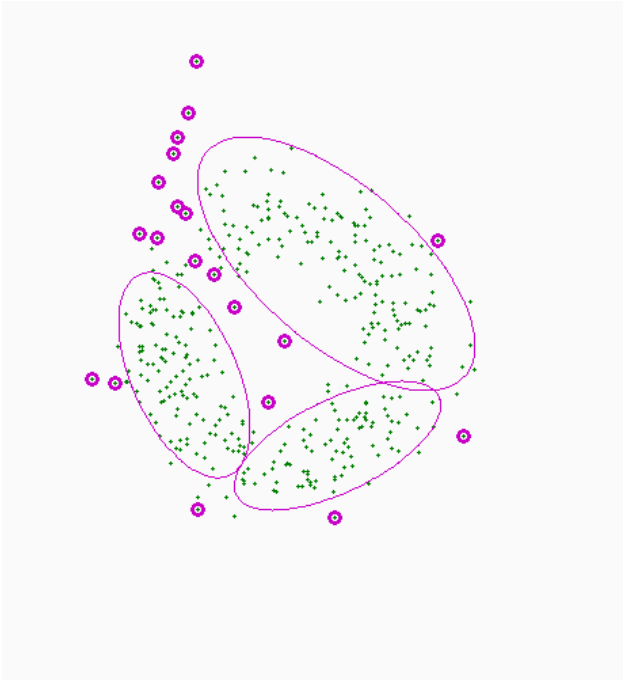
Kim & Scott

Ensemble of Gaussian Mixture Models

- $P(x) = \sum_{k=1}^{K} p_k \cdot \text{Normal}(x|\mu_k, \Sigma_k)$

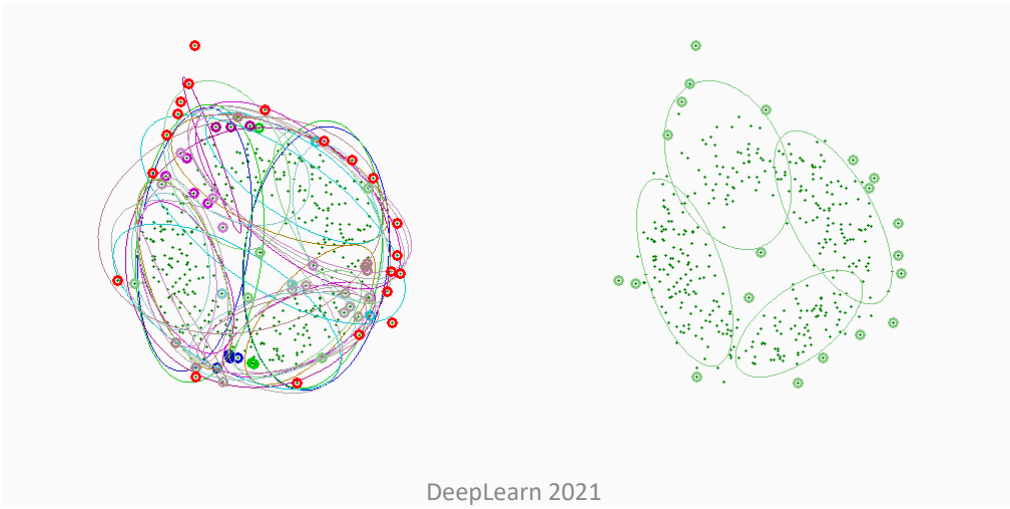
K=3





Ensemble of GMMs

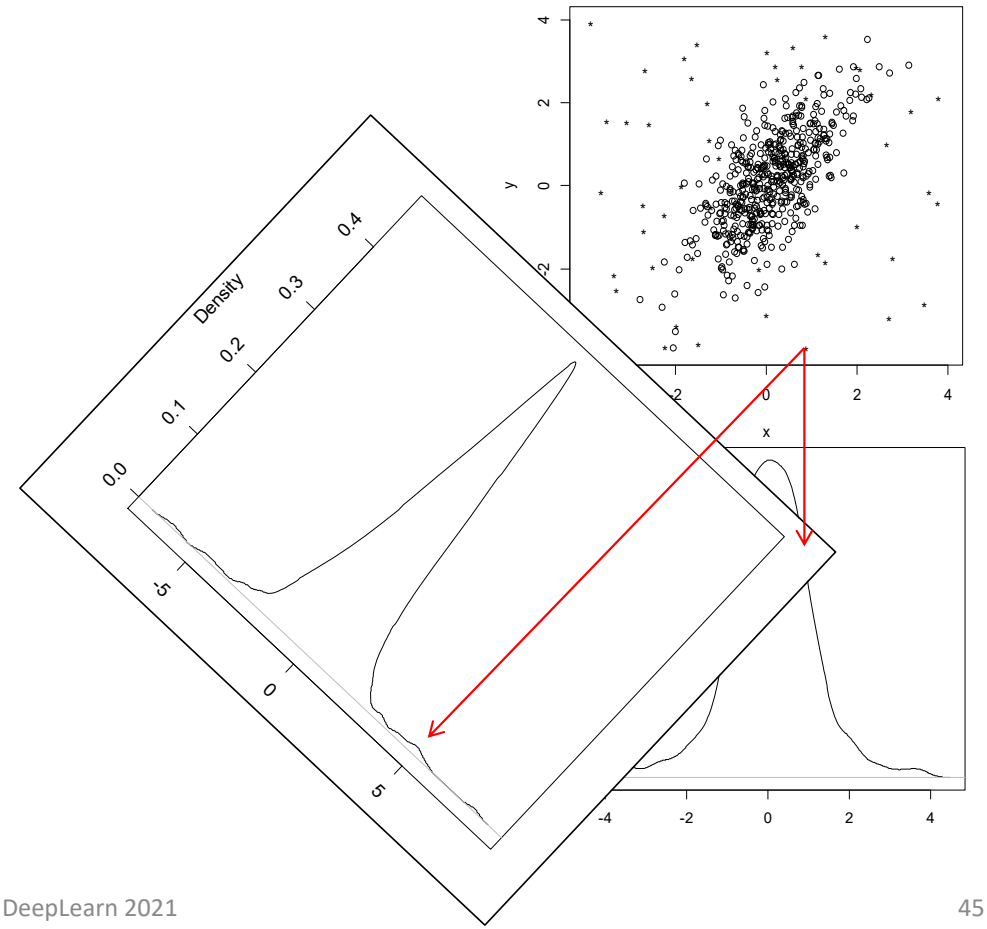
- Train M independent Gaussian Mixture Models
- Train model $m = 1, \ldots, M$ on a bootstrap replicate of the data
- Vary the number of clusters K
- Delete any model with log likelihood < 70% of best model
- Compute average surprise: $-\frac{1}{M}\sum_m \log P_m(x_i)$



LODA: Lightweight Online Detector of Anomalies

[Pevny, 2016]

- $\Pi_1, \ldots, \Pi_M$ set of M sparse random projections
  - Let $w_m = (0, \ldots, 0)$
  - Choose $\sqrt{d}$ elements of w_m and set them to normal random variate
  - $\Pi_m(x) = w_m \cdot x$
- $f_1, \ldots, f_M$ corresponding 1-dimensional density estimators
  - Pevny uses optimal histograms
- $S(x) = -\frac{1}{M}\sum_m \log f_m(x)$ average "surprise"



Quantile-Based Methods

- OCSVM: One-class SVM (Schoelkopf, et al., 1999)
- SVDD: Support Vector Data Description (Tax & Duin, 2004)

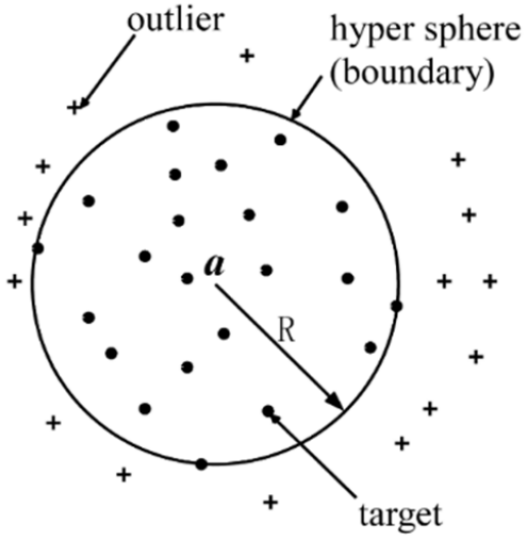
One-Class Support Vector Machine

(Schoelkopf, Williamson, Smola, Shawe-Taylor, Platt, NIPS 2000)

- Given a kernel $k(x, x')$, map the data into the feature space $\Phi(x)$ and find a hyperplane that is as far from the origin as possible and separates $1 - \nu$ of the data points from the origin
- Solution to the following
  - $\min_{w,\xi,\rho} \frac{1}{2}\|w\|^2 + \frac{1}{\nu N}\sum_{i=1}^{N} \xi_i - \rho$
  - Subject to $\left(w \cdot \Phi(x_i)\right) \geq \rho - \xi_i; \xi_i \geq 0$
- The discriminant function is
  - $f(x) = \sum_i \alpha_i k(x, x_i) - \rho$
  - It is positive for nominal points and negative for anomalies

Support Vector Data Description (Tax & Duin, 2004)

- Find the smallest hypersphere in feature space that contains $1 - \nu$ of the data points
- Solution to
  - $\min_{R,a} R^2 + C \sum_{i=1}^{N} \xi_i$
  - Subject to $\|x_i - a\|^2 \leq R^2 + \xi_i;\ \ \xi_i \geq 0$
- Only works well for the Gaussian kernel



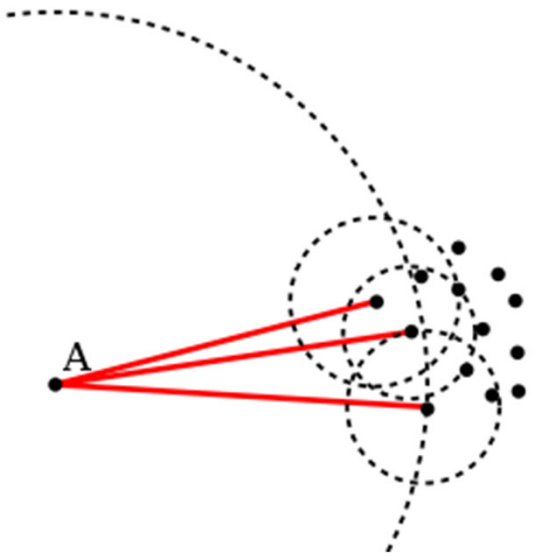
Saeid Homayouni

Distance-Based Methods

- Local Outlier Factor (Breunig, et al., 2000)
  - Normalized distance to k-nearest neighbors
- Isolation Forest (Liu, et al., 2011)
  - Tree-based method for approximating the L1 distance between the query and the training data

LOF: Local Outlier Factor
(Breunig, et al., 2000)

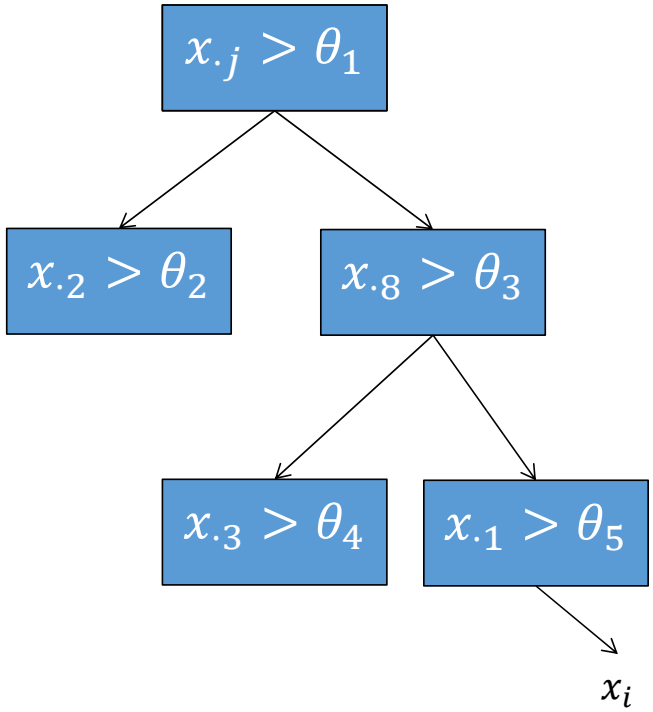
- Distance from x to its k-th nearest neighbor divided by the average distance of each of those neighbors to their k-th nearest neighbors
- [The actual calculation is slightly more complex.]



Breunig, et al.,

Isolation Forest [Liu, Ting, Zhou, 2011]

- Construct a fully random binary tree
  - choose attribute j at random
  - choose splitting threshold θ_1 uniformly from $\left[\min(x_{.j}), \max(x_{.j})\right]$
  - 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



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]
[Emmott, MS Thesis. 2020]

Benchmarking Methodology

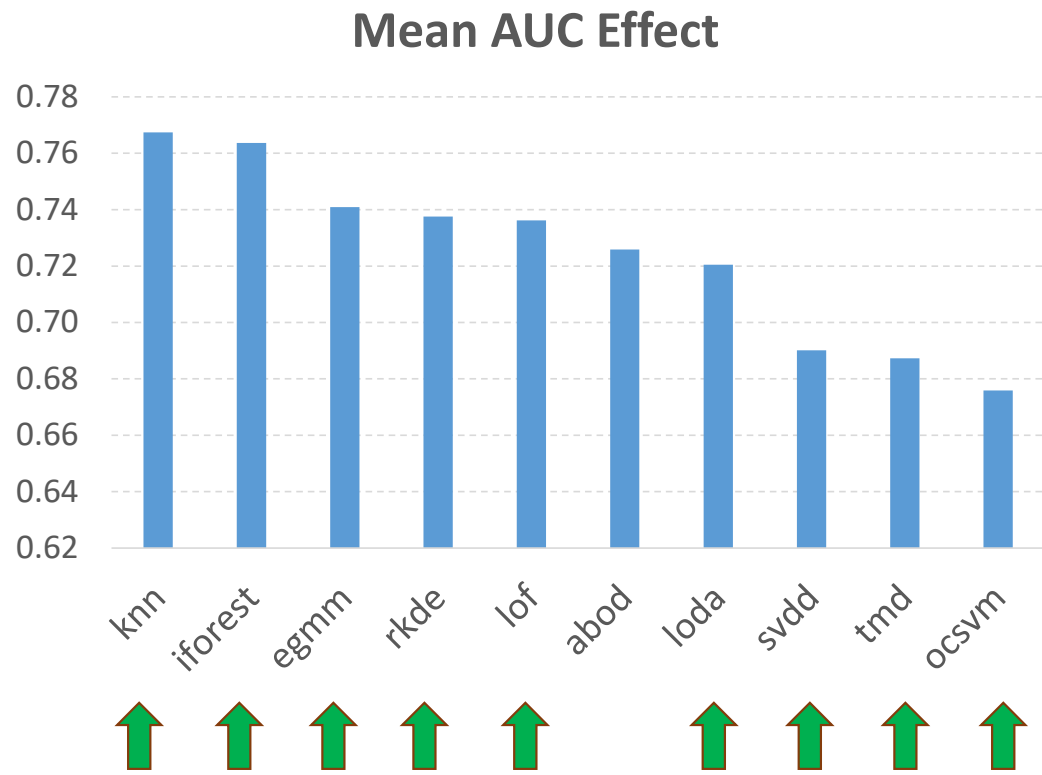
- Select 19 data sets from UC Irvine repository
- Choose one or more classes to be “anomalies”; the rest are “nominals”
- Manipulate
  - Relative frequency
  - Point difficulty
  - Irrelevant features
  - Clusteredness
- 20 replicates of each configuration
- Result: 11,888 Non-trivial Benchmark Datasets

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
- Worst methods: Kernel-based OCSVM and SVDD



Employs a distance

Outline

- Theoretical Approaches to Anomaly Detection
- Practical Algorithms for Hand-Crafted Features
- Deep Anomaly Detection
- Setting the Anomaly Detection Threshold

Deep Anomaly Detection

- Deep Learning learns a representation that is sufficient for the task
  - In classification, this means the representation separates the labeled training data by class
  - However, this does not necessarily provide a good representation for detecting anomalies
- Formally
  - Discriminative classification seeks to model $P(y|x)$
  - Anomaly detection needs $P(x)$ or else $P(x|y = k)$ for each k

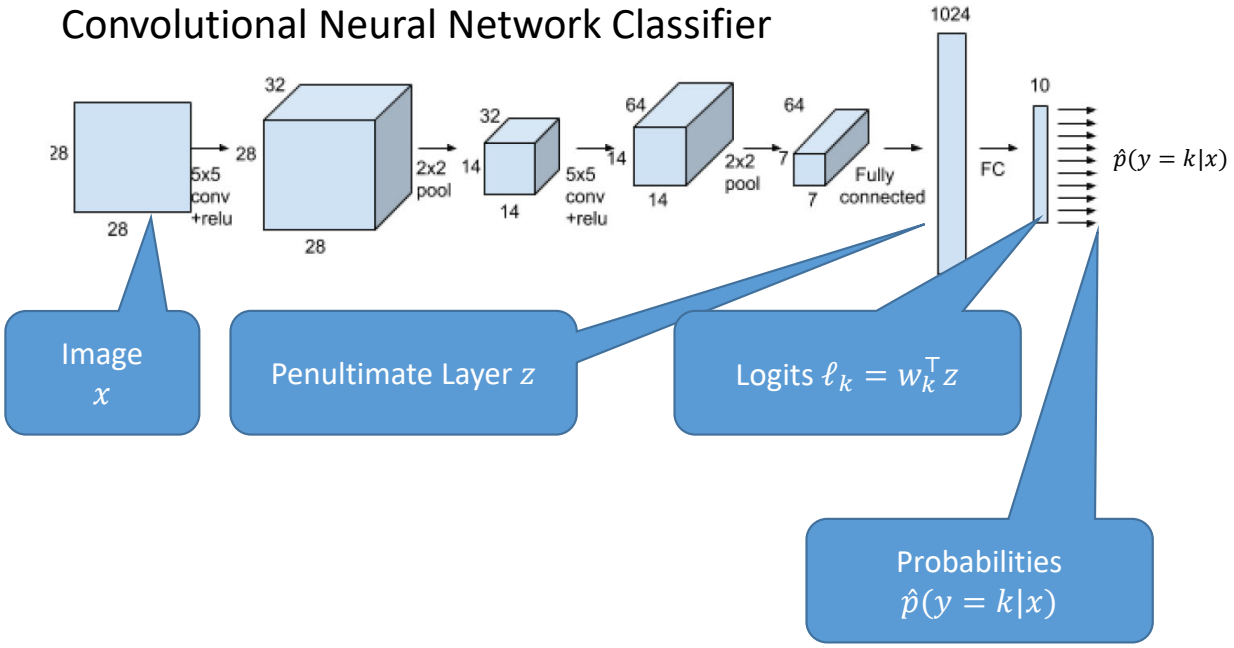
Representation Learning Approaches

- Method 1: Supervised Classification
- Method 2: Supervised Classification with Modified Output Layer
- Method 3: Hybrid Methods
- Method 4: Instance-Contrastive Learning

Method 1: Train Standard Multiclass Classifier

- Input image x
- Network backbone, also called the "encoder": $z = E(x)$
- Latent representation z
- Logits $\ell_k = w_k^\top z$
- Predicted probabilities

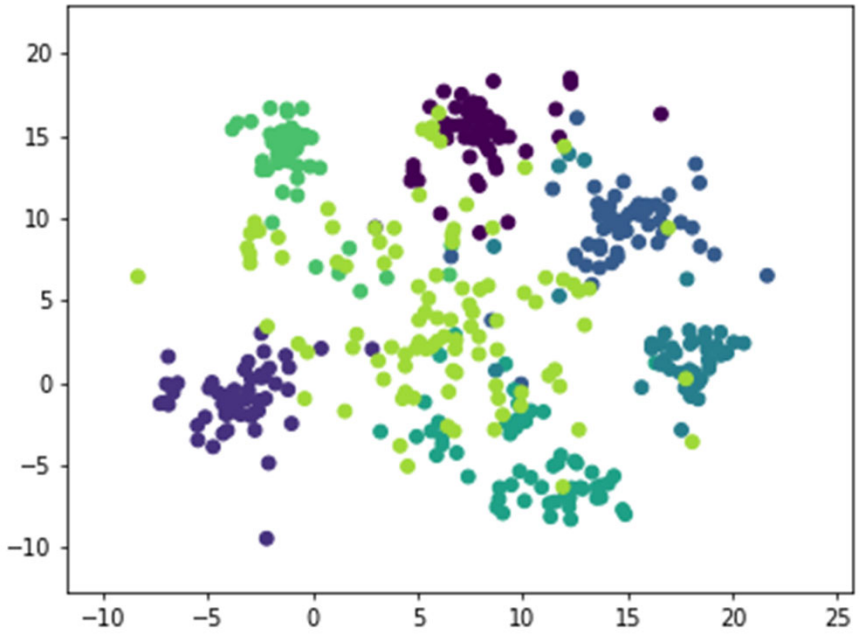
$$\hat{p}(y = k|x) = \frac{\exp \ell_k(z)}{\sum_{k'} \exp \ell_{k'}(z)}$$



Supervised Classification: UMAP Visualization

- DenseNet with 384-dimensional latent space.
- CIFAR-10: 6 known classes, 4 novel classes
- Light green: novel classes
- Darker greens: known classes

- Note that many novel classes stay toward the center of the space; others overlap with known classes
- Training was not required to “pull them out” so that they could be discriminated
- Distance to known classes is useful, but distances between novel points are not useful (e.g., for clustering to discover new classes)



Alex Guyer

Anomaly Scores $A(x)$

- Entropy of the predicted probabilities:
  - $A(x) = H(\hat{y}|x) = -\sum_k \hat{p}(y = k|x) \log \hat{p}(y = k|x)$
- (Lack of) confidence:
  - $A(x) = 1 - \max_k \hat{p}(y = k|x)$
- Maximum logit
  - $A(x) = -\max_k \ell_k(z)$
- In our experience, the max logit is slightly better than the other two

Headroom Analysis

Risheek Garrepalli (MS 2020)

- 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

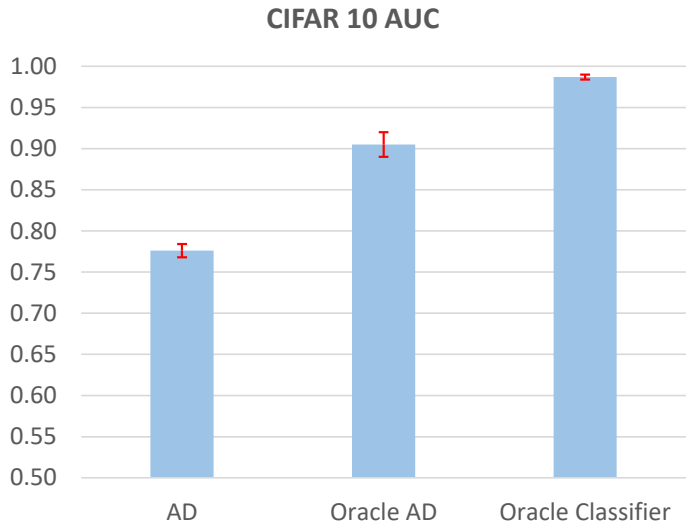
Methods:

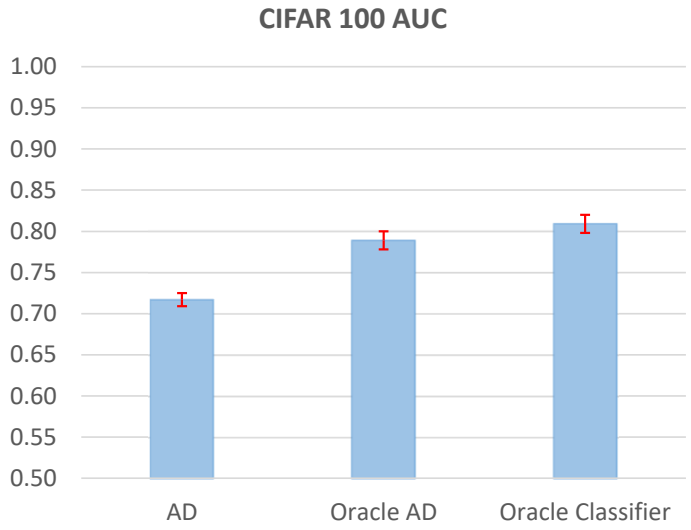
- CIFAR-10: 6 “nominal” classes and 4 “anomaly” classes
- CIFAR-100: 80 “nominal” classes and 20 “anomaly” classes

- 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 anomaly score on test set; measure AUC (“nominal” vs “anomaly” decision)
- Oracle Anomaly Detection
  - Take all validation data and label the nominal classes as “nominal” and the anomaly classes as “anomaly”
  - Train a random forest (1000 trees) that takes z as input and predicts “nominal” vs. “anomaly”
  - Compute test set anomaly scores using this classifier; measure AUC
- Oracle Representation
  - Train ResNet34 on all classes
  - Train a random forest (1000 trees) that takes z as input and predicts “nominal” vs. “anomaly”
  - Compute test set anomaly scores using this classifier; measure AUC

Results

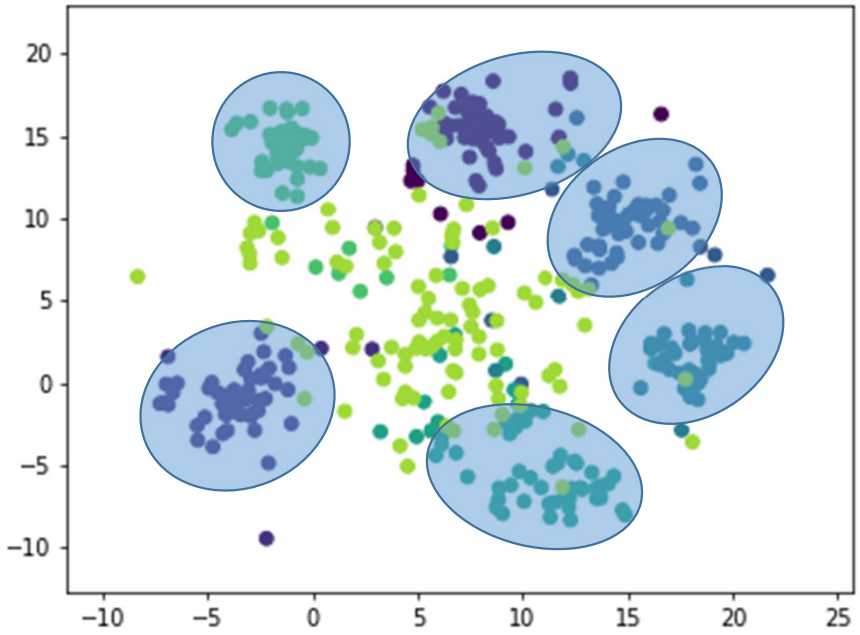
- Details:
  - Oracle Anomaly Detector: 1000-tree Random Forest
  - Anomaly Score: max logit
- Q1: The latent space contains much more anomaly information than is extracted by current anomaly scores
  - 0.776→0.905 = 0.129
  - 0.717→0.789 = 0.072
- Q2: There is additional anomaly information in the images that is not represented by the latent space
  - 0.905→0.987 = 0.082
  - 0.789→0.809 = 0.020





Extracting Improved Anomaly Scores via Density Estimation

- Fit a Gaussian to the z values
  - Let $Q_k(z; \mu_k, \Sigma_k)$ be the Gaussian fitted to class k
  - Anomaly score $\min_k -\log Q_k(z_q)$
  - where $z_q = E(x_q)$
- This is very practical and works surprisingly well
- Requires computing Σ_k^{-1}, which requires special tricks



Mahalanobis Method

(Lee, Lee, Lee & Shin 2018)

- Fit a shared Σ across all classes (after subtracting off μ_k for each class)
- This allows us to use the Mahalanobis distance rather than the density

$$A(x) = \min_k MD(z_q, \mu_k; \Sigma)$$

- Train on CIFAR-10
- Test on mix of CIFAR-10 and SVHN

Anomaly Score $A(x)$	AUC	TNR@95%TPR	Accuracy
$1 - \max_k \hat{p}(y = k \mid x_q)$	89.89	32.19	85.06
$\min_k MD(z_q, \mu_k; \Sigma)$	93.92	54.51	88.93

White lie warning: Lee et al. include several other improvements. This was the simplest of their methods.

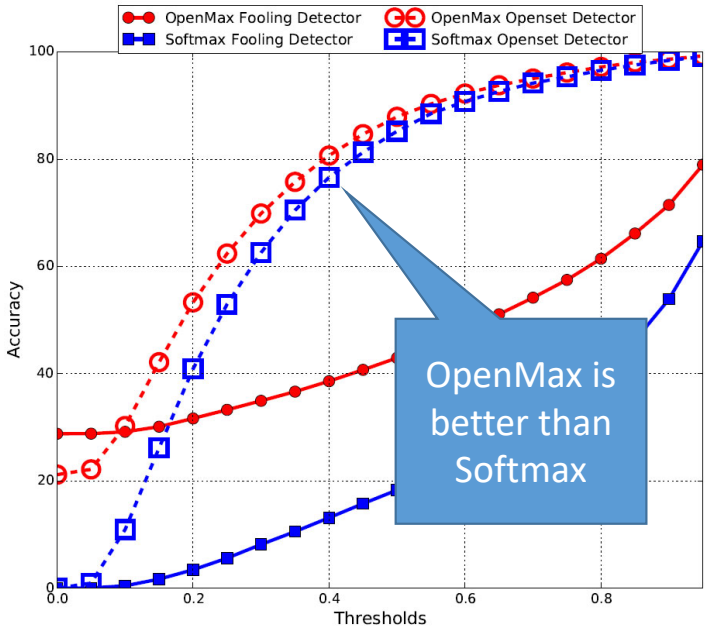
Extreme Value Distributions

[Bendale & Boult, CVPR 2016]

- Extreme value distribution
  - General model of the tails of probability distributions
  - Weibull distribution
- Goal: Shrink the logit score ℓ_k if the query x_q lies far from the mean logit score for class k
- Method
  - Let $\ell(z) = \left(\ell_1(z), \ldots, \ell_K(z)\right)$ be the vector of logits for $z = E(x)$
  - Let G_k be the set of correctly-classified training examples for class k
  - Compute the mean logit vector for each class k

$$\bar{\ell}_k = \frac{1}{N_k} \sum_{i \in G_k} \ell(z_i)$$

  - Fit the Weibull distribution to $\left\{\left\|\ell(z_i) - \bar{\ell}_k\right\| : i \in G_k\right\}$
  - Compute a weight $\omega_k(z_q)$ based on the CDF of the fitted Weibull distribution
  - $\tilde{\ell}_k(z_q) \coloneqq \omega_k(z_q) \cdot \ell_k(z_q)$
  - $\tilde{\ell}_0(z_q) \coloneqq \sum_k \ell_k(z_q) - \tilde{\ell}_k(z_q)$ collect the removed logit mass into an "open space" logit class 0
  - $\hat{p}(y = k | x_q) = \mathrm{softmax}\left(\tilde{\ell}_0(z_q), \tilde{\ell}_1(z_q), \ldots, \tilde{\ell}_K(z_q)\right)$
- This gives an explicit probability $\hat{p}(y = 0 | x_q)$ that x_q belongs to a novel class



Method 2:
Supervised Learning with a Modified Output Layer

Goal: Learn an improved representation

- Label Smoothing
- G-ODIN: Compute the logit as a ratio of two learned functions $\ell_k(z) \coloneqq h_k(z)/g(z)$. Then apply softmax
- $IsoMax_I$: Compute logits based on distance to learned prototypes for each class

Label Smoothing

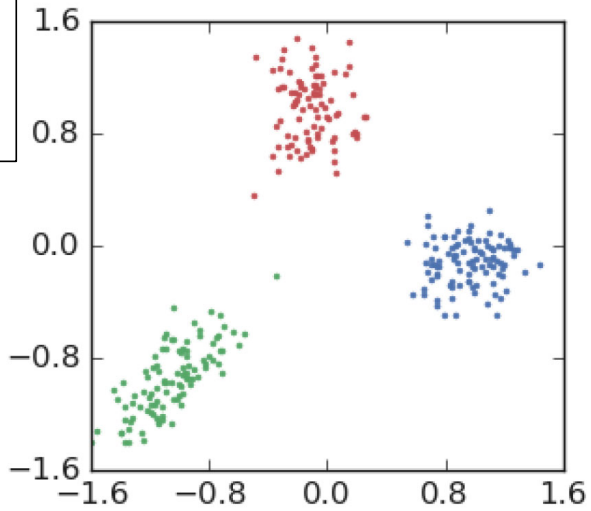
- Old idea from 1990s, reintroduced by Szegedy, et al. (CVPR 2015)
- Use a standard softmax output layer
- Change the usual 1-hot targets from

  $(0, \ldots, 1, \ldots, 0)$ to $\left(\frac{\alpha}{K}, \ldots, (1-\alpha)+\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right)$

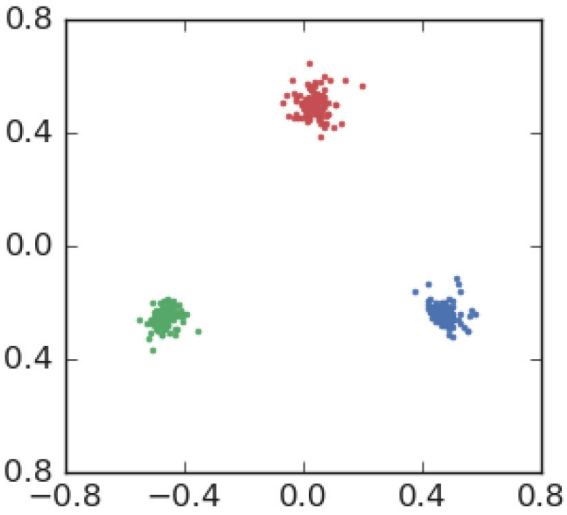
  Remove α probability mass from the target class and distribute it uniformly across all classes
- Müller, et al. (NeurIPS 2019) show that this causes the data points to form tighter clusters

CIFAR-100
3 semantically different classes

1-hot targets



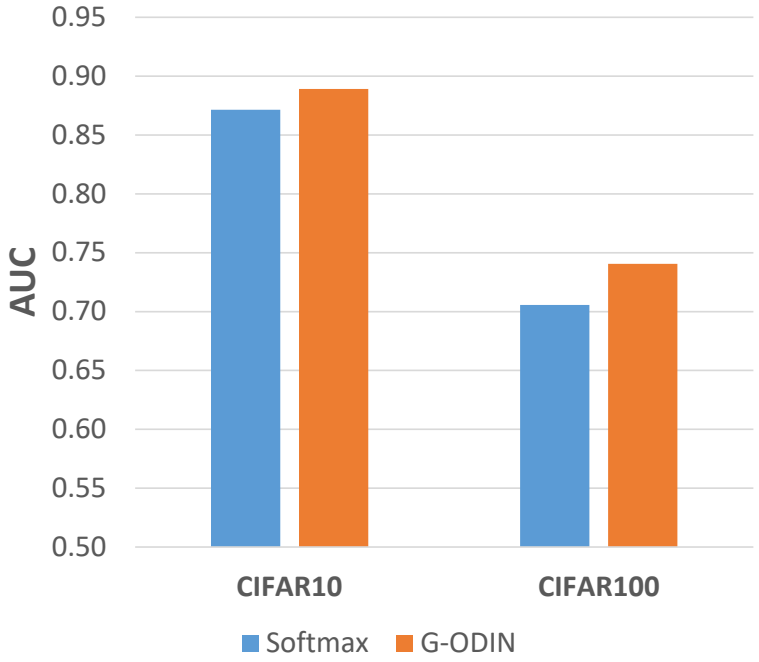
label smoothing



G-ODIN

(Hsu, Chen, Jin, & Kira, 2020)

- Compute the logit as a ratio of two learned functions $\ell_k(z) := h_k(z)/g(z)$. Then apply softmax
- Experiments
  - CIFAR-10: 6 known; 4 unknown
  - CIFAR-100: 80 known; 20 unknown
  - Modest improvements



(Alex Guyer, unpublished)

$IsoMax_I$

(Macedo & Ludermir, 2021)

- Learn a "prototype" μ_k for each class k in the latent space
  Normalize
  - the z values: $\tilde{z} \coloneqq z/\|z\|$ and
  - the prototypes: $\tilde{\mu}_k \coloneqq \mu_k/\|\mu_k\|$

$$\ell_k(z) \coloneqq -\kappa|c|\left\|\tilde{z}_q - \tilde{\mu}_k\right\|$$

  - c is a learned scaling parameter. $\kappa = 10$

- Anomaly score

$$A(x_q) = \min_k \left\|\tilde{z}_q - \tilde{\mu}_k\right\|$$

Anomaly Score $A(x)$	AUC	TNR@95%TPR
$1 - \max_k \hat{p}(y = k\|x_q)$	86.9	33.2
$IsoMax_I$ MDS	99.5	97.2

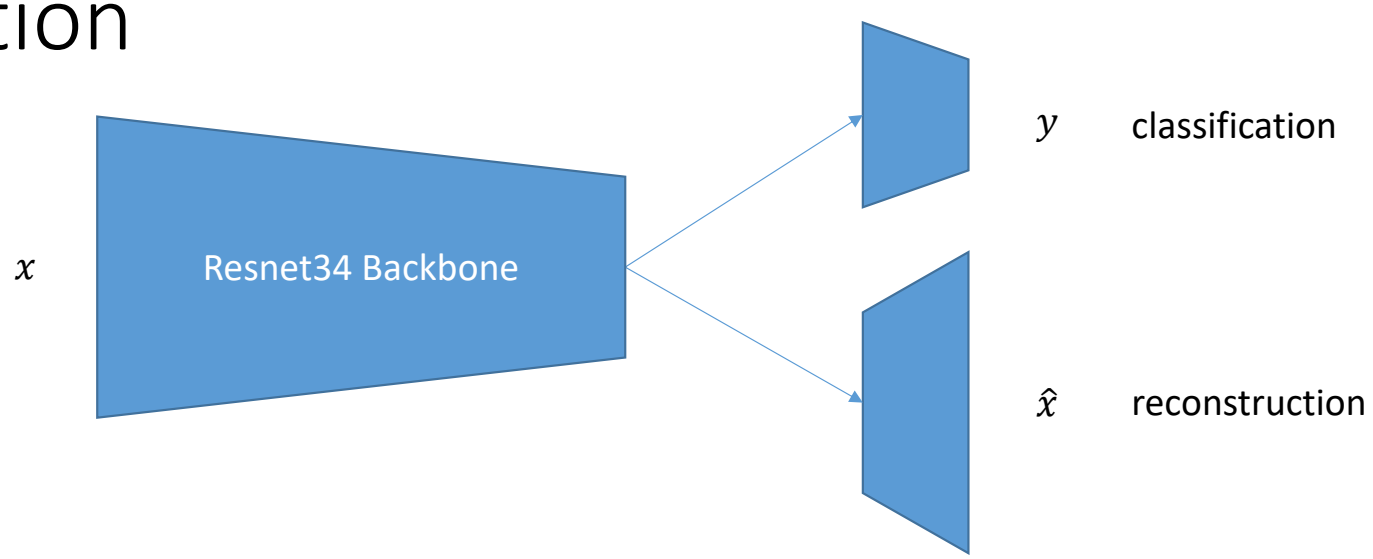
Other Methods

- Replace final layer with a Gaussian Process classifier
  - Liu, Lin, Padhy, et al. (NeurIPS 2020)
- Replace final layer with a Dirichlet distribution rather than a categorical (multinomial) distribution
  - Sensoy, Kaplan & Kanemir (NeurIPS 2018)
- Open question:
  - Compare the learned representations of these different methods

Method 3: Hybrid Methods

- Combine a supervised loss with an anomaly detection method
- Supervised + Autoencoder
- Supervised + Deep Density Estimator

Reconstruction

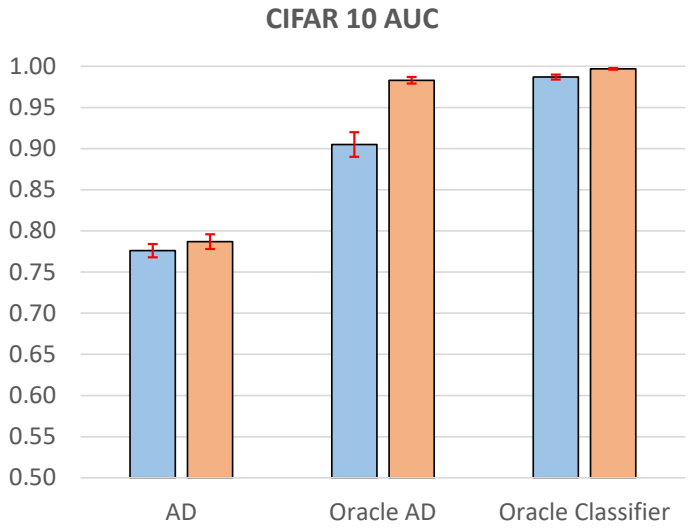


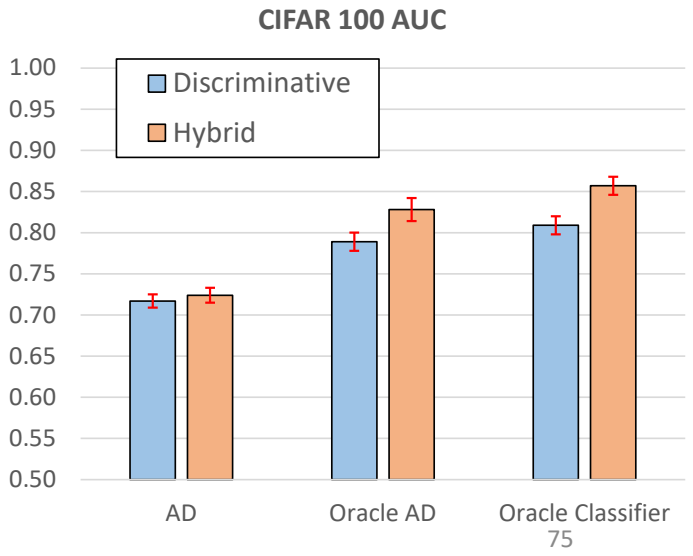
- Add a Reconstruction head to the network and jointly train the representation to support both classification and reconstruction (per-pixel squared error)
- Loss = Cross-Entropy + $\lambda \times$ Reconstruction Error
- CIFAR10: $\lambda = 0.9$, CIFAR100: $\lambda = 0.005$
- See also:
  - Oza, P., & Patel, V. M. C2AE: Class Conditioned Auto-Encoder for Open-set Recognition. CVPR 2019
  - Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., & Lakshminarayanan, B. (2019). Hybrid models with deep and invertible features. ICML 2019
  - Perera, P., Morariu, V. I., Jain, R., Manjunatha, V., Wigington, C., Ordonez, V., & Patel, V. M. Generative-discriminative feature representations for open-set recognition. CVPR 2020

Reconstruction Results

(Garrepalli, 2020 MS)

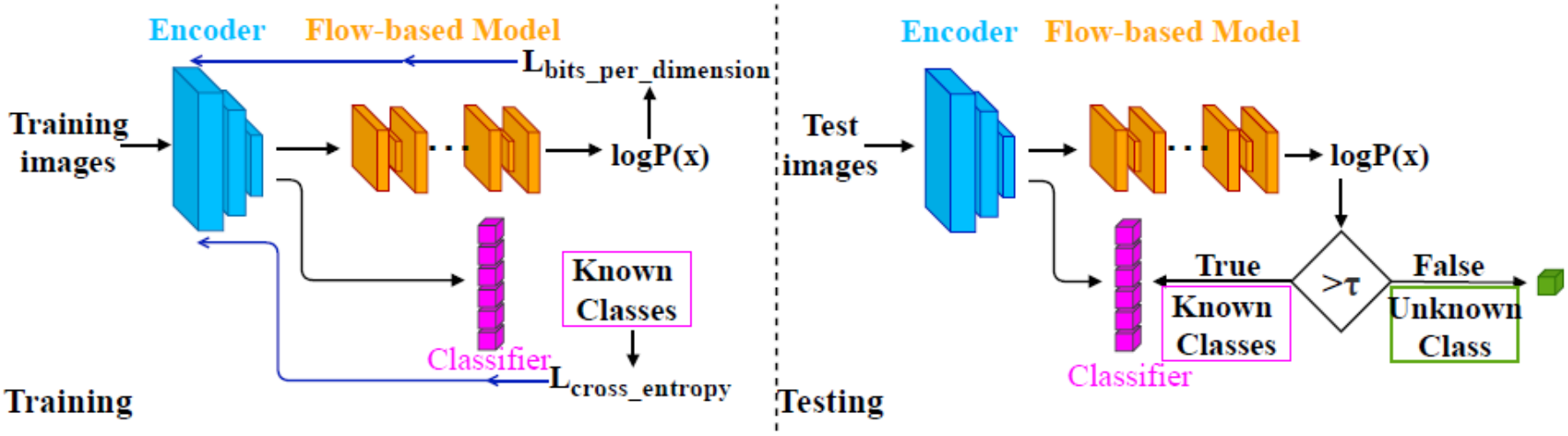
- Result: Hybrid representation improves performance
- Caution: λ tuned using labeled test data





Open Hybrid: Classification + Density Estimation

(Tack, Li, Guo, Guo, 2020)



- Residual Flow Deep Density Estimator
  - (Chen, Behrmann, Duvenaud, et al. NeurIPS 2019)
- Standard Cross-Entropy Supervised Loss
  - Claim: This helps focus $P(x)$ on relevant aspects of the images
- Anomaly Score: $A(x_q) = -\log P(x_q)$

OpenHybrid Results

- 6 Known and 4 Unknown classes

AUC	MNIST	SVHN	CIFAR-10
$1 - \max_k \hat{p}(y = k \mid x_q)$	0.978	0.886	0.677
OpenHybrid: $-\log P(x_q)$	0.995	0.947	0.883

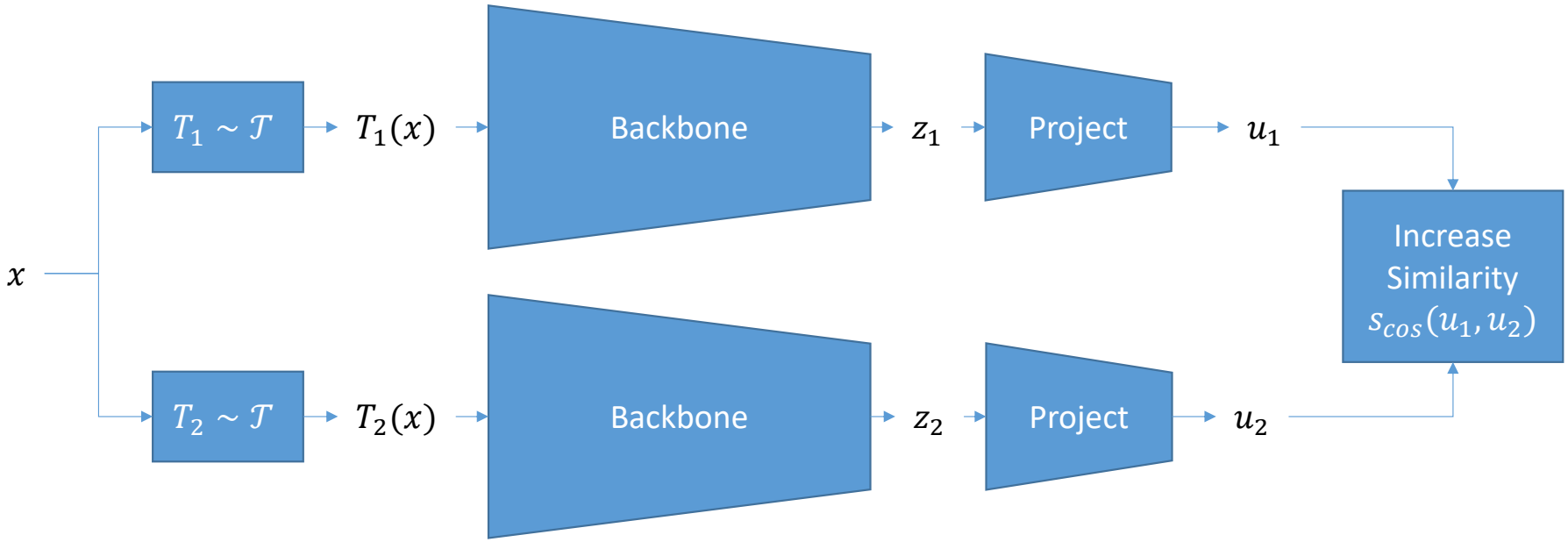
- 4 Known and many unknown classes drawn from CIFAR-100

AUC	CIFAR+10	CIFAR+50
$1 - \max_k \hat{p}(y = k \mid x_q)$	0.816	0.805
OpenHybrid: $-\log P(x_q)$	0.962	0.955

Method 4: Instance-Contrastive Learning

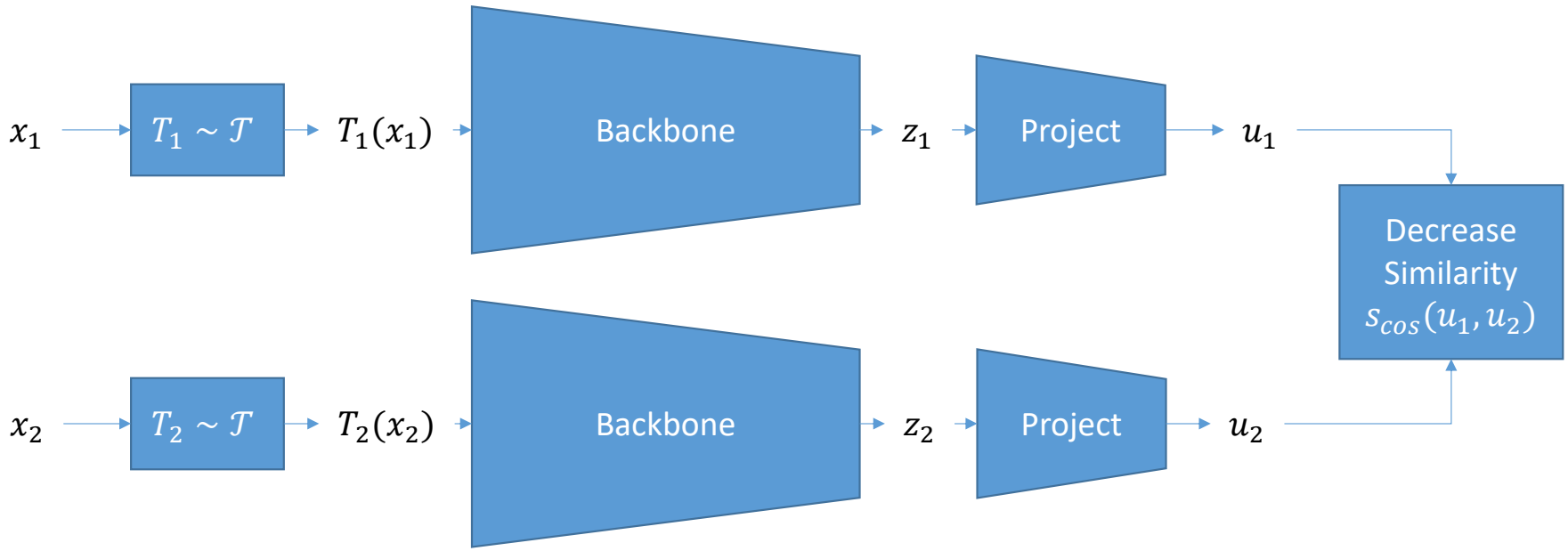
- SimCLR (Chen, et al. 2020)
- Set of transformations $\mathcal{T}$
- Case 1: Same image

cosine similarity: $s_{cos}(u_1, u_2) = \frac{u_1 \cdot u_2}{\|u_1\|\|u_2\|}$



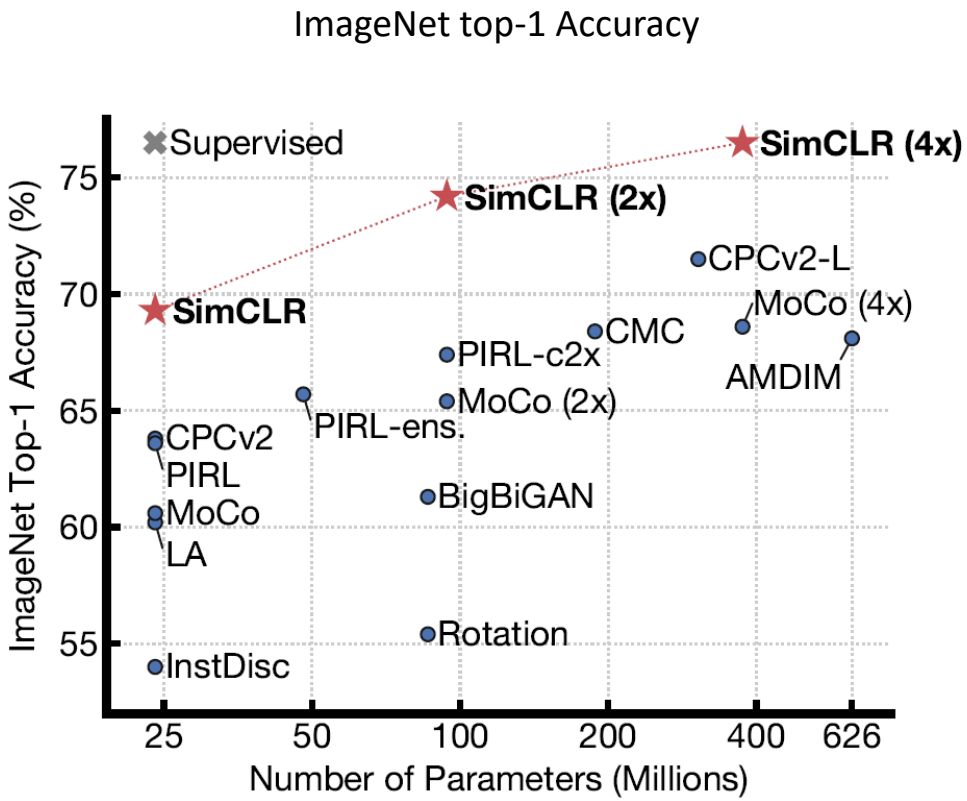
Method 4: Instance-Contrastive Learning

- Case 2: Different images



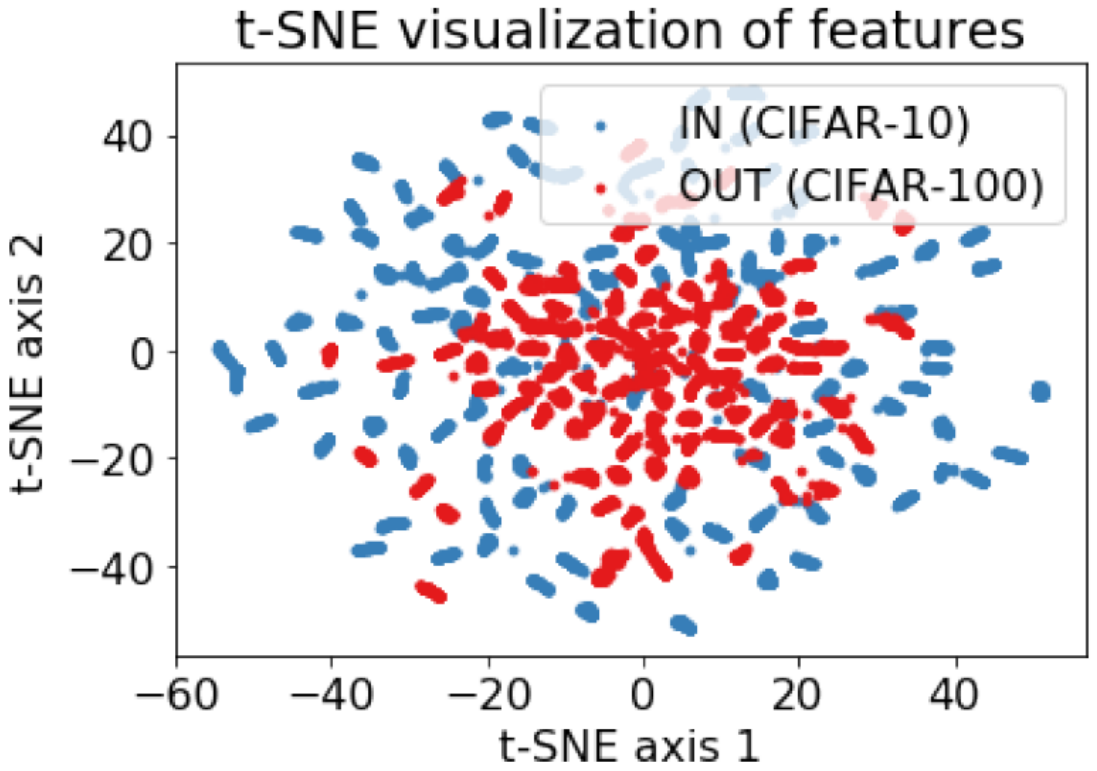
Classification

- Classification
  - Train linear logits on z using labeled training data and softmax



Visualization of Instance-Contrastive Representations

- As with supervised learning, we see that anomalies tend to cluster in the center of the Z space
- 100 images from CIFAR-10 (training) and CIFAR-100 (novel categories)
- Each instance was augmented 100 times



 Tack, et al. NeurIPS 2020

Multivariate Gaussian Scoring

- Winkins, Bunel, Roy, et al. 2020
- Train with two losses
  - SimCLR instance-contrastive loss
  - Softmax with label smoothing
- Fit multivariate Gaussians G_k to each class (μ_k, Σ_k)
- Anomaly score $\min_k -\log G_k(z_q)$

Known: CIFAR-10; Unknown: CIFAR-100

Method	AUC
$1 - \max_k \hat{p}(y = k \mid x_q)$	86.4
$\min_k G_k(x_q)$	92.9

Known: CIFAR-10; Unknown: CIFAR-100

Method	AUC	FPR@95%TPR	Accuracy
$\min_k -\log G_k(x_q)$ No contrastive training No label smoothing	81.3	67.1	73.8
$\min_k -\log G_k(x_q)$	92.9	39.9	85.9

CSI

(Tack, Mo, Jeong, Shin, NeurIPS 2020)

- Two kinds of transformations
  - $\mathcal{T}$: class-preserving transformations: Cutout, Edge image, Salt+Pepper Noise, Blur
  - $\mathcal{S}$: class-damaging transformations: Permutation, Rotation

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

- Maximize $s_{cos}\left(T_1(x), T_2(x)\right)$ and Minimize $s_{cos}\left(T_1(x), S_1(x)\right)$

Auxiliary Task and Anomaly Scores

- Given $z = E\big(T(x)\big)$, predict which transformation T was applied
- Confidence Calibration
- Anomaly Score combines many factors
  a. $\|z_q\|$
  b. $\max_i s_{cos}(x_q, x_i)$ most similar training data point x_i
  c. Expected value of b. over all $S \in \mathcal{S}$
  d. Expected value of b. over all $T \in \mathcal{T}$
  e. Accuracy of the predictions of the auxiliary classifier over all $S \in \mathcal{S}$
  f. (Somewhat ad hoc; probably reflects tuning to optimize results)

CSI Results combining unsupervised and supervised training

Train on CIFAR-10; Evaluate on CIFAR-10 ∪ Other Dataset

AUC	SVHN	CIFAR100	LSUN*	ImageNet*
$1 - \max_k \hat{p}(y = k \mid x_q)$	88.6	85.8	87.5	87.4
CSI	96.5	90.5	93.5	94.0

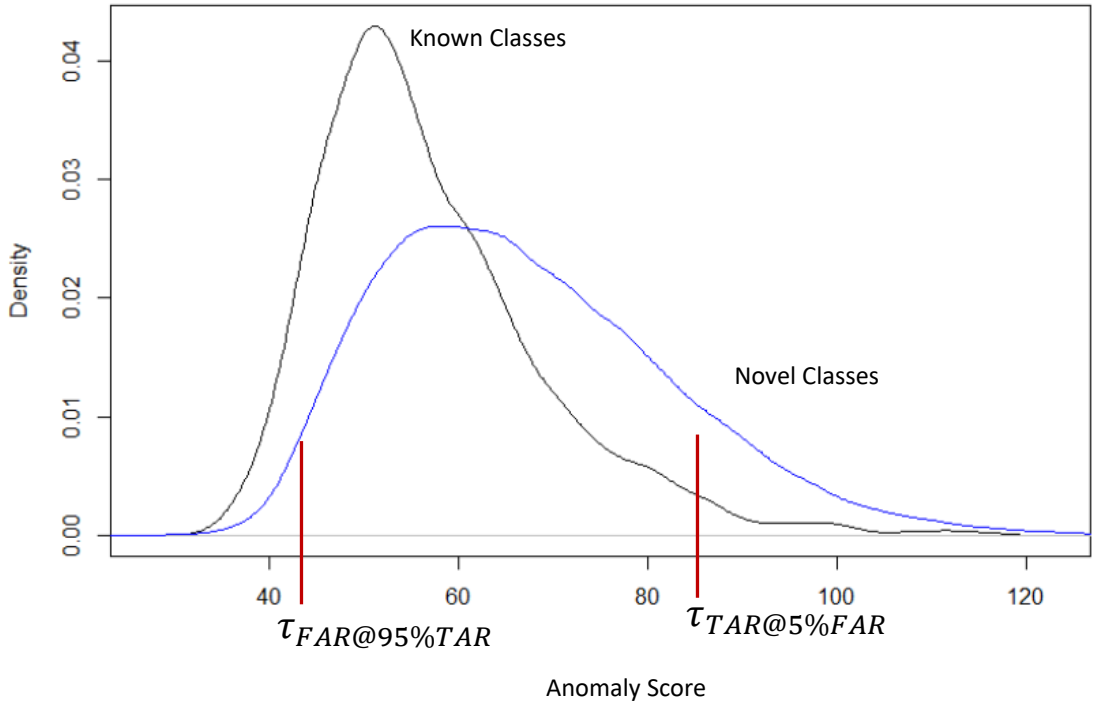
*Fixes problem with buggy resize operation to remove easy images

Outline

- Theoretical Approaches to Anomaly Detection
- Practical Algorithms for Hand-Crafted Features
- Deep Anomaly Detection
- Setting the Anomaly Detection Threshold

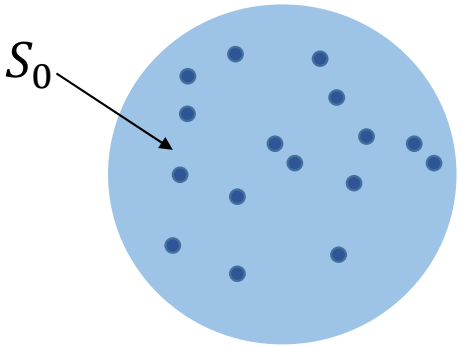
Setting the Anomaly Score Threshold

- Maximize TAR@5%FAR
  - Set $\tau_{TAR@5\%FAR}$ to the 95% quantile of the Known Classes distribution
  - Only requires the Known Classes density, which can be computed on a validation set
- Minimize FAR@95%TAR
  - Set $\tau_{FAR@95\%TAR}$ to the 5% quantile of the Novel Classes distribution
  - We need to estimate this distribution

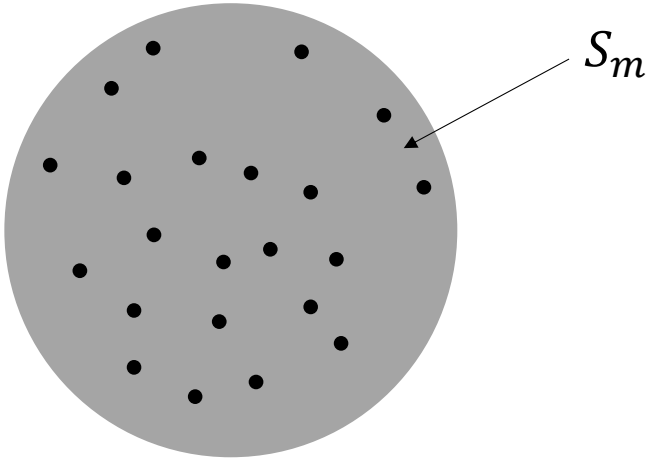


Another Resource: Unlabeled Data

Nominal Distribution



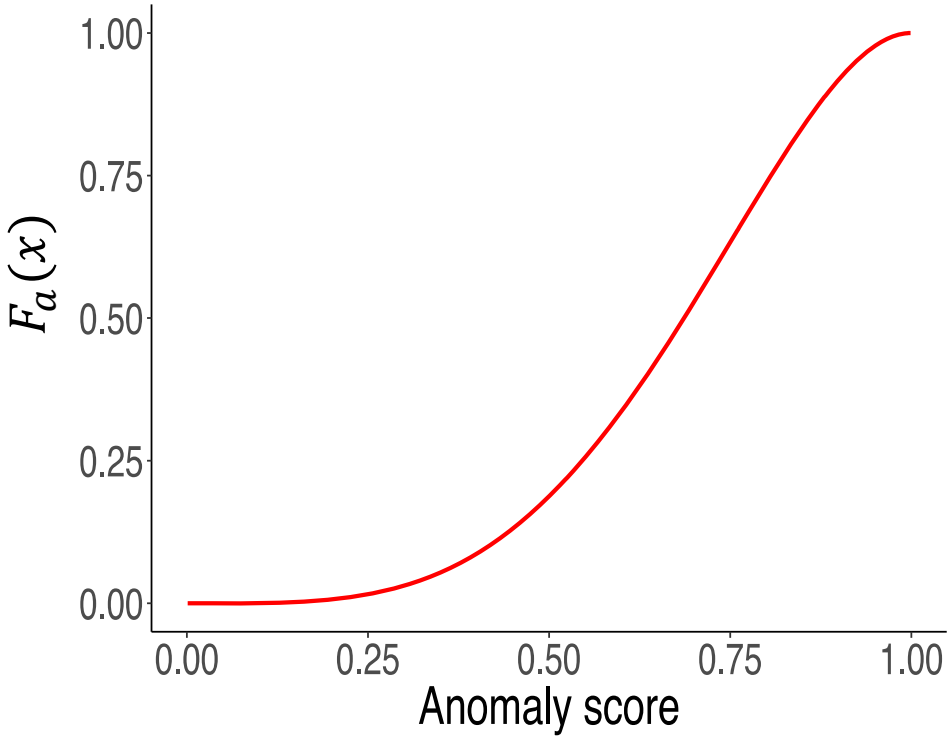
Mixture Distribution



Proportion of Aliens = α

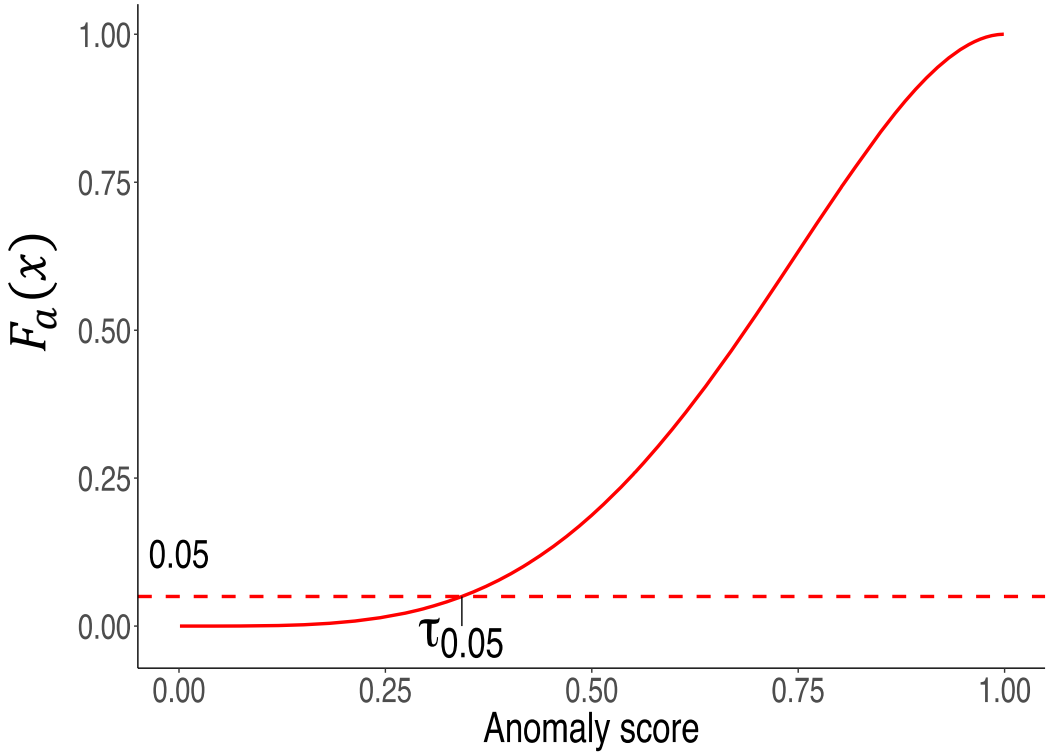
$$P_m = (1-\alpha)P_0 + \alpha P_a$$

Cumulative CDF of Alien Anomaly Scores: F_a



Want to have
TAR $= 1 - q$

Choosing τ for target quantile q



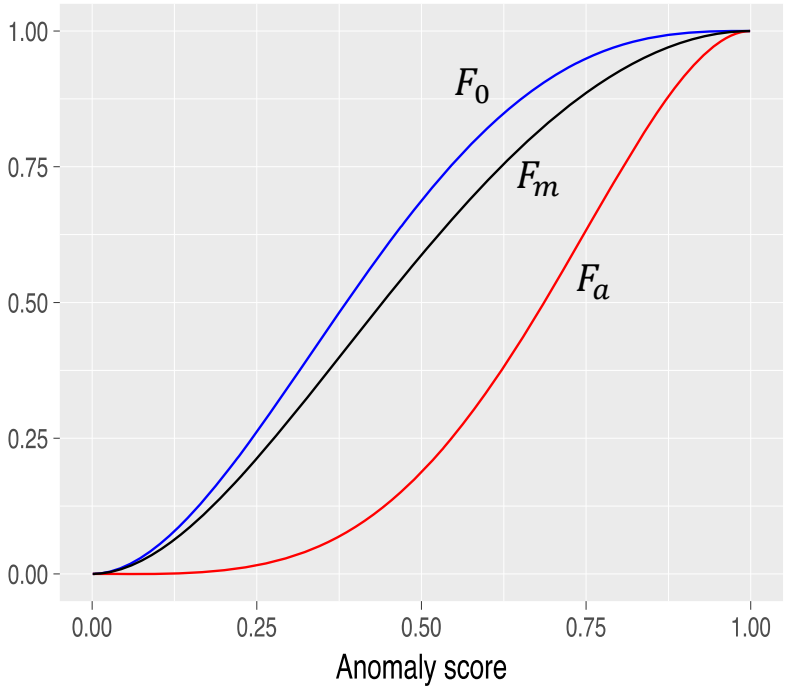
$$q = 0.05$$

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

implies that

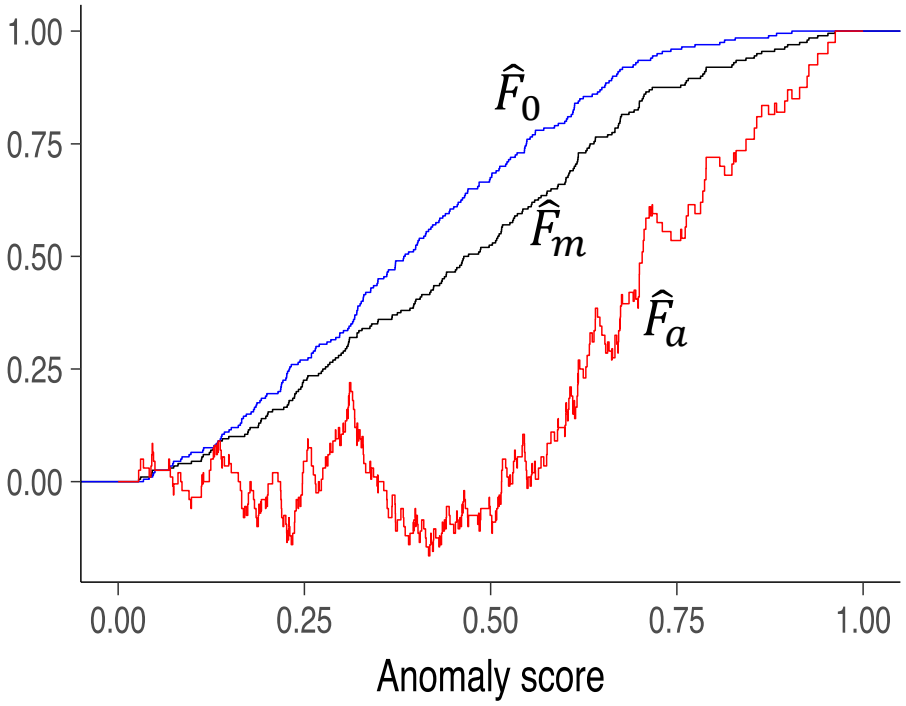
$$F_m(x) = (1 - \alpha)F_0(x) + \alpha F_a(x)$$

CDFs of Nominal, Mixture, and Alien Anomaly Scores



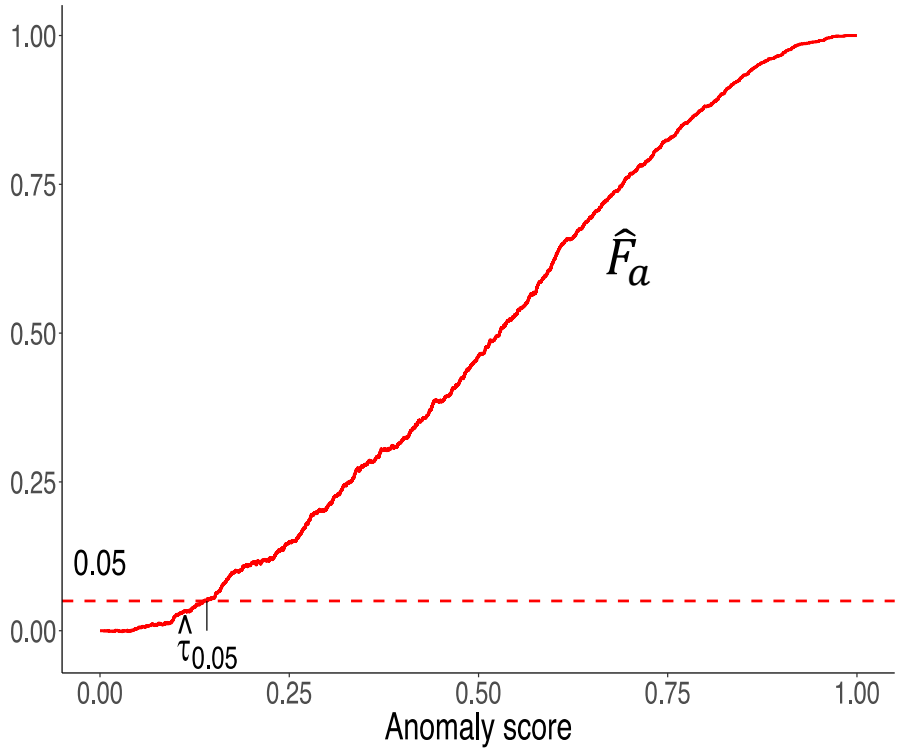
$$F_a(x) = \frac{F_m(x) - (1-\alpha)F_0(x)}{\alpha}$$

What We Have Are Empirical CDFs



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

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



Theoretical Guarantee

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

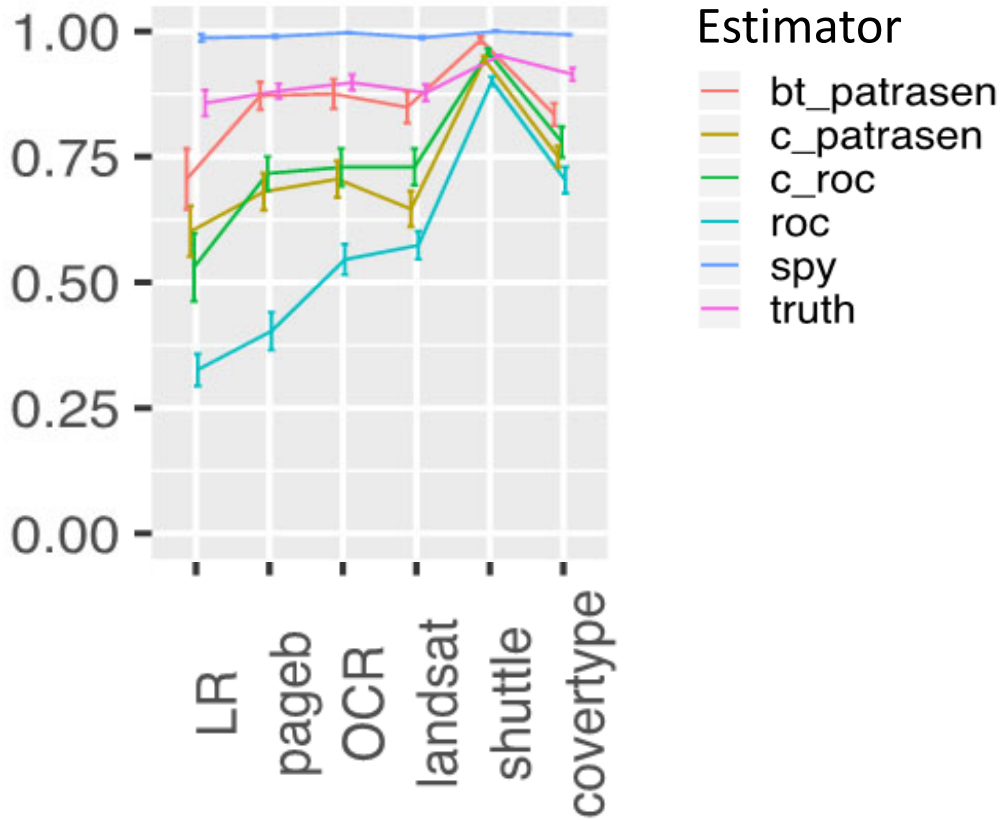
- Theorem: If

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

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

Estimating the mixing proportion α

- The mixing proportion is not identifiable in general
- However, under reasonable assumptions, we can obtain an estimate α_0 guaranteed with high probability to be a lower bound on α
- Comparison of five estimators
  - bt_patrasen comes closest to achieving the target TPR of 0.95 on six datasets
  - (Patra & Sen, 2016)
- Liu, Mondal, Dietterich (under review)



Summary

Main Anomaly Detection Methods

- Density estimation
- Density quantile estimation
- Distance
- Reconstruction

Representation Learning Methods

- Supervised classification with softmax
- Supervised classification with a different final layer
- Hybrid network: supervised classification + anomaly detection loss
- Instance-contrastive learning

Open Questions / Research Needs

- Improved methods for representation learning
  - meaningful representation of distances between novel-class points
- Methods for comparing learned representations
- Open framework for controlled comparison of methods
- Methods for
  - Explaining anomalies to users
  - Incorporating user feedback to improve anomaly detection
  - Discovering and incorporating new classes into the classifier

Citations

- Bendale, A., & Boult, T. (2016). Towards Open Set Deep Networks. In CVPR 2016 (pp. 1563–1572). http://doi.org/10.1109/CVPR.2016.173
- Chen, R. T. Q., Behrmann, J., Duvenaud, D., & Jacobsen, J.-H. (2019). Residual Flows for Invertible Generative Modeling. *ArXiv*, *1906.02735*(v1), 1–18. http://arxiv.org/abs/1906.02735
- Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. *ArXiv*, *2002.05709*(v2). http://arxiv.org/abs/2002.05709
- Hassen, M., Chan, P., (2018). Learning a Neural-network-based Representation for Open Set Recognition. arXiv 1802.04365.
- Hsu, Y.-C., Shen, Y., Jin, H., & Kira, Z. (2020). Generalized ODIN: Detecting Out-of-distribution Image without Learning from Out-of-distribution Data. ArXiv, 2002.11297(v1). http://arxiv.org/abs/2002.11297
- Lan, C. Le, & Dinh, L. (2020). Perfect density models cannot guarantee anomaly detection. *ArXiv*, *2012.03808*(v1), 1–15. http://arxiv.org/abs/2012.03808
- Lee, K., Lee, K., Lee, H., & Shin, J. (2018). A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, NeurIPS2018, 7167–7177. http://arXiv.org/abs/1807.03888
- Liang, S., Li, Y., Srikant, R. (2018). Enhancing the Reliability of Out-of-distribution Image Detection in Neural Networks. *ICLR 2018.*
- Liu, J. Z., Lin, Z., Padhy, S., Tran, D., Bedrax-Weiss, T., & Lakshminarayanan, B. (2020). Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness. *ArXiv*, *2006.10108*(v2). http://arxiv.org/abs/2006.10108
- 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. http://proceedings.mlr.press/v80/liu18e.html
- Macêdo, D., & Ludermir, T. (2020). Neural Networks Out-of-Distribution Detection: Hyperparameter-Free Isotropic Maximization Loss, The Principle of Maximum Entropy, Cold Training, and Branched Inferences. *ArXiv*, *2006.04005*(v2). http://arxiv.org/abs/2006.04005

Citations (2)

- Macêdo, D., & Ludermir, T. (2021). Improving Entropic Out-of-Distribution Detection using Isometric Distances and the Minimum Distance Score. *ArXiv*, *2105.14399*(v3), 1–11. http://arxiv.org/abs/2105.14399
- Mendes-Júnior, P., de Souza, R.,Werneck, R., Stein, B. V., Pazinato, D. V., de Almeida, W. R. (2017). Nearest neighbors distance ratio open-set classifier. *Machine Learning* 106: 359–386.
- Müller, R., Kornblith, S., & Hinton, G. (2019). When does label smoothing help? *Advances in Neural Information Processing Systems*, *32* (NeurIPS 2019).
- Neal, L., Olson, M., Fern, A., Wong, W-K., Li, F. (2018). Open Set Learning with Counterfactual Images. Proceedings of the European Conference on Computer Vision (ECCV 2018).
- Papamakarios, G., Pavlakou, T., & Murray, I. (2017). Masked Autoregressive Flow for Density Estimation. *NIPS 2017*. http://arxiv.org/abs/1705.07057
- Patra, R. K., & Sen, B. (2016). Estimation of a two-component mixture model with applications to multiple testing. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, *78*(4), 869–893.
- Sensoy, M., Kaplan, L., & Kandemir, M. (2018). Evidential deep learning to quantify classification uncertainty. *Advances in Neural Information Processing Systems*, *2018-December*(Nips), 3179–3189.
- Shafaei, A., Schmidt, M., & Little, J. (2018). Does Your Model Know the Digit 6 Is Not a Cat? A Less Biased Evaluation of "Outlier" Detectors. arXiv 1809.04729
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 2818–2826. https://doi.org/10.1109/CVPR.2016.308
- 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)*.
- 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*.