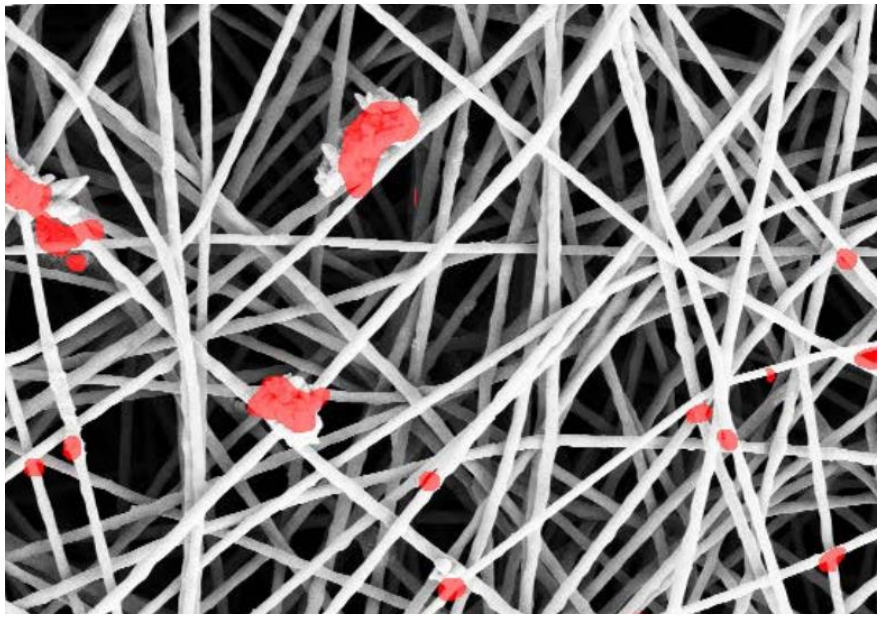
ECE 4252/8803: Fundamentals of Machine Learning (FunML) Fall 2024

Lecture 24: Anomaly Detection



OLIVES @GeorgiaTech

Georgia Tech

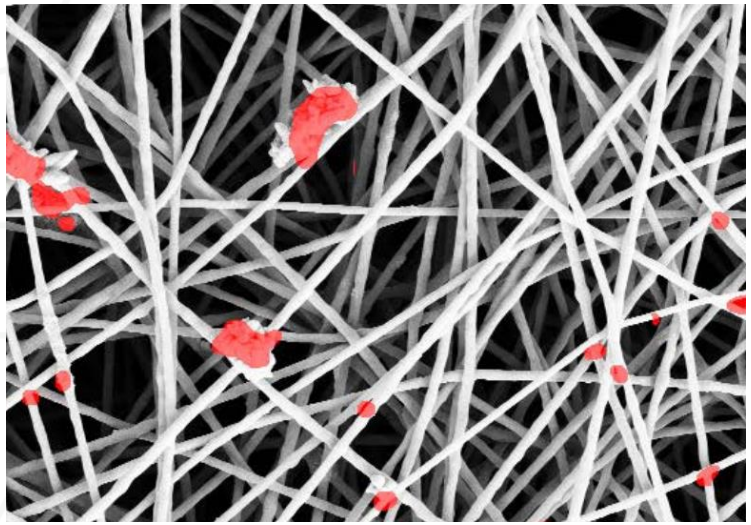
Overview

In this Lecture..

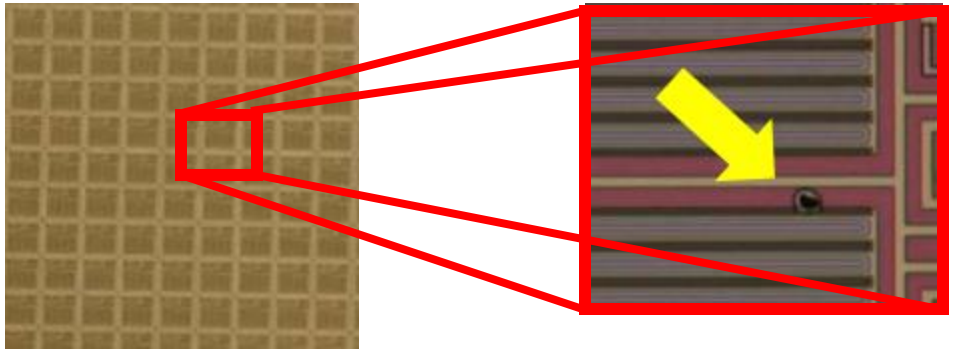
- Anomaly Definition
- Problem Setup
- Performance Metrics
- Anomaly Detection Settings
- Statistical Methods
- Reconstruction Methods
- GradCON

Anomaly Detection

Definition

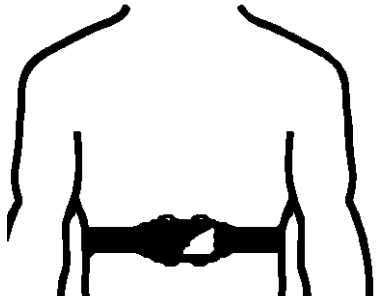


1

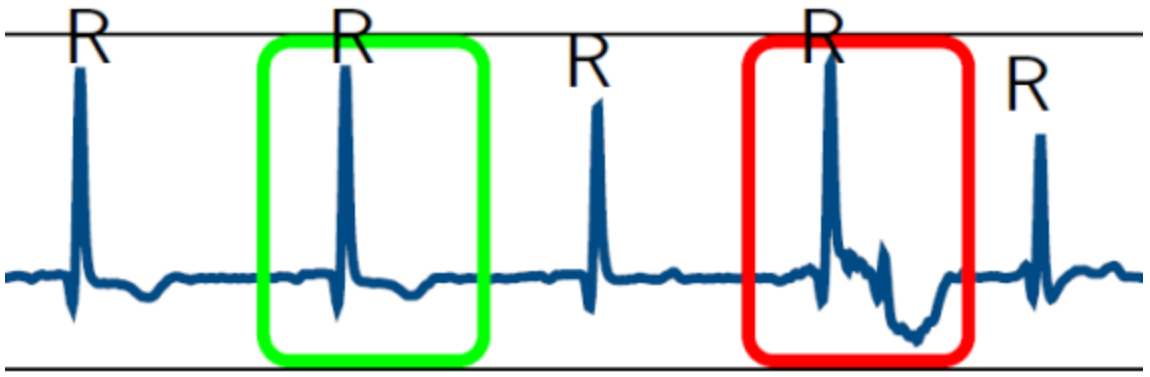


2

Anomaly Detection: Finding Rare Events in Normal Patterns



4



3

Anomaly Detection

Definition

"Anomalies are patterns in data that do not conform to a well defined notion of normal behavior"[1]

- Normal data are generated from a stationary process P_N
- Anomalies are generated from a different process $P_A \neq P_N$

- Examples:
  - Frauds in the stream of all the credit card transactions
  - Arrhythmias in ECG tracings
  - Defective regions in an image, which do not conform a reference pattern

- Anomalies might appear as spurious elements, and are typically the most informative samples in the stream

[1] V. Chandola, A. Banerjee, V. Kumar. "Anomaly detection: A survey". ACM Comput. Surv. 41, 3, Article 15 (July 2009), 58 pages

OLIVES @GeorgiaTech

Georgia Tech

Overview

In this Lecture..

Anomaly Definition

Problem Setup

- Anomaly Detection in Images
- Anomaly Detection in a statistical framework
- Common solution template

Performance Metrics

Anomaly Detection Settings

Statistical Methods

Reconstruction Methods

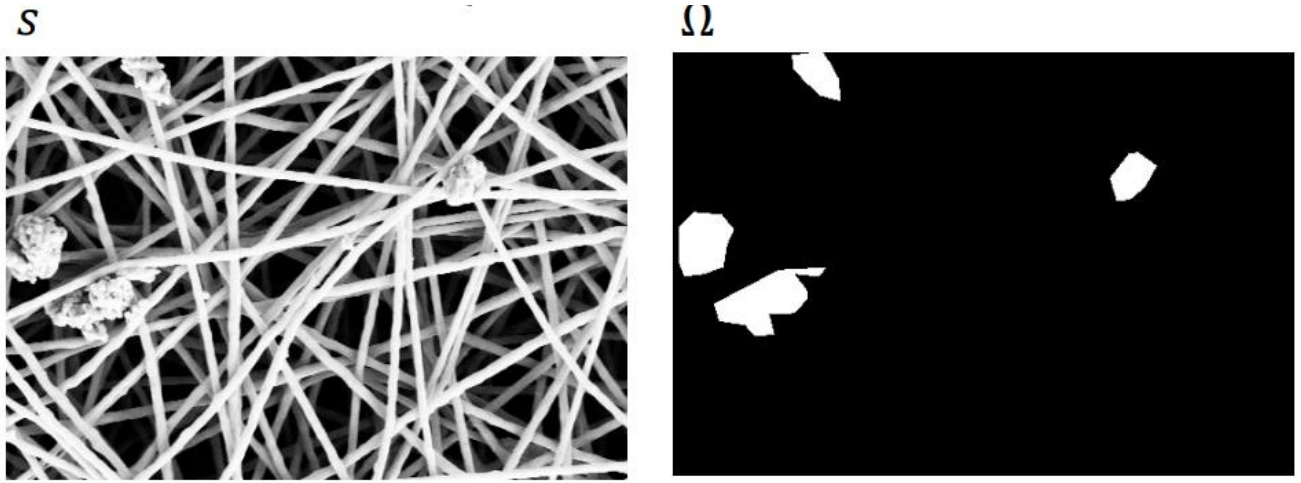
GradCON

Problem Formulation

Anomaly Detection in Images

- Let s be an image defined over the pixel domain $x \in \mathbb{Z}^2$
- Let $c \in x$ be a pixel and $s(c)$ the corresponding intensity
- Our goal is to locate any anomalous region in s, i.e., estimating the unknown anomaly mask Ω defined as

$$\Omega(c) = \begin{cases} 0 & \text{if } c \text{ falls in a normal region} \\ 1 & \text{if } c \text{ falls in an anomalous region} \end{cases}$$



Problem Formulation

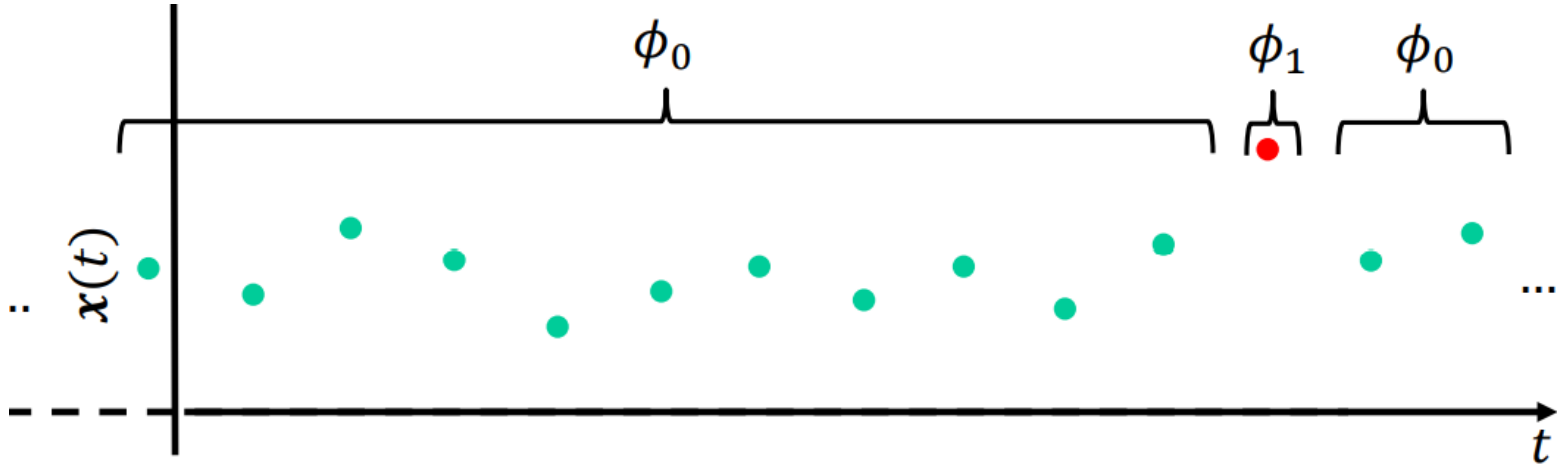
Anomaly Detection in a Statistical Framework

Monitor a set of data (not necessarily a stream)

$$\{x(t), t = t_0, \dots\}, x(t) \in \mathbb{R}^d$$

where $x(t)$ are realizations of a random variable having pdf ϕ_0, and detect outliers i.e., those points that do not conform with ϕ_0.

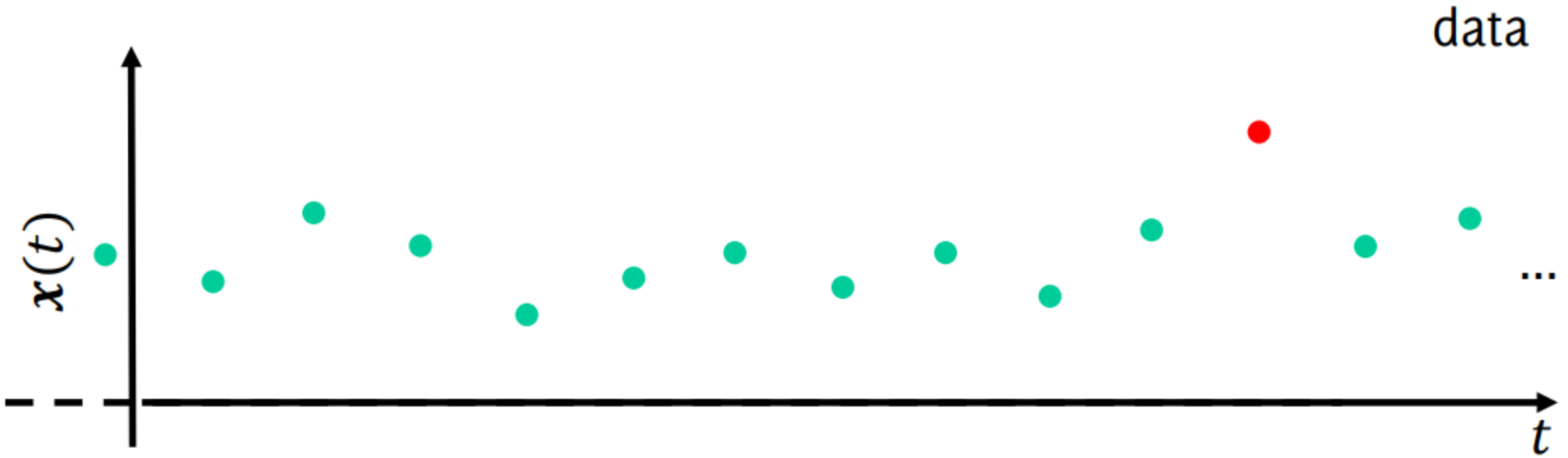
$$x(t) = \begin{cases} \phi_0 & \text{Normal data} \\ \phi_1 \end{cases}$$



Problem Formulation

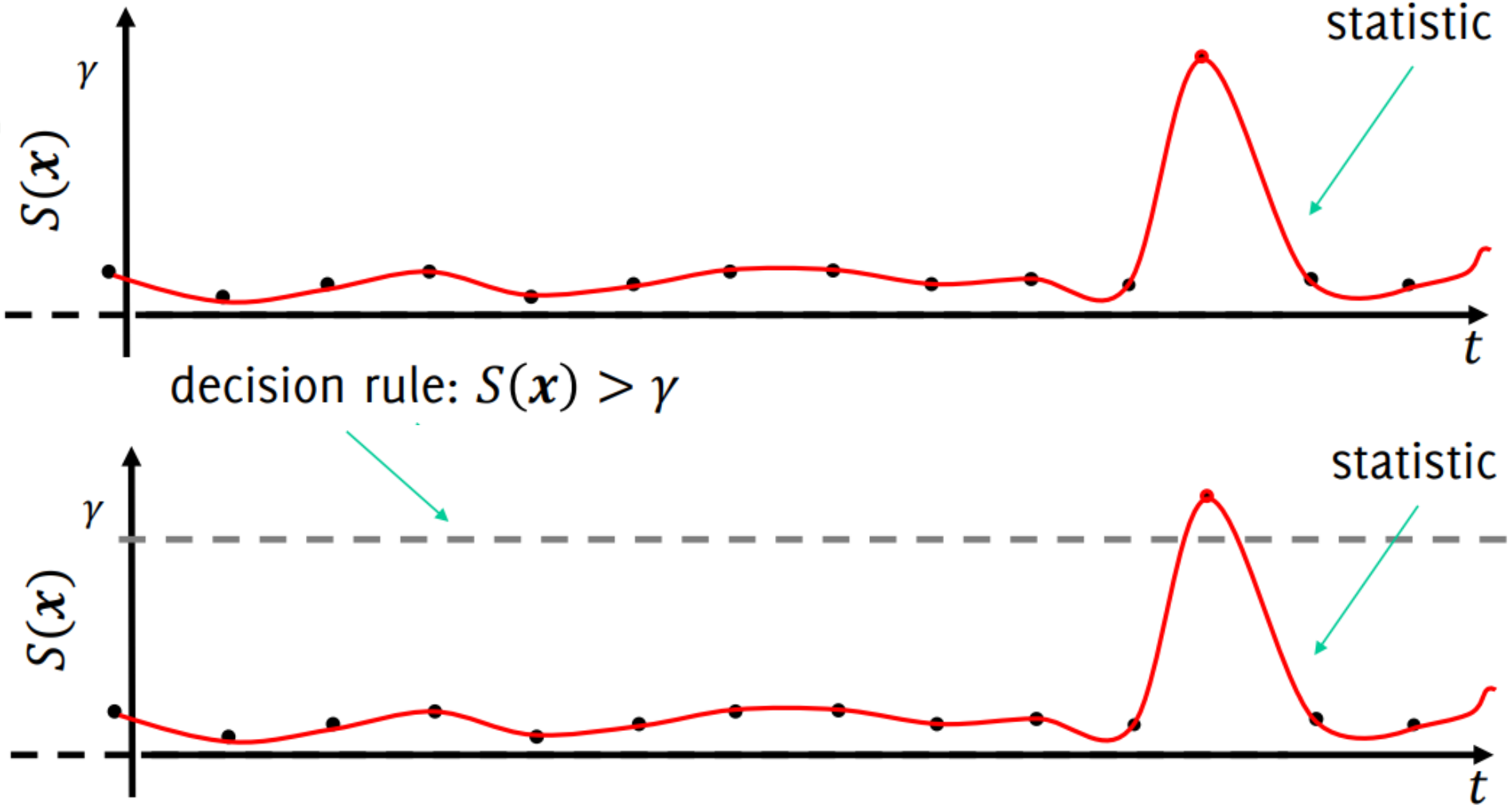
Anomaly Detection in a Statistical Framework

- Most algorithms are composed of:
  - A **statistic** that has a known response to normal data (e.g., the average, the sample variance, the log-likelihood, the confidence of a classifier, an “anomaly score”…)
  - A **decision rule** to analyze the statistic (e.g., an adaptive threshold, a confidence region)



Problem Formulation

Anomaly Detection in a Statistical Framework



Overview

In this Lecture..

- Anomaly Definition
- Problem Setup
- Performance Metrics
  - TPR/FPR
  - TPR-FPR Tradeoff
  - AUC
- Anomaly Detection Settings
- Statistical Methods
- Reconstruction Methods
- GradCON

Performance Measures

TPR/FPR

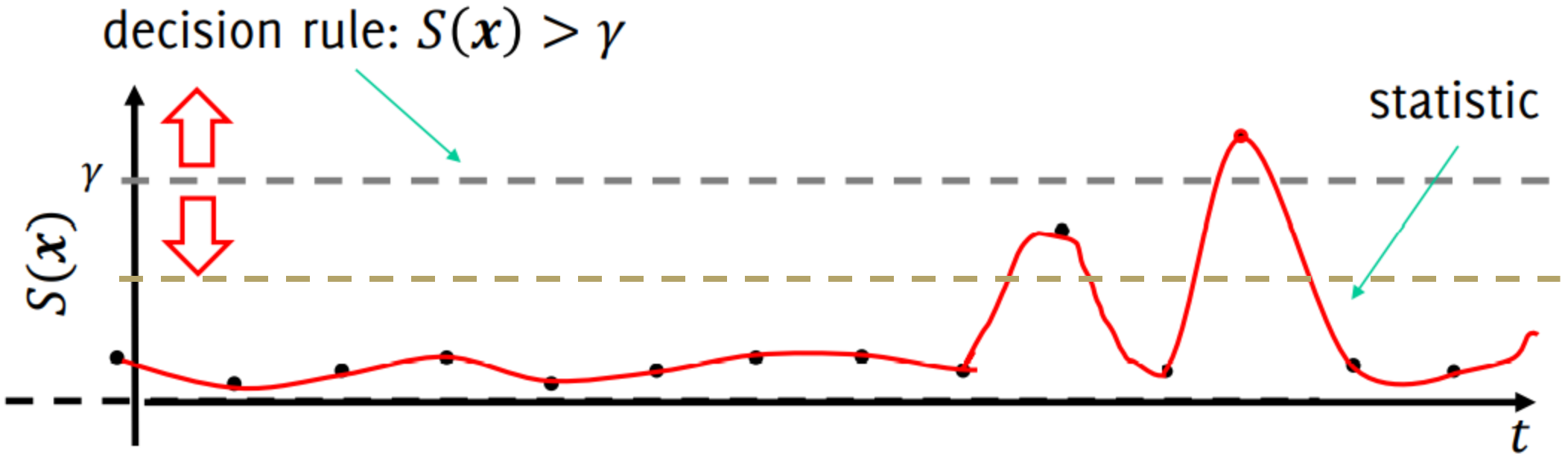
- True Positive Rate, TPR $= \frac{\#\{\text{anomalies detected}\}}{\#\{\text{anomalies}\}}$
- False Positive Rate, FPR $= \frac{\#\{\text{normal samples detected}\}}{\#\{\text{normal samples}\}}$

- You have already seen
  - False negative rate (or miss-rate): FNR = 1 − TPR
  - True negative rate (or specificity): TNR = 1 − FPR
  - Precision on anomalies: $\frac{\#\,\text{anomalies detected}}{\#\,\text{detections}}$
  - Recall on anomalies (or sensitivity, hit-rate): TPR

Performance Measures

TPR/FPR Tradeoff

- There is always a trade-off between TPR and FPR (and similarly for derived quantities), which is ruled by algorithm parameters
- By changing γ, performance changes (e.g. true positive increases but also false positives do



Performance Measures

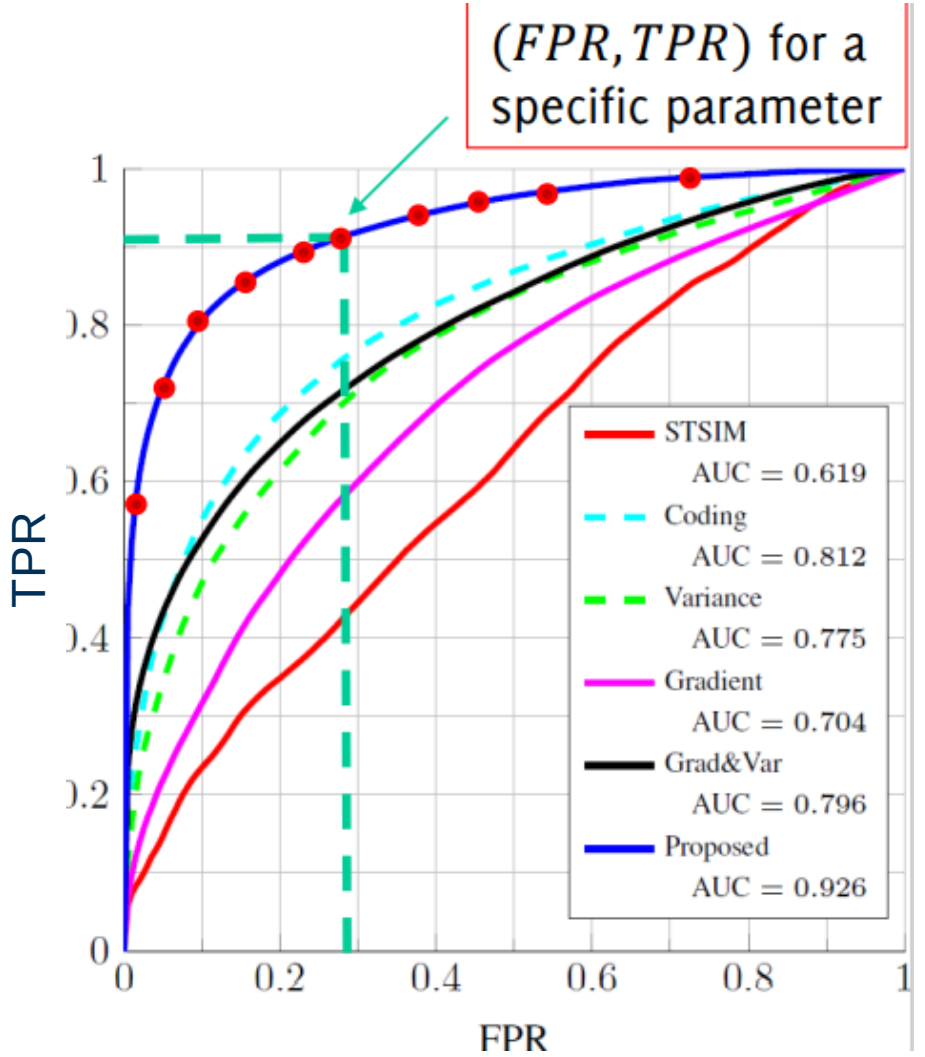
TPR/FPR Tradeoff

- Thus, to correctly assess performance it is necessary to consider at least two indicators (e.g., TPR, FPR)
- Example indicators combining both TPR and FPR:
  - $\text{Accuracy} = \frac{\#\{\text{anomalies detected}\} + \#\{\text{normal samples not detected}\}}{\#\{\text{samples}\}}$
  - $\text{F1 score} = \frac{2\#\{\text{anomalies detected}\}}{\#\{\text{detections}\}+\#\{\text{anomalies}\}}$
- These equal 1 in case of “ideal detector” which detects all the anomalies and has no false positives

Performance Measures

Area Under Curve (AUC)

- Comparing different methods might be tricky since we have to make sure that both have been configured in their best conditions
- Testing a large number of parameters lead to the ROC (receiver operating characteristic) curve
- The ideal detector would achieve: FPR = 0%, TPR = 100%
- Thus, the closer to (0,1) the better
- The larger the Area Under the Curve (AUC), the better
- The optimal parameter is the one yielding the point closest to (0,1)



Overview

In this Lecture..

- Anomaly Definition
- Problem Setup
- Performance Metrics
- Anomaly Detection Settings
  - Semi-supervised
  - Unsupervised
- Statistical Methods
- Reconstruction Methods
- GradCON

Anomaly Detection

Labeled Data Assumptions

- Often in classification/regression/segmentation, only a training set TR is provided
- There are three scenarios:
  - **Semi-Supervised**: Only normal training data are provided, i.e. no anomalies in TR.
  - **Unsupervised**: TR is provided without label.
  - **Supervised**: Both normal and anomalous training data are provided in TR

Semi-supervised Anomaly Detection

Settings

- In semi-supervised methods the TR is composed of normal data

$$\mathrm{TR} = \{x(t), x \sim \phi_0 \text{ and } t < t_0\}$$

- Very practical assumptions:
  - Normal data are easy to gather and the vast majority
  - Anomalous data are difficult/costly to collect/select and it would be difficult to gather a representative training set
  - Training examples in TR might not be representative of all the possible anomalies that can occur
- All in all, it is often safer to detect any data departing from the normal conditions
- Semi-supervised anomaly-detection methods are also referred to as novelty-detection method

Semi-supervised Anomaly Detection

Density-based Methods

- Fall into umbrella of semi-supervised methods
- Monitoring the log-likelihood of data w.r.t ϕ_0 allow to address anomaly-detection problems in multivariate data

1. During training, estimate ϕ_0 from TR
2. During testing, compute $\mathscr{L}\big(x(t)\big) = \log(\phi_0(x(t))$
3. Monitor $\{\mathscr{L}\big(x(t)\big), t = 1, \dots\}$

This is quite a popular approach in either anomaly and change detection algorithm

Semi-supervised Anomaly Detection

Density-based Methods

- Fall into umbrella of semi-supervised methods
- *Assumption*: Normal data occur in high probability regions of a stochastic model, while anomalies occur in the low probability regions of the mode
- **During training:** ϕ_0 can be estimated from the training set using e.g., GMMs
- **During testing:** Anomalies are detected as data yielding $\phi_0(x) < \eta$

Semi-supervised Anomaly Detection

Density-based Methods

Advantages:

- $\phi_0(x)$ indicates how safe a detection is (like a p-value)
- If the density estimation process is robust to outliers, it is possible to tolerate few anomalous samples in TR

Challenges:

- Challenging to fit models for high dimensional data

Unsupervised Anomaly Detection

Setting

- The training set TR might contain both normal and anomalous data. However, no labels are provided

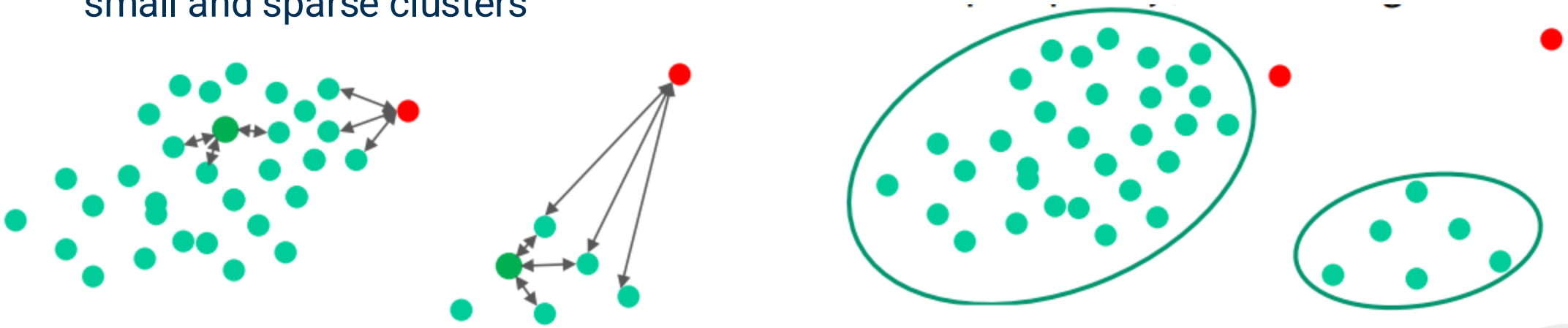
$$\text{TR} = \{x(t), \text{and } t < t_0\}$$

- Underlying assumption: anomalies are rare w.r.t normal data TR
- In principle:
  - Density/Domain based methods that are robust to outliers can be applied in an unsupervised scenario
  - Unsupervised methods can be improved whenever labels are available

Unsupervised Anomaly Detection

Distance-based Methods

- **Distance-based methods:** normal data fall in dense neighborhoods, while anomalies are far from their closest neighbors.
- A critical aspect is the choice of the similarity measure to use.
- Anomalies are detected by monitoring:
  - distance between each data and its k−nearest neighbor
  - the above distance considered relatively to neighbors
  - whether they do not belong to clusters, or are at the cluster periphery, or belong to small and sparse clusters



Unsupervised Anomaly Detection

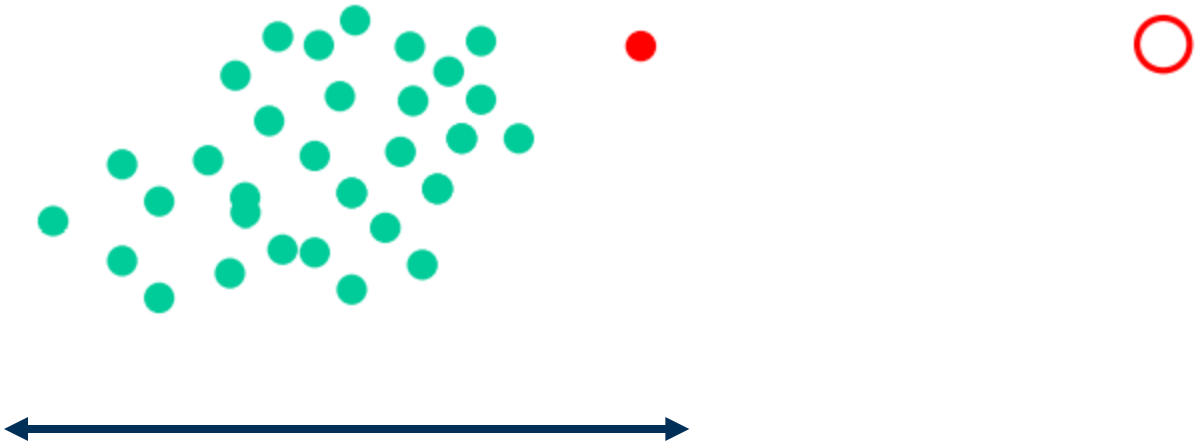
Isolation Forest

- Builds upon the rationale that "anomalies are easier to separate from the rest of normal data"
- This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure

Unsupervised Anomaly Detection

Isolation Forest: Step-by-step

- Builds upon the rationale that "anomalies are easier to separate from the rest of normal data"
- This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure



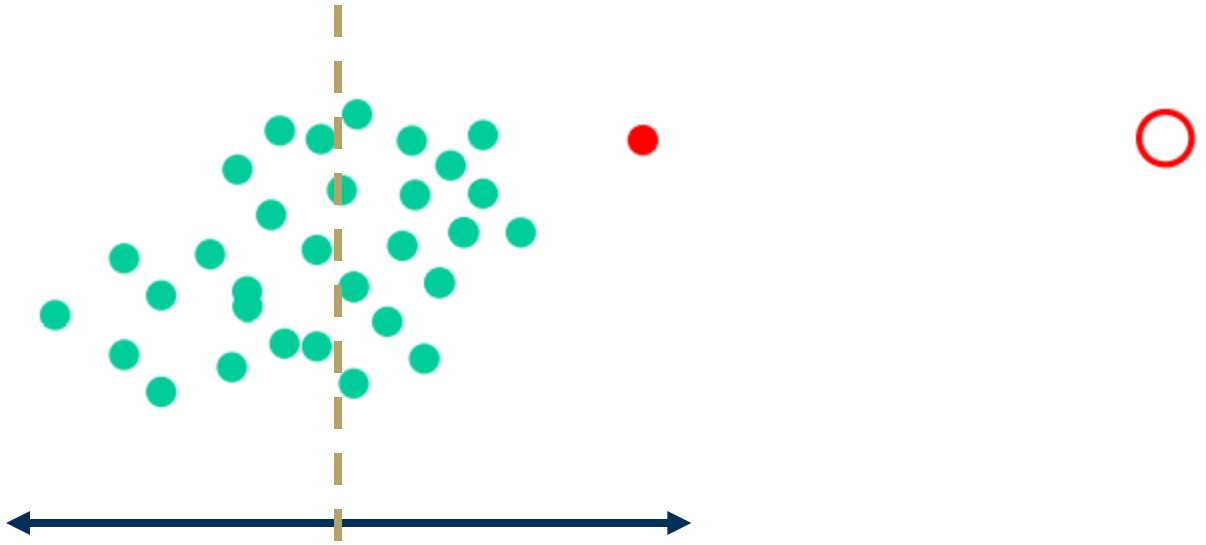
Randomly choose

- A component x_i

Unsupervised Anomaly Detection

Isolation Forest: Step-by-step

- Builds upon the rationale that "anomalies are easier to separate from the rest of normal data"
- This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure



Randomly choose

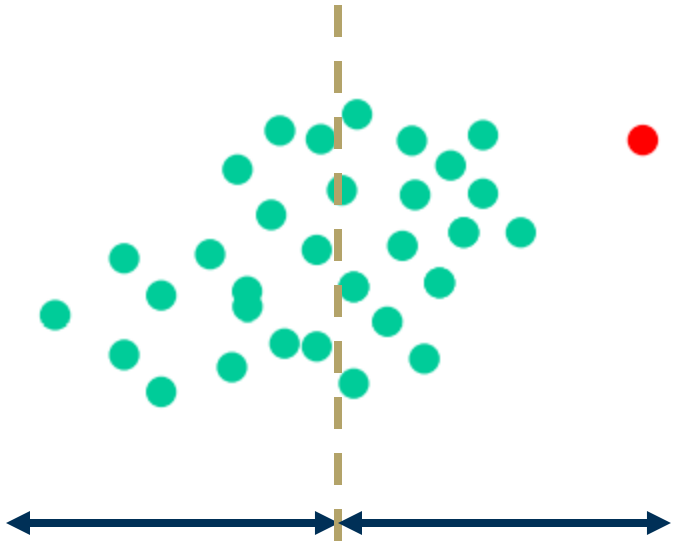
- A component x_i
- A value in the range of projections of TR over the i-th component

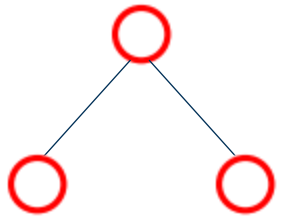
This yields a splitting

Unsupervised Anomaly Detection

Isolation Forest: Step-by-step

- Builds upon the rationale that "anomalies are easier to separate from the rest of normal data"
- This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





Randomly choose

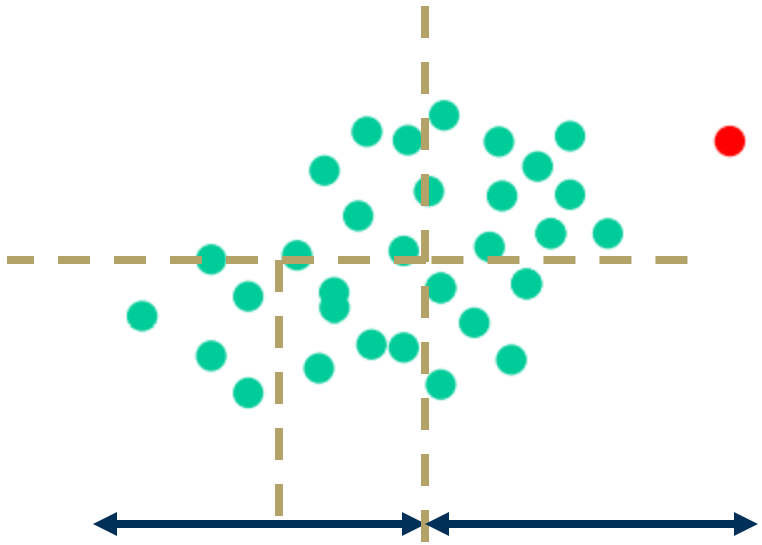
- A component x_i
- A value in the range of projections of TR over the i-th component

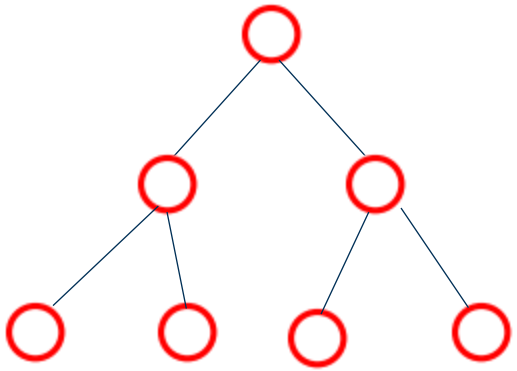
This yields a splitting

Unsupervised Anomaly Detection

Isolation Forest: Step-by-step

- Builds upon the rationale that "anomalies are easier to separate from the rest of normal data"
- This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





Repeat

Randomly choose

- A component x_i
- A value in the range of projections of TR over the i-th component

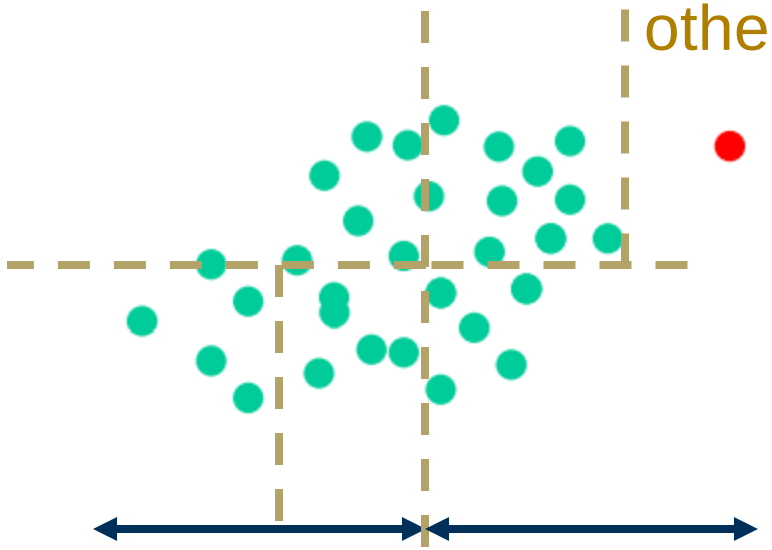
This yields a splitting

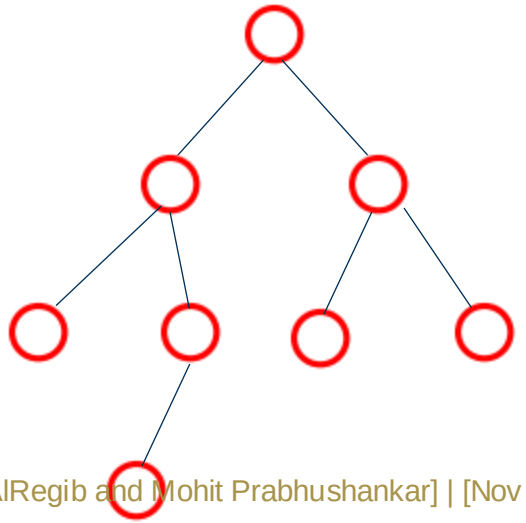
Unsupervised Anomaly Detection

Isolation Forest: Step-by-step

- Builds upon the rationale that "anomalies are easier to separate from the rest of normal data"
- This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure

Anomaly lies separated from the other points





Repeat

Randomly choose

- A component x_i
- A value in the range of projections of TR over the i-th component

This yields a splitting

Overview

In this Lecture..

- Anomaly Definition
- Problem Setup
- Performance Metrics
- Anomaly Detection Settings
- Statistical Methods
  - Image Patch-based analysis
  - Manifolds
  - MSP
- Reconstruction Methods
- GradCON

Anomaly Detection

Patch-based Image Analysis

- Analyze image to isolate normal patches

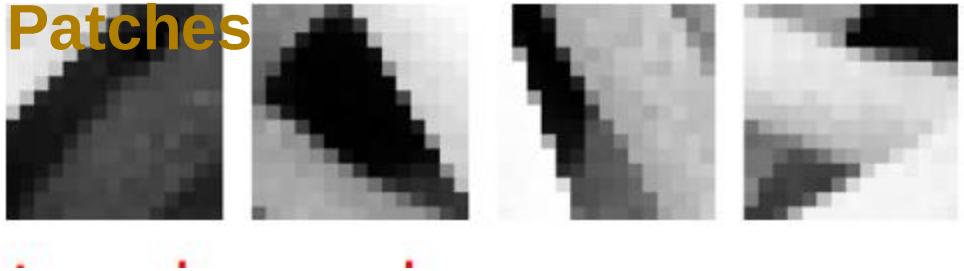
Training:

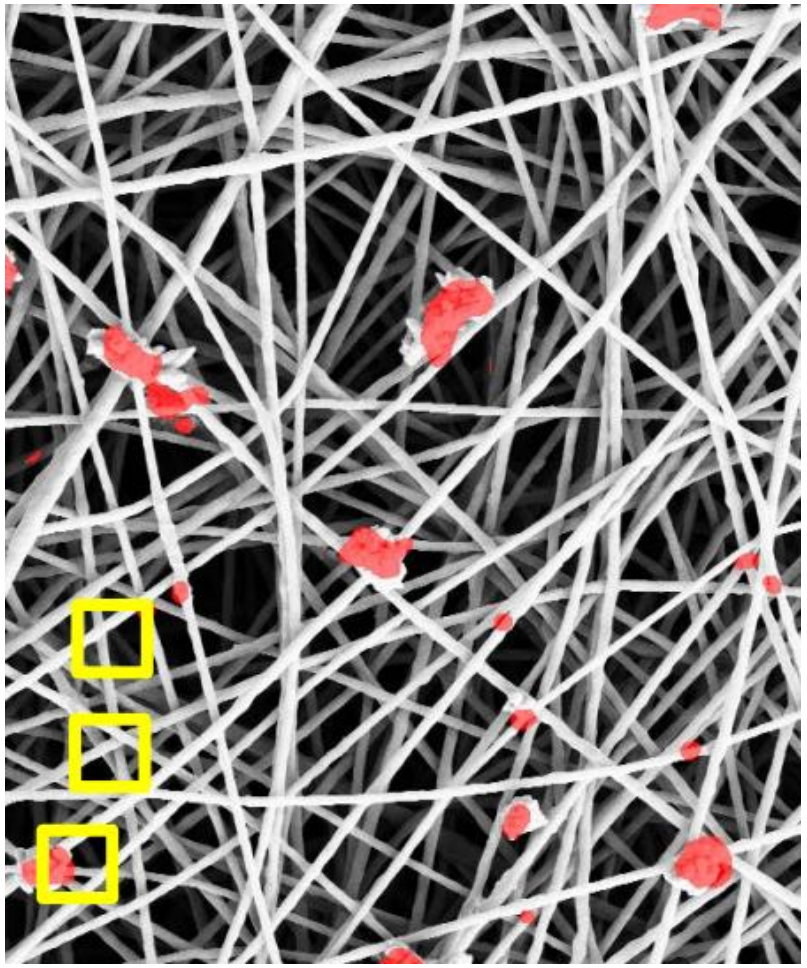
- Split normal images in patches s
- Fit a statistical model $\phi_0 = N(\mu, \Sigma)$ describing normal patches

Testing:

- Split test image in patches
- Compute $\phi_0(s)$ likelihood of each test patch s
- Detect anomalies by thresholding likelihood

Normal Patches

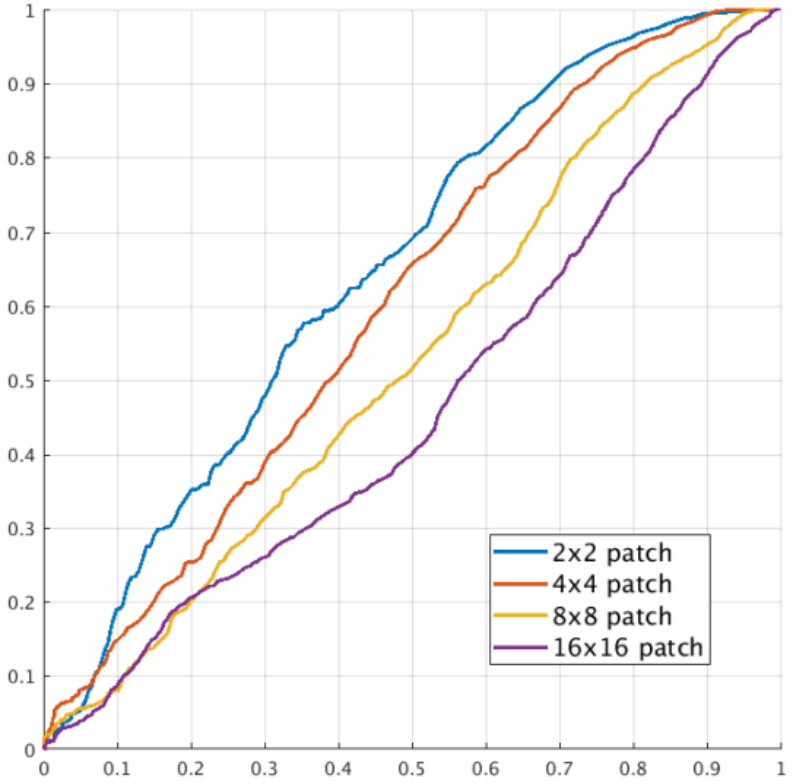




Anomaly Detection

Patch-based Image Analysis

- Model assumes image pixels to be decorrelated from each other
- Poor performance for that deteriorates the greater the patch size

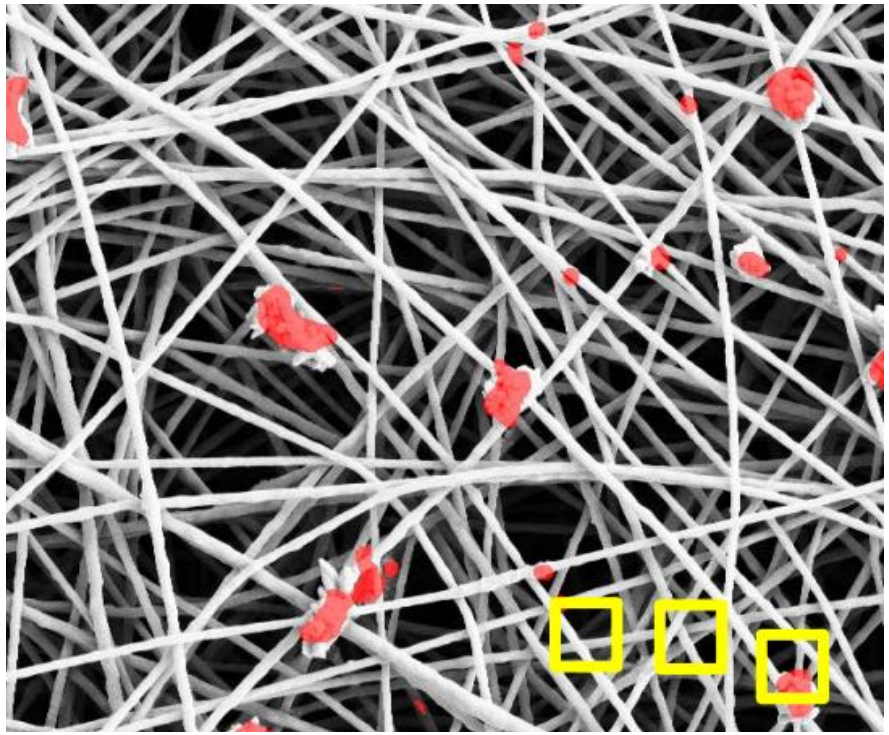


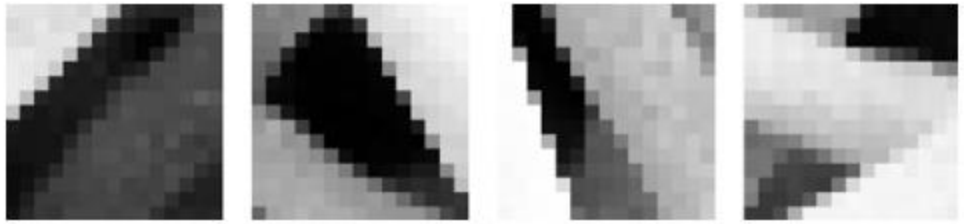
Stacking each patch $s \in \mathbb{R}^d$ in a vector x is not convenient since:

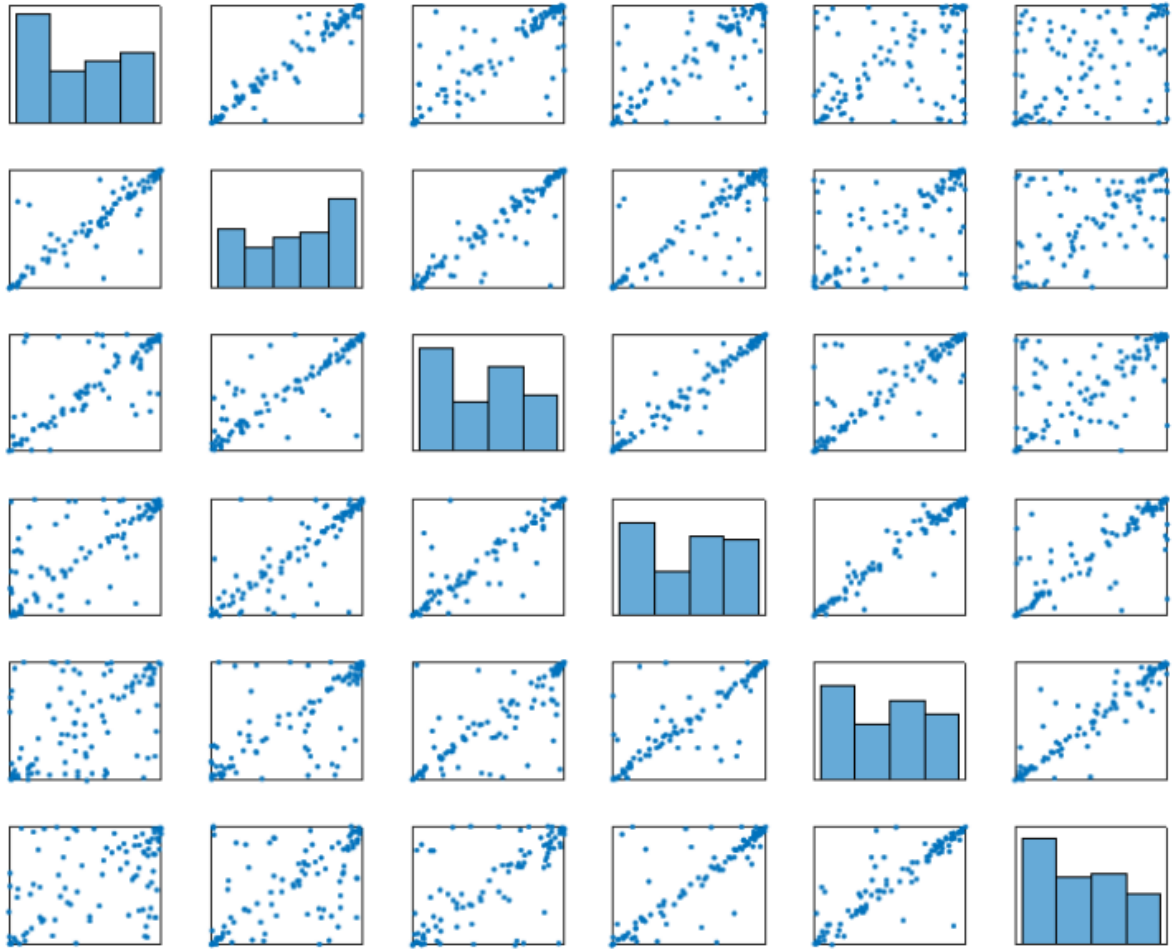
- Data dimension d becomes huge
- Strong correlations among components difficult to model with a PDF ϕ_0.

Anomaly Detection

Adjacent Pixel-value Distribution

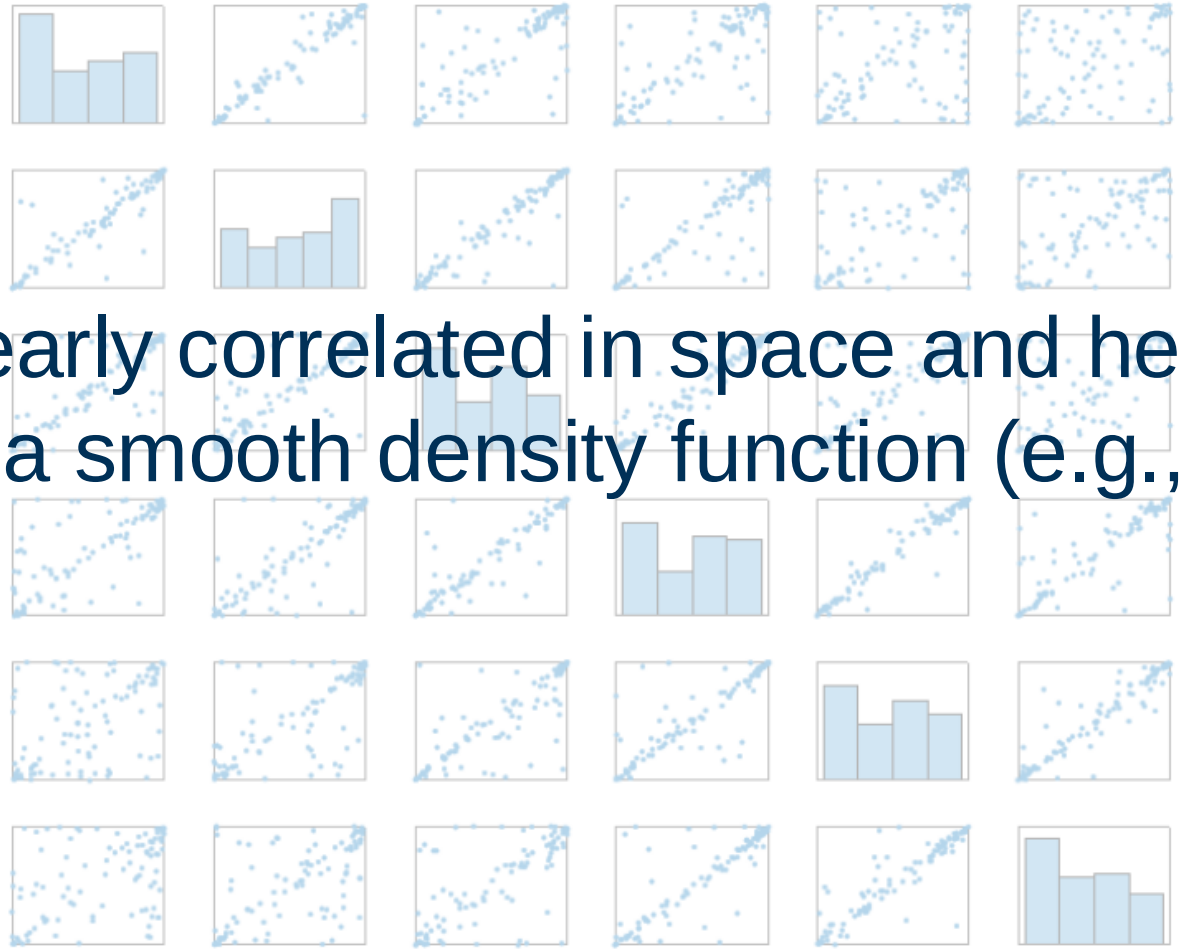






Anomaly Detection

Adjacent Pixel-value Distribution

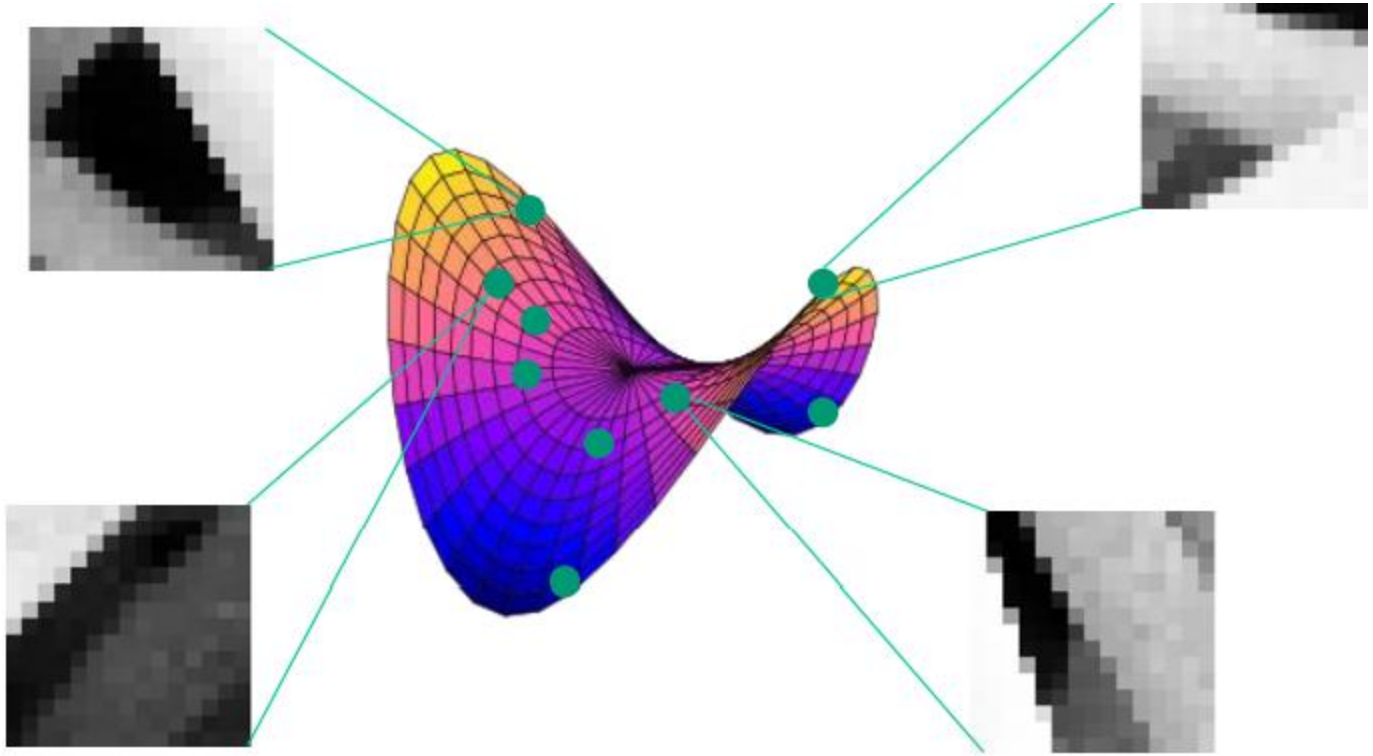


Data are clearly correlated in space and hence difficult to model via a smooth density function (e.g., Gaussians)

Anomaly Detection

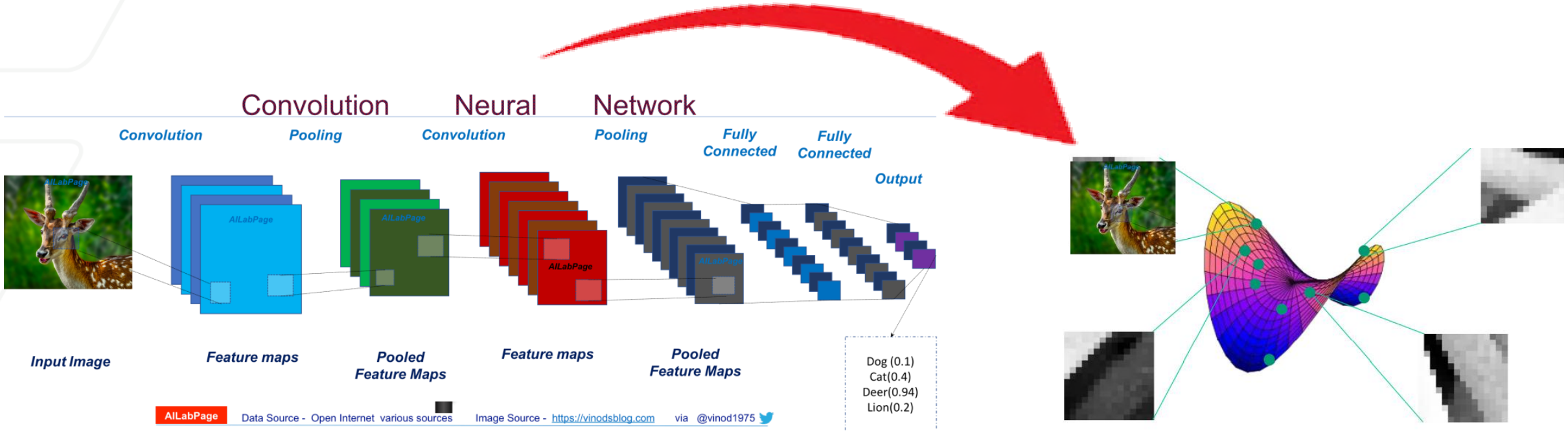
Manifolds and Natural Images

- Patches from natural images live close to a low dimensional manifold
- These means that patches can be well described by few latent variables



Anomaly Detection

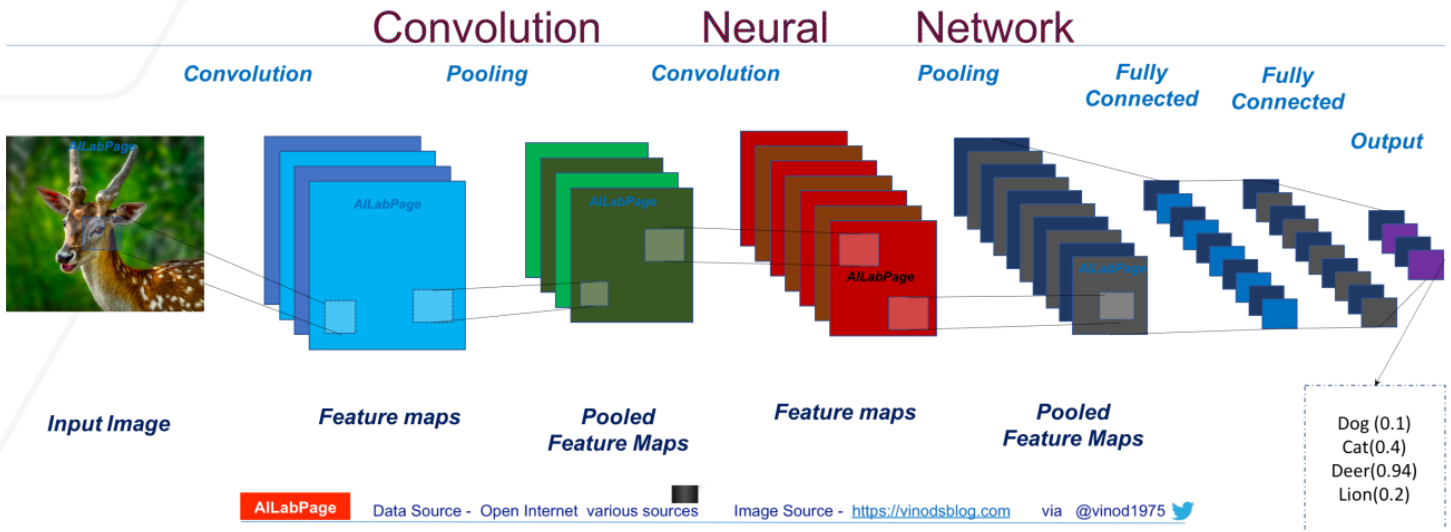
Manifolds and Natural Images



Latent representations of similar patches live closer together within the manifold

Anomaly Detection

Maximum Softmax Probability (MSP)



Based on Softmax scores:

- Construct a threshold based on softmax scores
- Normal data prediction probability is greater than threshold
- Anomalous data probability does not pass the threshold

Allows us to view anomalies via train (in-distribution) and test distributions (out-of-distribution - OOD)

Hendrycks, D., & Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Overview

In this Lecture..

- Anomaly Definition
- Problem Setup
- Performance Metrics
- Anomaly Detection Settings
- Statistical Methods
- Reconstruction Methods
- GradCON

Anomaly Detection

Reconstruction-based Methods

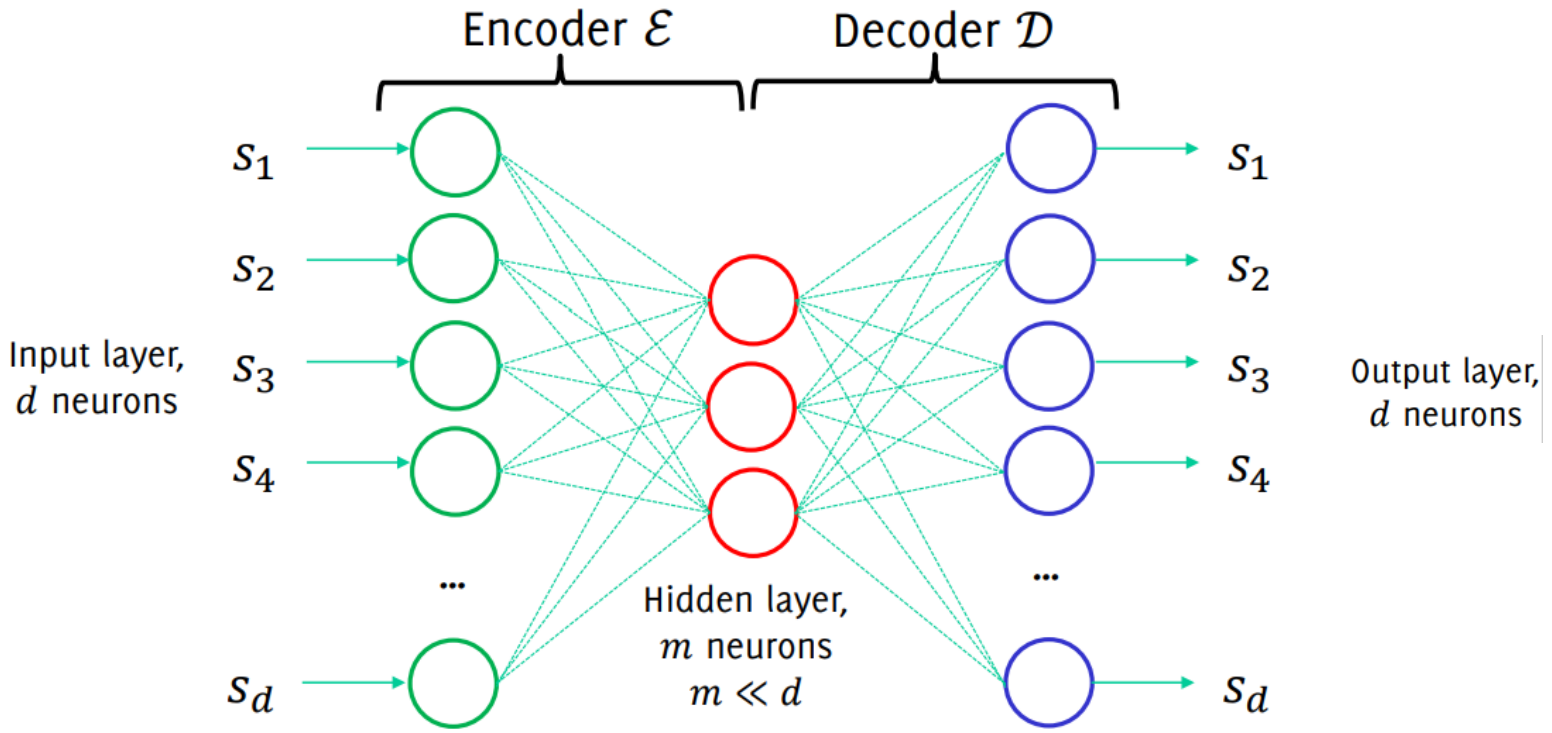
Fit a statistical model to the observation to describe dependence, apply anomaly detection on the independent residuals.

- Detection is performed by using a model $\mathcal{M}$ which can encode and reconstruct normal data:
  - During training: learn the model $\mathcal{M}$ from training set TR
  - During testing:
    - Encode and reconstruct each test signal $\boldsymbol{s}$ through $\mathcal{M}$
    - Assess err($\boldsymbol{s}$), namely the residual between $\boldsymbol{s}$ and its reconstruction through $\mathcal{M}$
- The rationale is that $\mathcal{M}$ can reconstruct only normal data, thus anomalies are expected to yield large reconstruction errors.

Reconstruction-based Methods

Autoencoders

- Autoencoders are neural networks used for data reconstruction (they learn the identity function)
- The typical structure of an autoencoder is:



OLIVES @GeorgiaTech
Georgia Tech

Reconstruction-based Methods

Autoencoders

- Autoencoders are trained to reconstruct all the samples in the training set. The reconstruction loss over the training set TR is

$$L(TR) = \sum_{s \in TR} ||s - D\big(E(s)\big)||_2$$

- And training of $D\big(E(\cdot)\big)$ is performed through standard backpropagation methods (e.g., SGD)

Remarks:

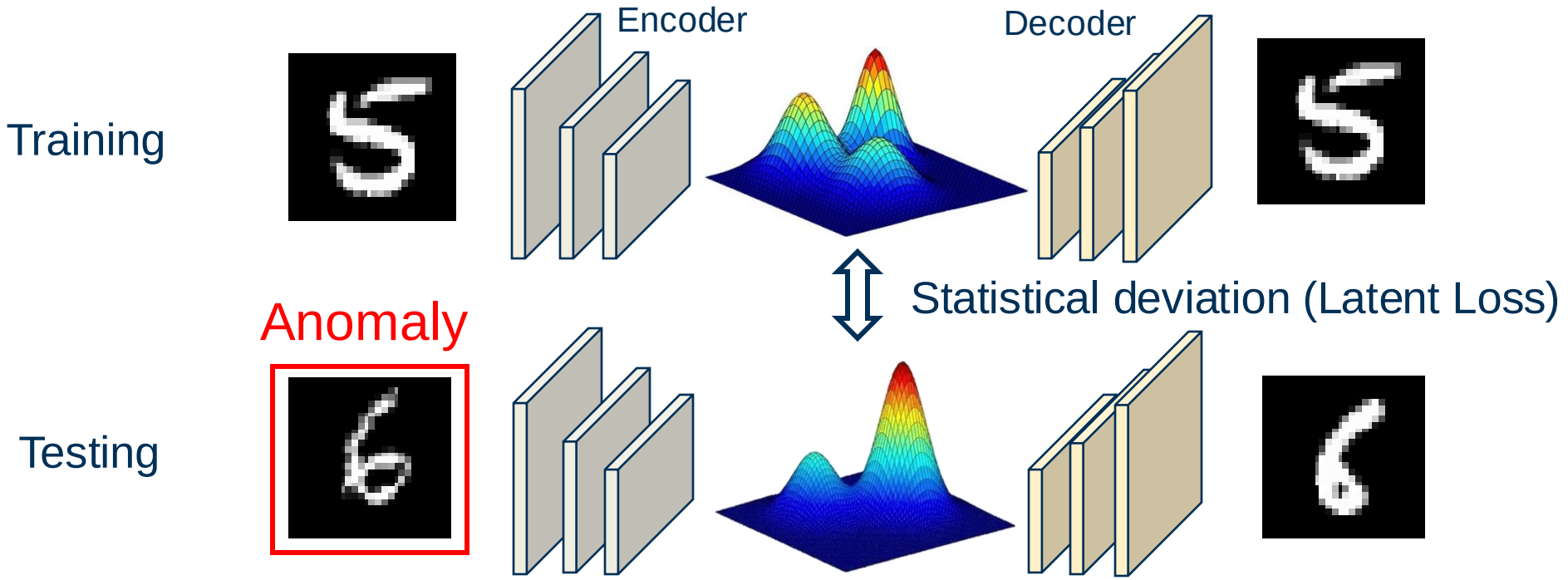
- Typically, $D(E(\cdot))$ does not provide perfect reconstruction, since $m \ll d$.
- Regularization terms might be included in the loss function for latent representations $E(s)$ to have specific feature properties.

Reconstruction-based Methods

Constraining Activations

Constrained Representation

Some quantity is constrained to identify anomalies



[1] David MJ Tax and Robert PW Duin. Support vector data description. Machine learning, 54(1):45–66, 2004.
[2] Yaxiang Fan, Gongjian Wen, Deren Li, Shaohua Qiu, and Martin D Levine. Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder. arXiv preprint arXiv:1805.11223, 2018. 1, 2
[3] S. Pidhorskyi, R. Almohsen, and G. Doretto, “Generative probabilistic novelty detection with adversarial autoencoders,” in Advances in Neural Information Processing Systems, 2018, pp. 6822–6833.
[4] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, “Latent space autoregression for novelty detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 481–490.

OLIVES

Georgia Tech

Reconstruction-based Methods

Other Constraints

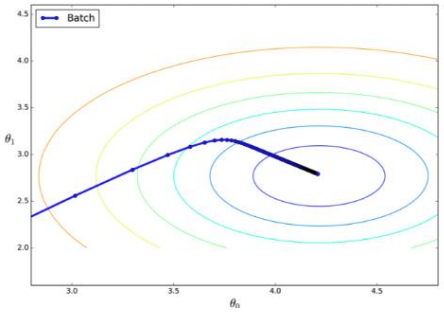
What other constraints have we seen? – Gradients

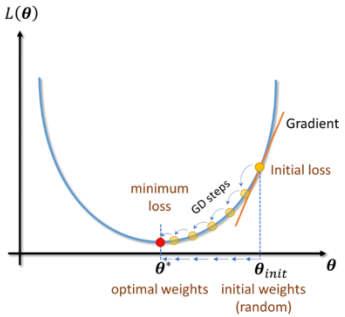
Gradients for learning

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) - \alpha \frac{\partial\, L(\boldsymbol{\theta})}{\partial \boldsymbol{W}}$$

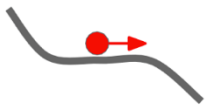
$$\boldsymbol{b}(\mathrm{t}+1) = \boldsymbol{b}(\mathrm{t}) - \alpha \frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{b}}$$

Convergence path of Batch GD

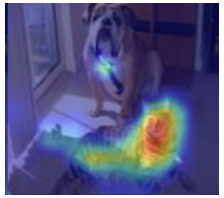




Saddle points



Gradients for explanations

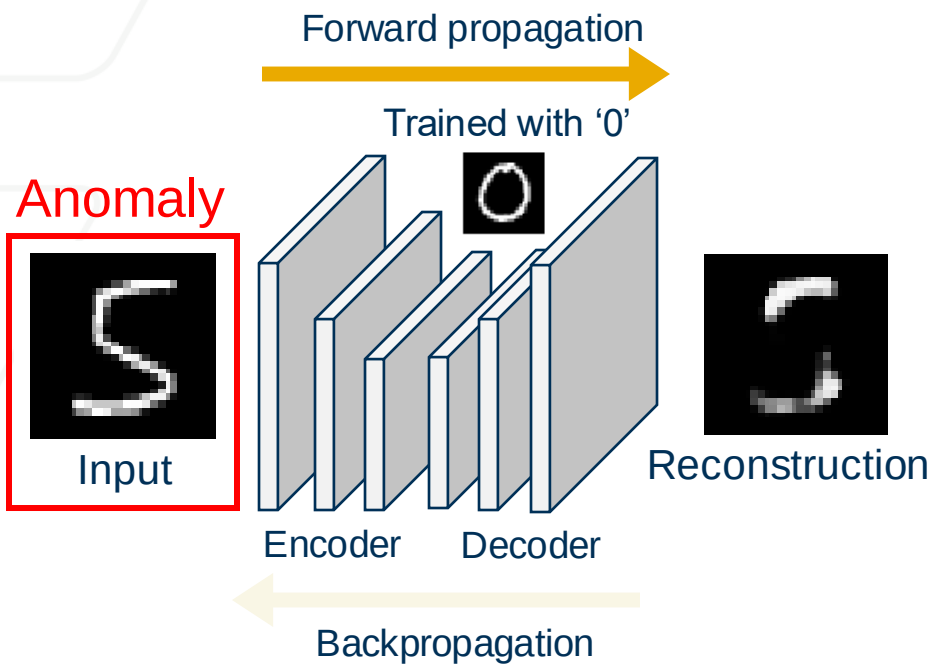


$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i \sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial J(P,Q)}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

Reconstruction-based Methods

Gradient Constraints



Activation Constraints

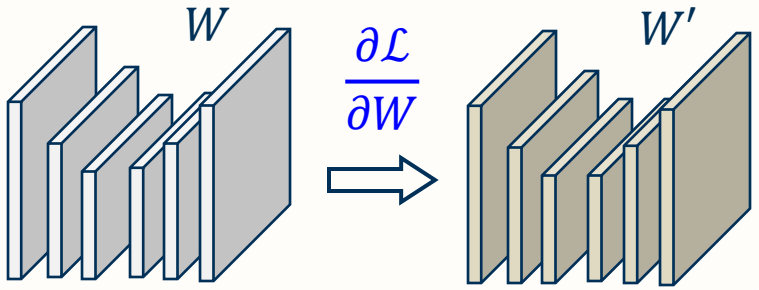
Activation-based representation
(Data perspective)

e.g. Reconstruction error ($\mathcal{L}$)

How much of the input does not correspond to the learned information?

Gradient Constraints

Gradient-based Representation
(**Model** perspective)



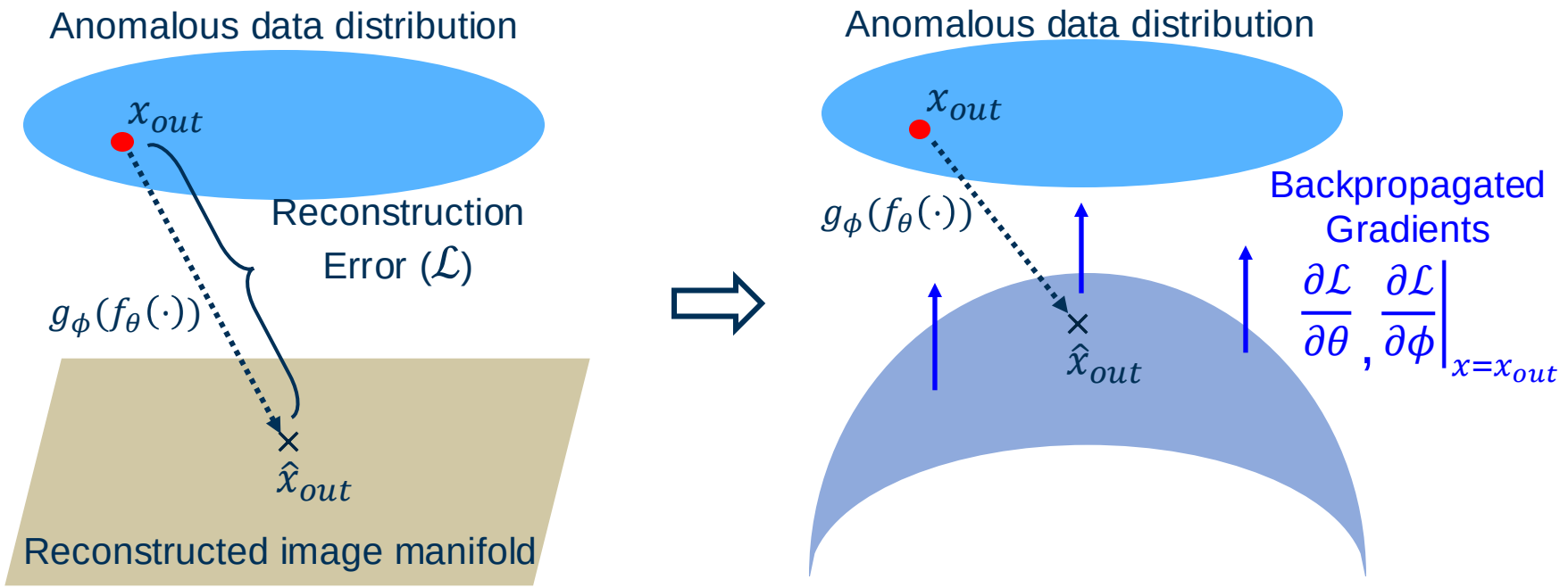
How much **model update** is required by the input?

G. Kwon, M. Prabhushankar, D. Temel, and G. AlRegib, "Novelty Detection Through Model-Based Characterization of Neural Networks," 2020

G. Kwon, M. Prabhushankar, D. Temel, and G. AlRegib, "Backpropagated Gradient Representations for Anomaly Detection," 2020

Reconstruction-based Methods

Gradient Constraints: Geometric Interpretation



1) Provide directional information to characterize anomalies

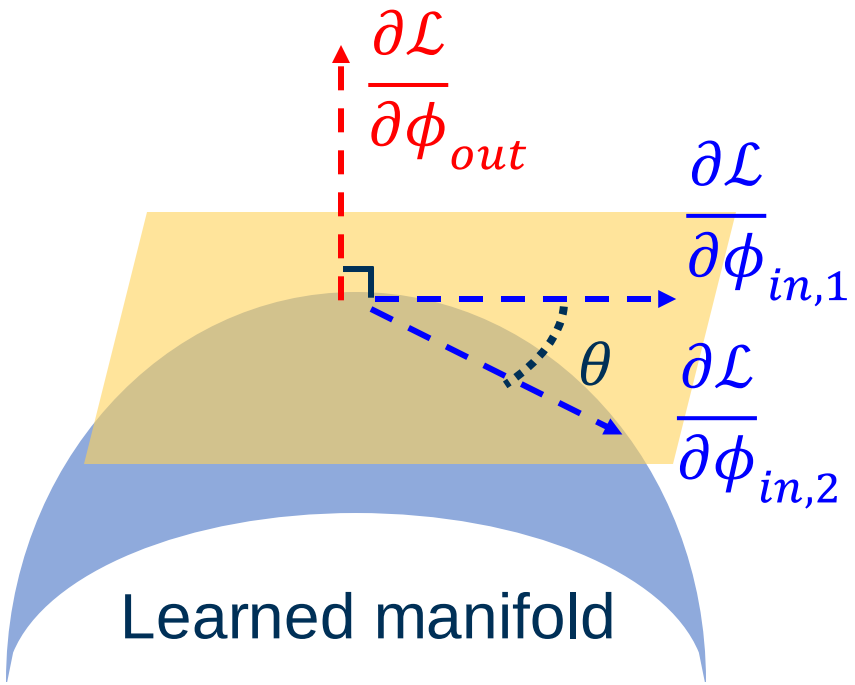
2) Gradients from different layers capture abnormality at different levels of data abstraction

G. Kwon, M. Prabhushankar, D. Temel, and G. AlRegib, "Backpropagated Gradient Representations for Anomaly Detection," 2020

Reconstruction-based Methods

GradCON Method

Constrain gradient-based representations during training to obtain clear separation between normal data and abnormal data



ϕ: Weights $\mathcal{L}$: Reconstruction error

At *k*-th step of training,

Gradient loss

$$J = \mathcal{L} - \mathbb{E}_i \left[\text{cosSIM} \left(\frac{\partial J}{\partial \phi_i}^{k-1}, \frac{\partial \mathcal{L}}{\partial \phi_i}^{k} \right) \right]$$

Avg. training gradients until (k-1) th iter.

Gradients at k-th iter.

where $\frac{\partial J}{\partial \phi_{i_{avg}}}^{k-1} = \sum_{t=1}^{k-1} \frac{\partial J}{\partial \phi_i}^{t}$

G. Kwon, M. Prabhushankar, D. Temel, and G. AlRegib, "Backpropagated Gradient Representations for Anomaly Detection," 2020

Reconstruction-based Methods

With and Without GradCON

Anomalous “class” detection (CIFAR-10)

e.g.

Normal

Abnormal

AUROC Results

Model	Loss	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Average
CAE	Recon	0.682	0.353	0.638	0.587	0.669	**0.613**	0.495	0.498	0.711	0.390	0.564
CAE	Recon	0.659	0.356	**0.640**	0.555	0.695	0.554	0.549	0.478	0.695	0.357	0.554
+ Grad	Grad	**0.752**	0.619	0.622	0.580	0.705	0.591	0.683	**0.576**	**0.774**	**0.709**	**0.661**
VAE	Recon	0.553	0.608	0.437	0.546	0.393	0.531	0.489	0.515	0.552	0.631	0.526
	Latent	0.634	0.442	**0.640**	0.497	**0.743**	0.515	**0.745**	0.527	0.674	0.416	0.583
VAE	Recon	0.556	0.606	0.438	0.548	0.392	0.543	0.496	0.518	0.552	0.631	0.528
+ Grad	Latent	0.586	0.396	0.618	0.476	0.719	0.474	0.698	0.537	0.586	0.413	0.550
	Grad	0.736	**0.625**	0.591	**0.596**	0.707	0.570	0.740	0.543	0.738	0.629	0.647

Recon: Reconstruction error, Latent: Latent loss, Grad: Gradient loss

1) (CAE vs. CAE + Grad) Effectiveness of the gradient constraint

2) (CAE vs. VAE) Performance sacrifice from the latent constraint

3) (VAE vs. VAE + Grad) Complementary features from the gradient constraint

G. Kwon, M. Prabhushankar, D. Temel, and G. AlRegib, "Backpropagated Gradient Representations for Anomaly Detection,” 2020

Reconstruction-based Methods

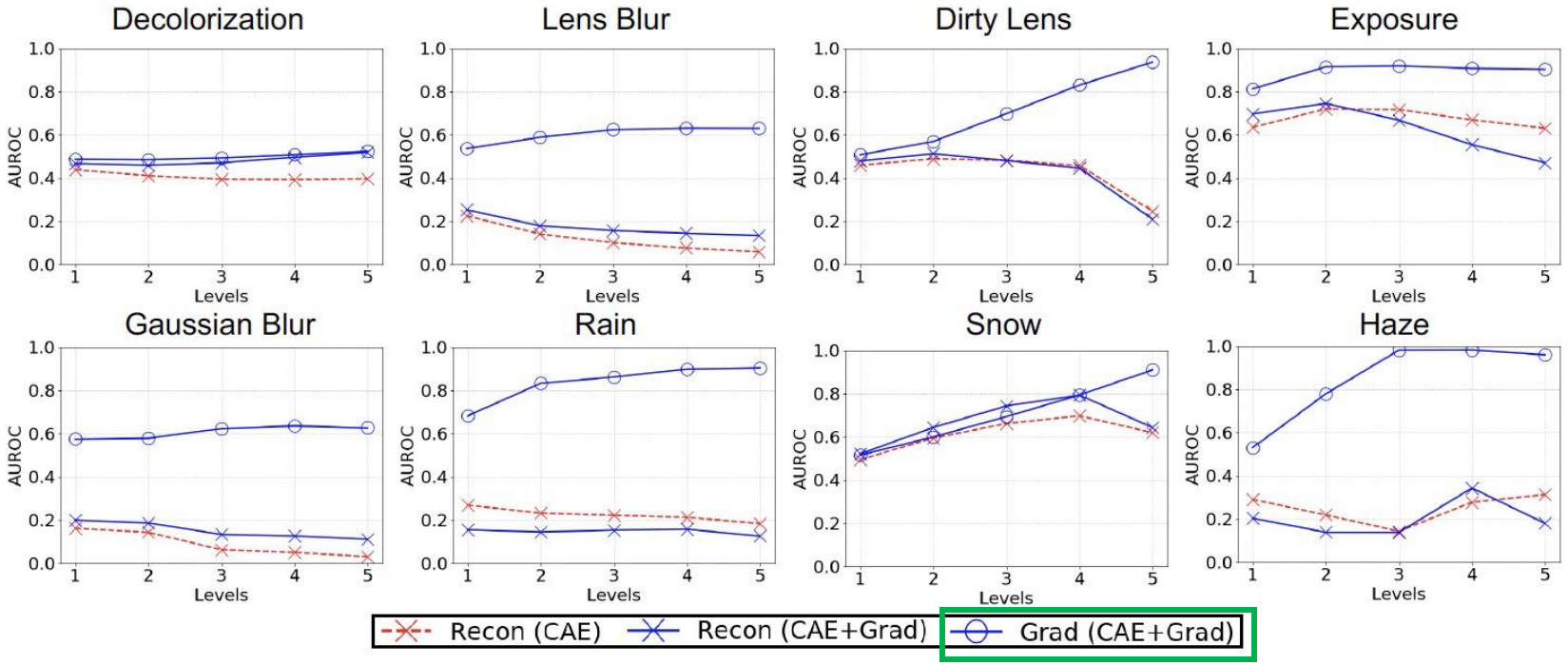
With and Without GradCON: Abnormal condition detection

Abnormal “condition” detection (CURE-TSR)

Normal

Anomalous

AUROC Results



Recon: Reconstruction error, Grad: Gradient loss

G. Kwon, M. Prabhushankar, D. Temel, and G. AlRegib, "Backpropagated Gradient Representations for Anomaly Detection,” 2020

Acknowledgements

- A significant portion of the material has been adapted from tutorial slides by Boracchi and Carrera delivered at the ICIP 2020.

Terminology

- $Distribution$: (sample space) the set of all possible samples
- $Dataset$: a set of samples drawn from a distribution
- $Batch$: a subset of samples drawn from the dataset
- $Sample$: a single data object represented as a set of features
- $Feature$: value of a single attribute, property, in a sample. Could be numeric or categorical.

Appendix

Notations

- x_i: a single feature
- $\boldsymbol{x}_i$: feature vector (a data sample)
- $\boldsymbol{x}_{:,i}$: feature vector of all data samples
- $\boldsymbol{X}$: matrix of feature vectors (dataset)
- N: number of data samples
- $\boldsymbol{W}$: weight matrix
- $\boldsymbol{b}$: bias vector
- $\boldsymbol{v}(t)$: first moment at time t
- $\boldsymbol{G}(t)$: second moment at time t
- $\boldsymbol{H}(\boldsymbol{\theta})$: Hessian matrix
- P: number of features in a feature vector
- α: learning rate
- Bold letter/symbol: vector
- Bold capital letters/symbol: matrix