Data Mining

Chapter 8: Outlier Detection

Yunming Ye, Baoquan Zhang

School of Computer Science

Harbin Institute of Technology, Shenzhen

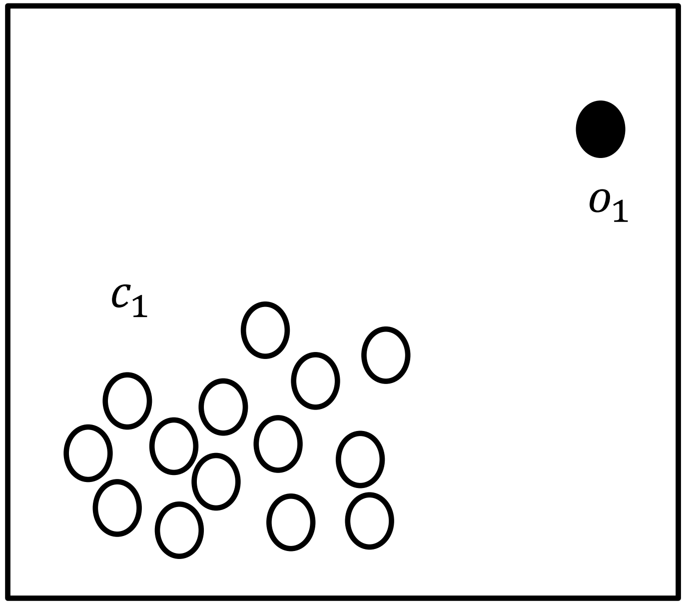
Agenda

- Introduction to Outlier Detection
- Maximum Likelihood Method
- One-class SVM and Isolation Forest
- Reconstruction Methods

8.1 Introduction to Outlier Detection

What Is Outlier Detection?

- An outlier is a data object that *deviates significantly* from the rest of the objects



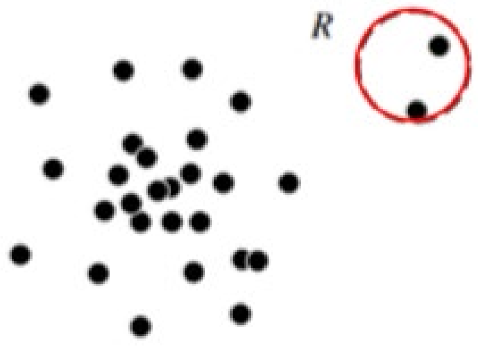
- Outlier detection (anomaly detection) is the process of finding data objects with behaviors that are very different from expectation

Applications of Outlier Detection

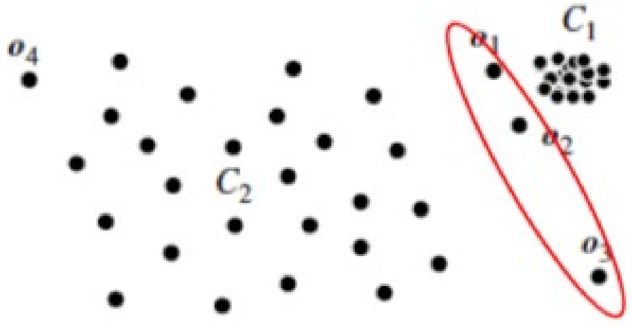
- Network intrusion detection
- Insurance / Credit card fraud detection
- Healthcare Informatics / Medical diagnostics
- Industrial Damage Detection
- Image Processing / Video surveillance
- Novel Topic Detection in Text Mining
- …

Types of outliers

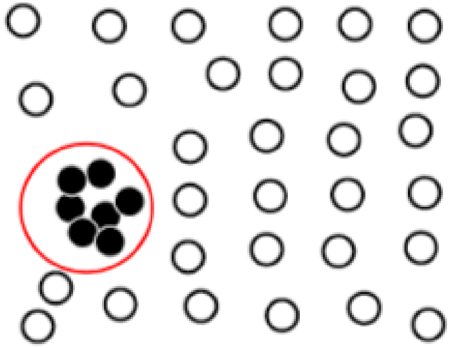
- **Global outliers (point anomalies)**
  - A global outlier deviates significantly from the rest of the data set



- **Contextual outliers**
  - A contextual outlier deviates significantly w.r.t a specific context of the object



- **Collective outliers**
  - A collective outlier refer to objects as a whole deviate significantly from the entire data set



Challenges of Outlier Detection

- **Modeling normal objects and outliers effectively**
- **Application-specific outlier detection**
- **Handling noise in outlier detection**
- **Understandability**

Methods for Outlier Detection

- Supervised methods
  - train classifier with “normal” and “abnormal” data
  - Challenge: umbalanced data
- Unsupervised methods
  - Proximity-based: an outlier’s nearest neighbors should be far away

Nearest-Neighbor Based Approach

- Compute the distance between every pair of data points

- There are various ways to define outliers:
  - Data points for which there are fewer than *p* neighboring points within a distance *D*
  - The top n data points whose distance to the kth nearest neighbor is greatest
  - The top n data points whose average distance to the k nearest neighbors is greatest

Methods for Outlier Detection

- Supervised methods
  - train classifier with “normal” and “abnormal” data
  - Challenge: umbalanced data
- Unsupervised methods
  - Proximity-based: an outlier’s nearest neighbors should be far away
  - Clustering-based: normal data belonging to large and dense clusters
  - One-class Method
    - Statistical method : data normality from some statistical model
    - Other one-class methods
  - Reconstruction method
  - ......

Outlier detection using one-class model

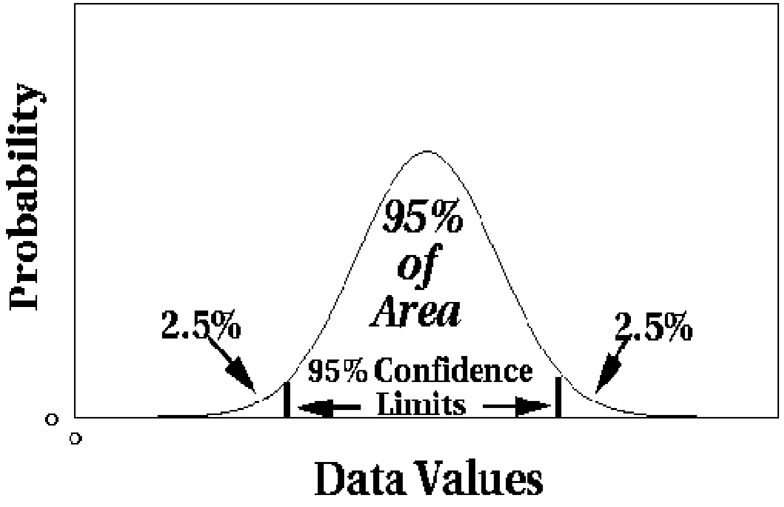
- **Only modeling the normal class (with large amount of objects).**
  - Centroid-based method
  - Statistical method
  - One-class SVM
  - Isolation forest
  - ……

8.2 Maximum Likelihood Method

Statistical Approaches

- Assume a parametric model describing the distribution of the data (e.g., normal distribution)

- Apply a statistical test that depends on
  - Data distribution
  - Parameter of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit)



Maximum Likelihood Method: problem definition

- Given a data set $\mathbb{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(m)}\}$
- Assume the probability density function of $\mathbb{X}$ is known to be $f_\theta(x)$
  - θ is the parameters, and to be learned from data
- Likelihood of $\mathbb{X}$:

$$L(\theta) = \prod_{i=1}^{m} f_\theta(\boldsymbol{x}^{(i)}) = f_\theta\ (\boldsymbol{x}^{(1)}) f_\theta(\boldsymbol{x}^{(2)}) \dots\ f_\theta(\boldsymbol{x}^{(m)})$$

- Maximize:

$$\theta^*:\ \underset{\theta}{\operatorname{argmax}}\, L(\theta) = \underset{\theta}{\operatorname{argmax}} \left(\prod_{i=1}^{m} f_\theta(\boldsymbol{x}^{(i)}) \right)$$

- Given a data set $\mathbb{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \dots, \boldsymbol{x}^{(m)}\}$ $\quad\quad \boldsymbol{x}^{(i)} \in \boldsymbol{R}^d$

- Assume the distribution of $\mathbb{X}$ be Gaussian:

$$\theta\!: \boldsymbol{\mu}, \boldsymbol{\sigma} \qquad f_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}} \frac{1}{|\boldsymbol{\sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}$$

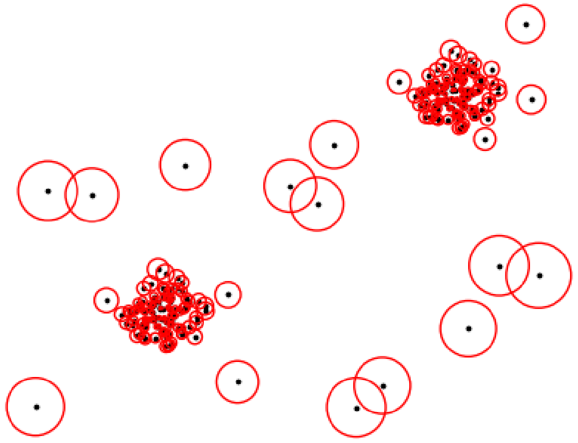
- Maximize:

$$\theta^*\!: \underset{\theta:\, \boldsymbol{\mu},\boldsymbol{\sigma}}{\operatorname{argmax}} L(\theta) = \underset{\boldsymbol{\mu},\boldsymbol{\sigma}}{\operatorname{argmax}} \left(\prod_{i=1}^{m} f_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{x}^{(i)}) \right)$$

$$\boldsymbol{\mu}^* = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}^{(i)} \qquad \boldsymbol{\sigma}^* = \frac{1}{m}\sum_{i=1}^{m} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^*\right)\left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^*\right)^T$$

Detection Phase

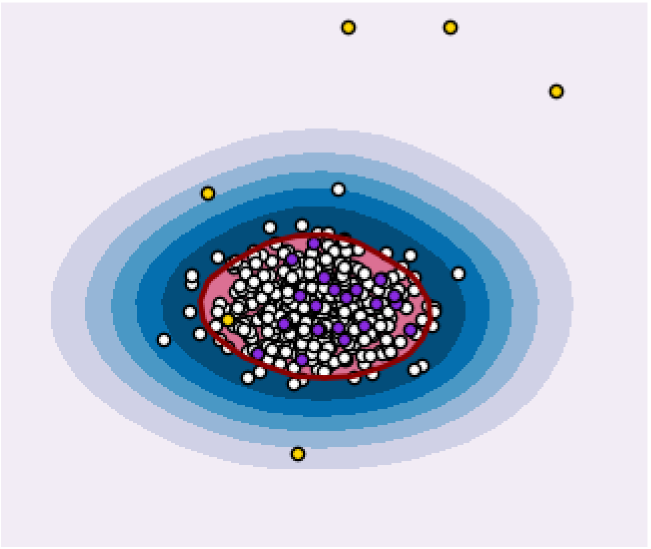
- Given a threshold value $\boldsymbol{\delta}$
- Decide whether a given data object $\boldsymbol{x}$ is an outlier



$$f_{\boldsymbol{\mu}^*,\boldsymbol{\sigma}^*}(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}} \frac{1}{|\boldsymbol{\sigma}^*|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}^*)^T {\boldsymbol{\sigma}^*}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}^*)\right\}$$

$$\begin{cases} f_{\boldsymbol{\mu}^*,\boldsymbol{\sigma}^*}(\boldsymbol{x}) \geq \boldsymbol{\delta} & \text{Normal data object} \\ f_{\boldsymbol{\mu}^*,\boldsymbol{\sigma}^*}(\boldsymbol{x}) < \boldsymbol{\delta} & \text{Outlier} \end{cases}$$

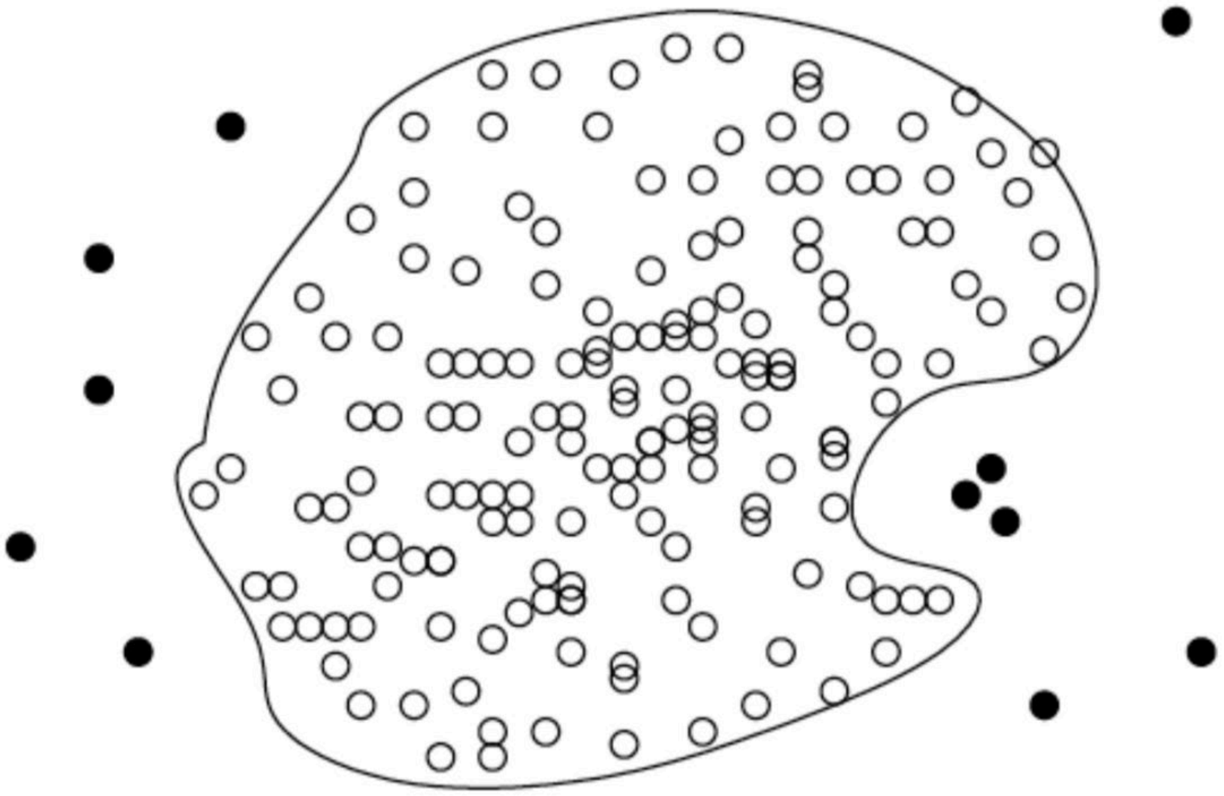
$\boldsymbol{\delta}$ determine the results



8.3 One-class SVM and Isolation Forest

One-class SVM: Basic Idea

- **Learning the Boundary of the input data (i.e. normal class with large amount of objects).**

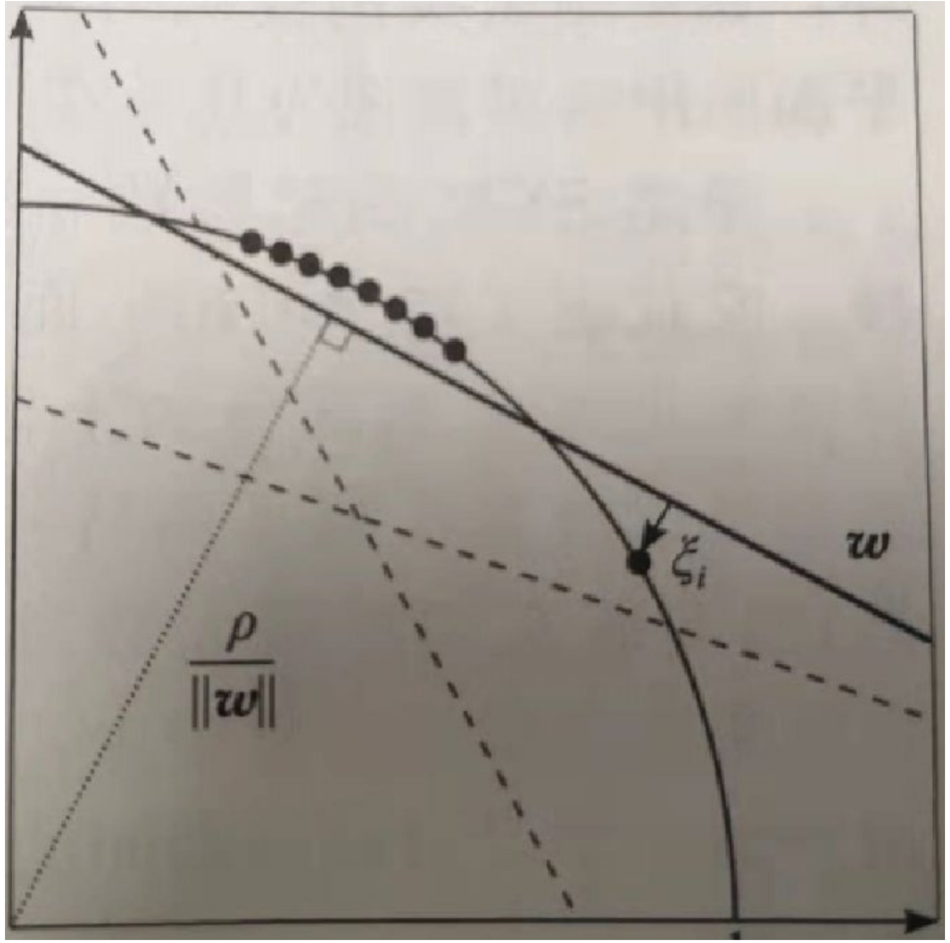


One-class SVM: *ν-SVM*

- How to define the second class?
  - $\emptyset(x)$: Projection to high-dimensional feature space
  - With Gaussian kernel:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right)$$

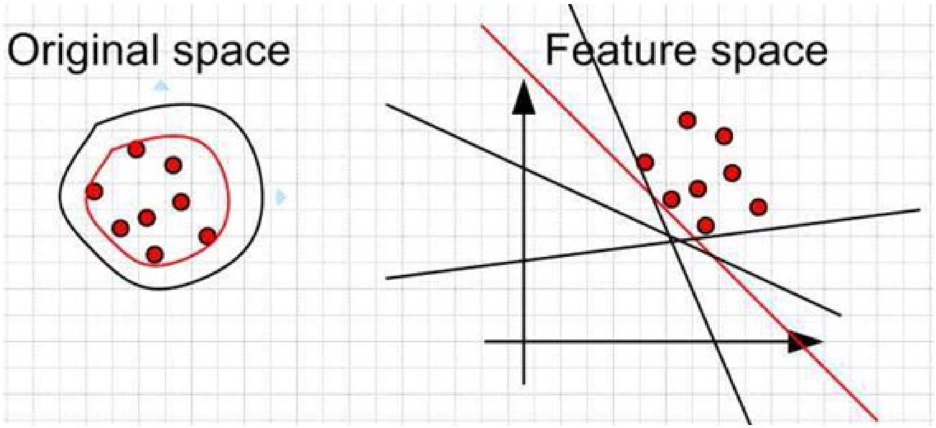
$$K(\boldsymbol{x}, \boldsymbol{x}) \; = < \emptyset(x), \emptyset(x) > \; = \|\emptyset(x)\|^2 \quad = 1$$



- the origin of the feature space is the second class!

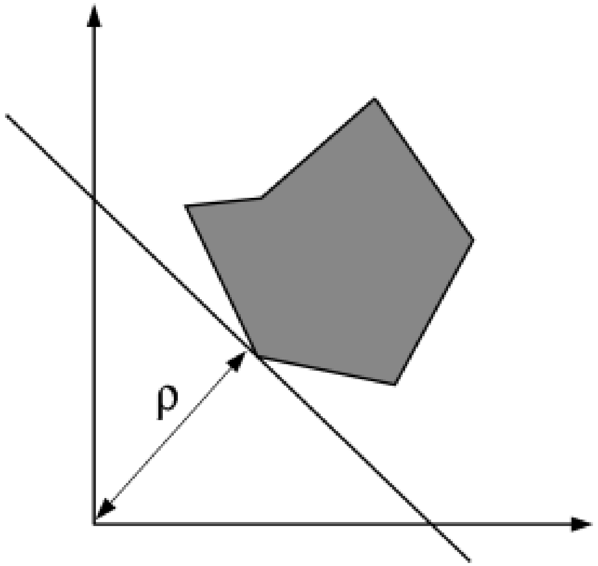
- Define the separating hyperplane:

$$w \cdot \emptyset(x) = \rho$$



- Outlier detection function:

$$f(x) = sgn(< w, \emptyset(x) > -\rho)$$



- Learning task:
  - Learned model: $w \cdot \emptyset(x) = \rho$
  - Objective function:

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{2\sigma^2}\right)$$

$$\left.\begin{aligned} &\min_{\boldsymbol{w},\boldsymbol{\xi},\rho} \frac{\mathbf{1}}{\mathbf{2}}\|\boldsymbol{w}\|^2 + \frac{1}{vm}\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{m}} \xi_i - \rho \\ &\text{subject to: } \langle \boldsymbol{w}, \emptyset(x_i)\rangle \geq \rho - \xi_i, \xi_i \geq 0 \\ &0 < v \leq 1 \end{aligned}\right\}$$

$$\boldsymbol{w}^* = \sum_{i=1}^{m} \lambda_i^* \phi(\boldsymbol{x}_i)$$

$$\rho^* = \sum_{i=1}^{m} \lambda_i^* \phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_s)$$

$$= \sum_{i=1}^{m} \lambda_i^* K(\boldsymbol{x}_i, \boldsymbol{x}_s)$$

- Outlier detection:

$$f(x) = sgn(< w, \emptyset(x) > -\rho)$$

$$= sgn(\sum_{i=1}^{m} \lambda_i^* \phi(\boldsymbol{x}_i)\, \emptyset(x) - \rho) \quad = sgn(\sum_{i=1}^{m} \lambda_i^* K(\boldsymbol{x}_i, x) - \rho)$$

One-class SVM: *SVDD*

- Support Vector Domain Description:

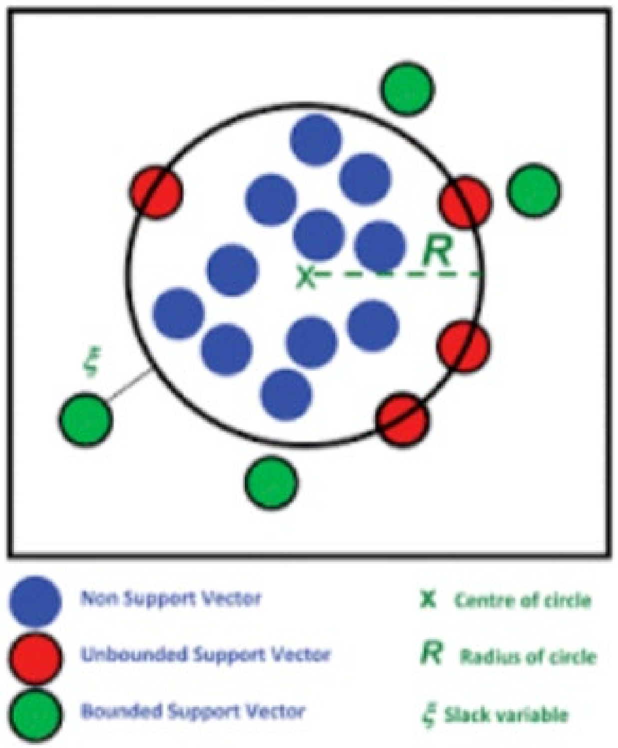
  Constraining "normal" data in a ball with relative small radius

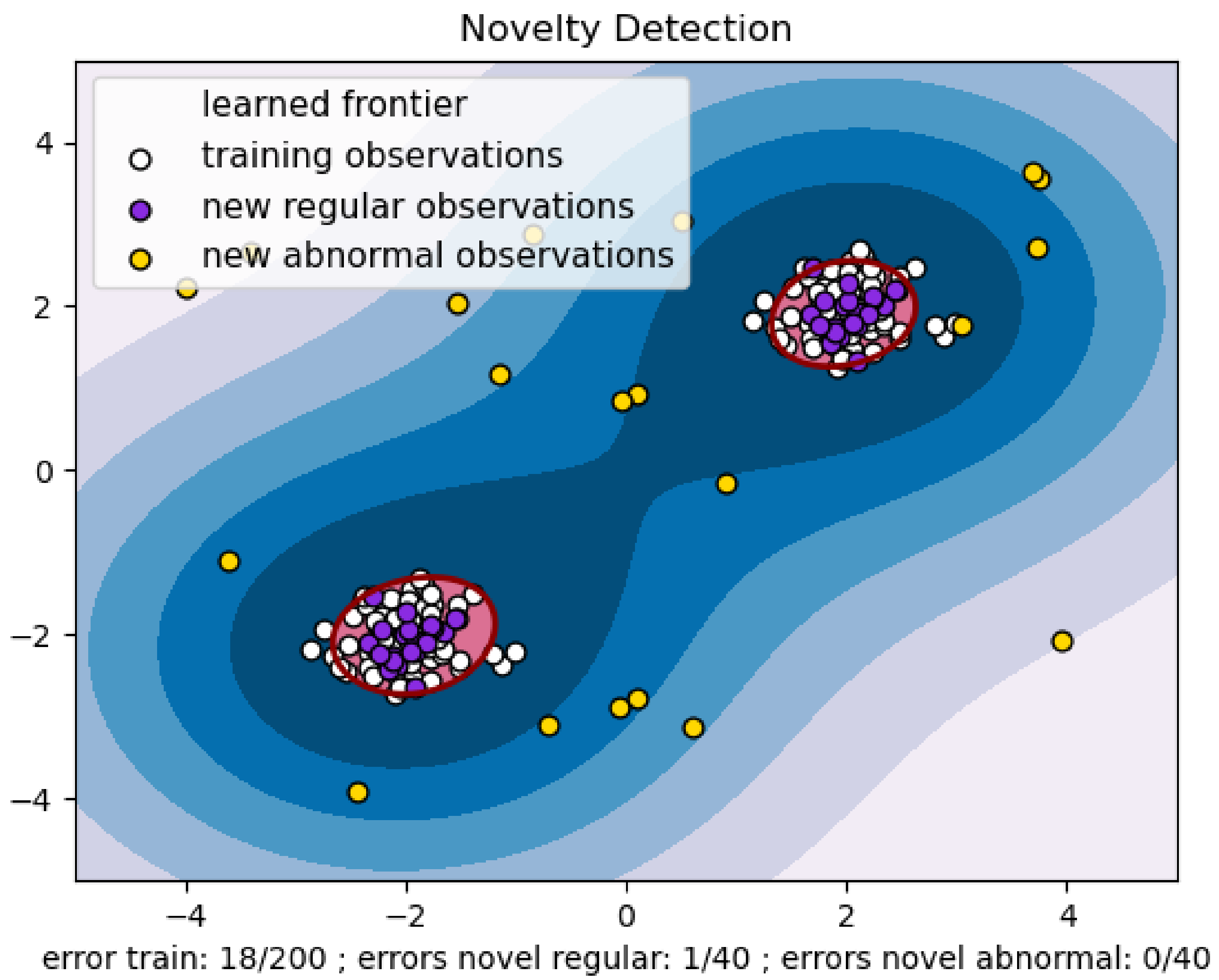
$$\min_{R,\boldsymbol{\xi},\boldsymbol{c}} \boldsymbol{R}^2 + \frac{1}{vm}\sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{m}} \xi_i \qquad R \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^m, \boldsymbol{c} \in \mathcal{H}$$

subject to:

$$\|\emptyset(x_i) - \boldsymbol{c}\|^{\mathbf{2}} \le R^{\mathbf{2}} + \xi_i,\ \ \xi_i \ge 0$$

$$0 < v \le 1$$

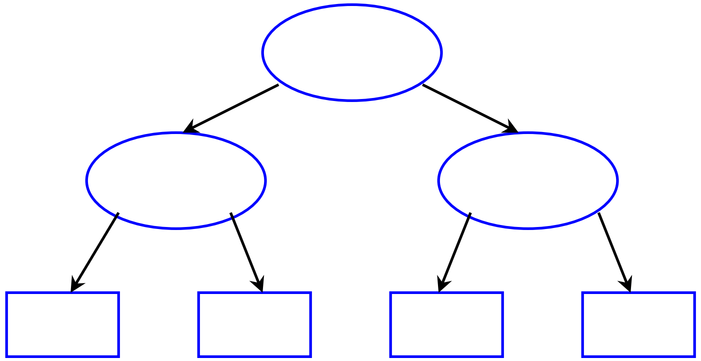


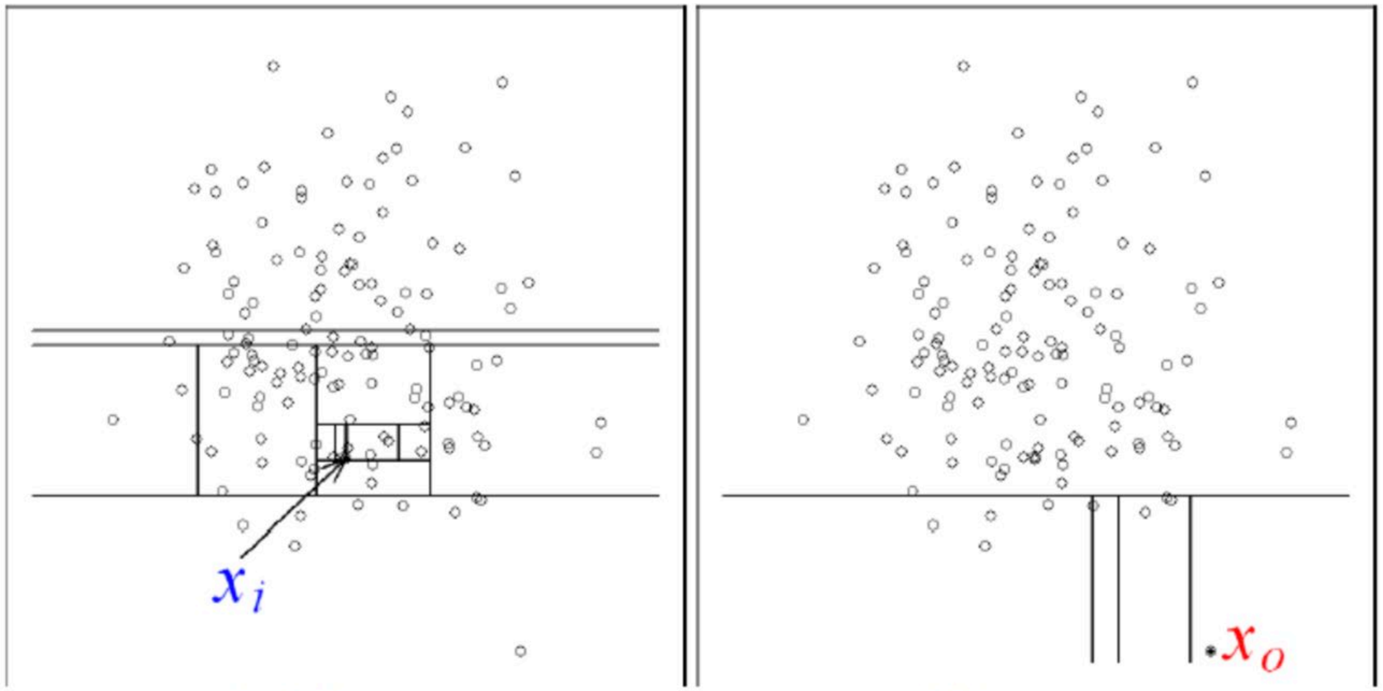


https://scikit-learn.org/stable/auto_examples/svm/plot_oneclass.html

Isolation-based outlier detection

- Outliers are few and different
- when randomly split the space into small region, an outlier is more likely to be *ISOLATED*

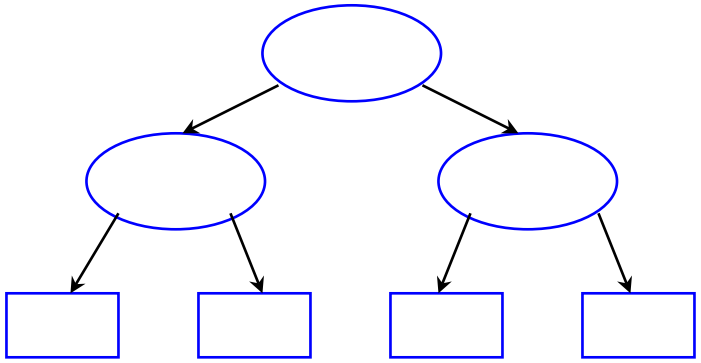




(a) Isolating x_i (b) Isolating x_o

- F. T. Liu, K. M. Ting and Z. H. Zhou， ***Isolation-based Anomaly Detection. ACM*** TKDD， 2011

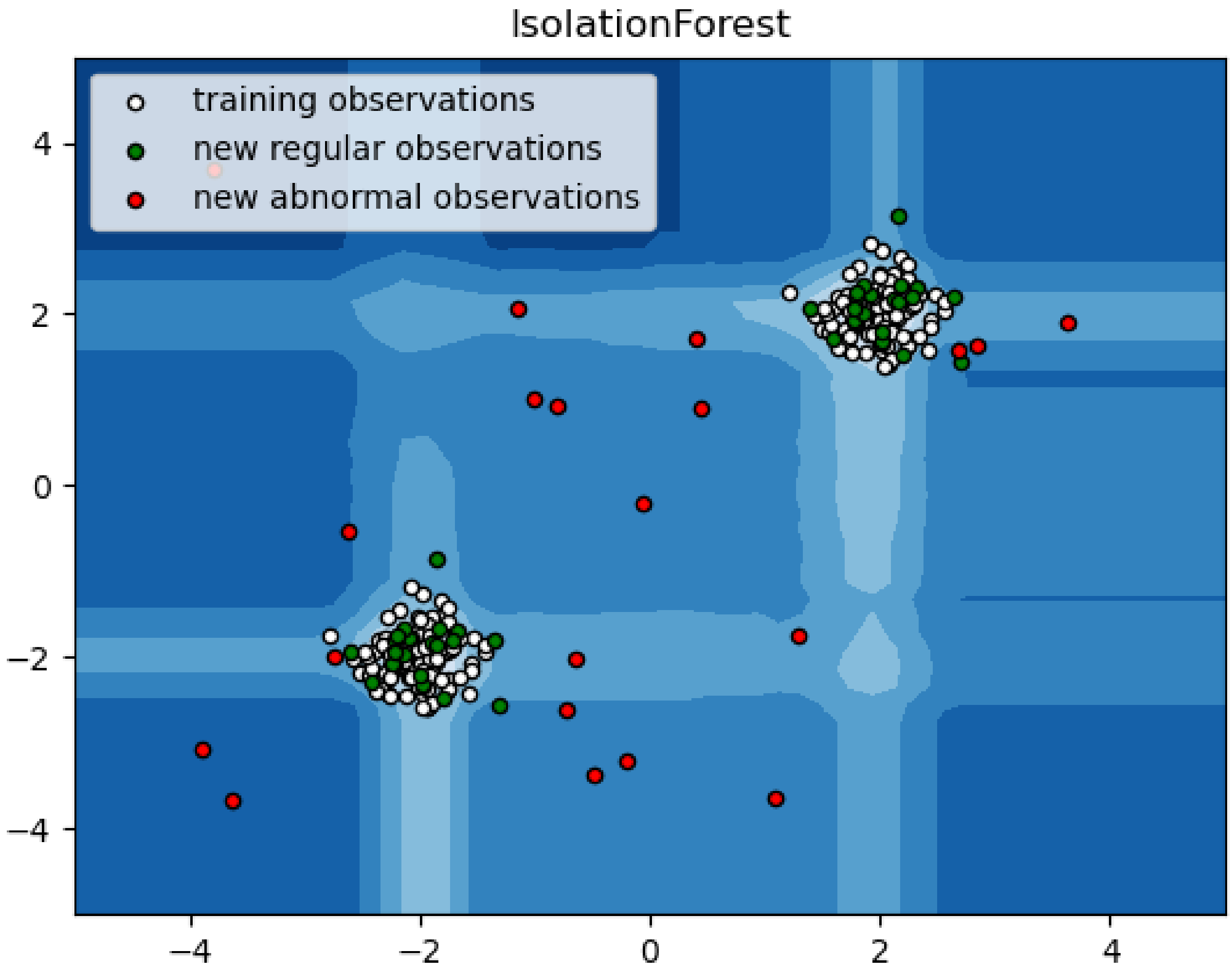
Key idea: Modeling "Isolation" using tree structure, and characterize the outlier suspiciousness with the path length from the root to the isolated object



- General steps:
  - Randomly subsample the original data
  - On each subsampled data, grow an *iTree* by
    - ✓ Randomly pick an attribute and a split value between min and max
    - ✓ Split the data into two subtrees
    - ✓ This process iterates until "isolation" is reached (no more points to be split or instances share the same value)
  - Compute outlier score by consulting the average path length from root to the isolated objects

Output of outlier detection

- Label
  - Each test instance is given a normal or anomaly label
- Score
  - Each test instance is assigned an anomaly score
    - Allows the output to be ranked
    - Requires an additional threshold parameter



https://scikit-learn.org/stable/auto_examples/ensemble/plot_isolation_forest.html#sphx-glr-auto-examples-ensemble-plot-isolation-forest-py

8.4 Reconstruction Method

Reconstruction Method: basis idea

$x_i \Rightarrow z_i \Rightarrow \hat{x}_i \qquad x_i, \hat{x}_i \in R^n \qquad z_i \in R^{d'}$

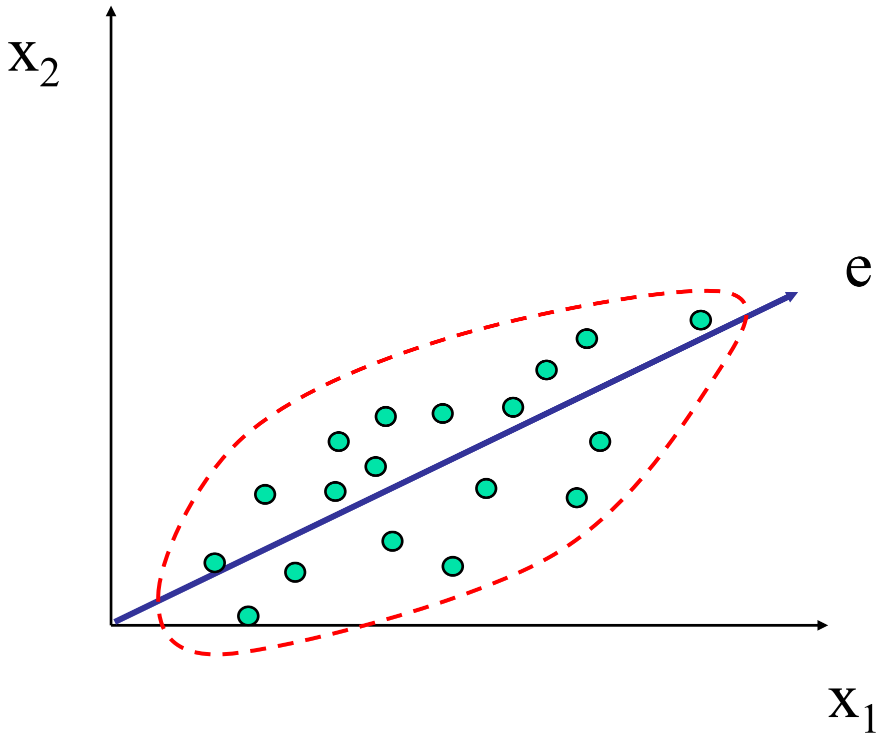
$d' \ll n$

Minimize:

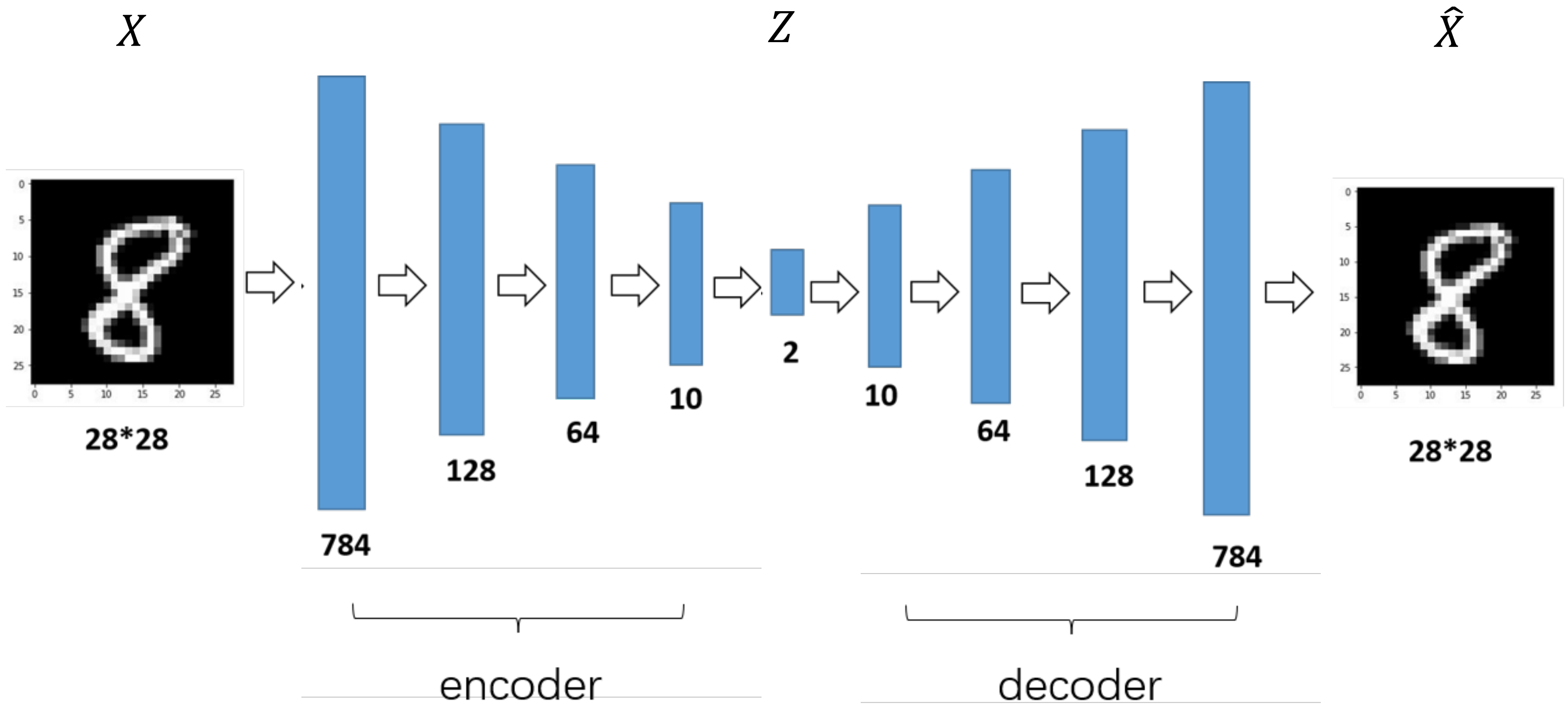
$$\boldsymbol{e} = \|x_i - \boldsymbol{\hat{x}_i}\|^2$$

Reconstruction with PCA

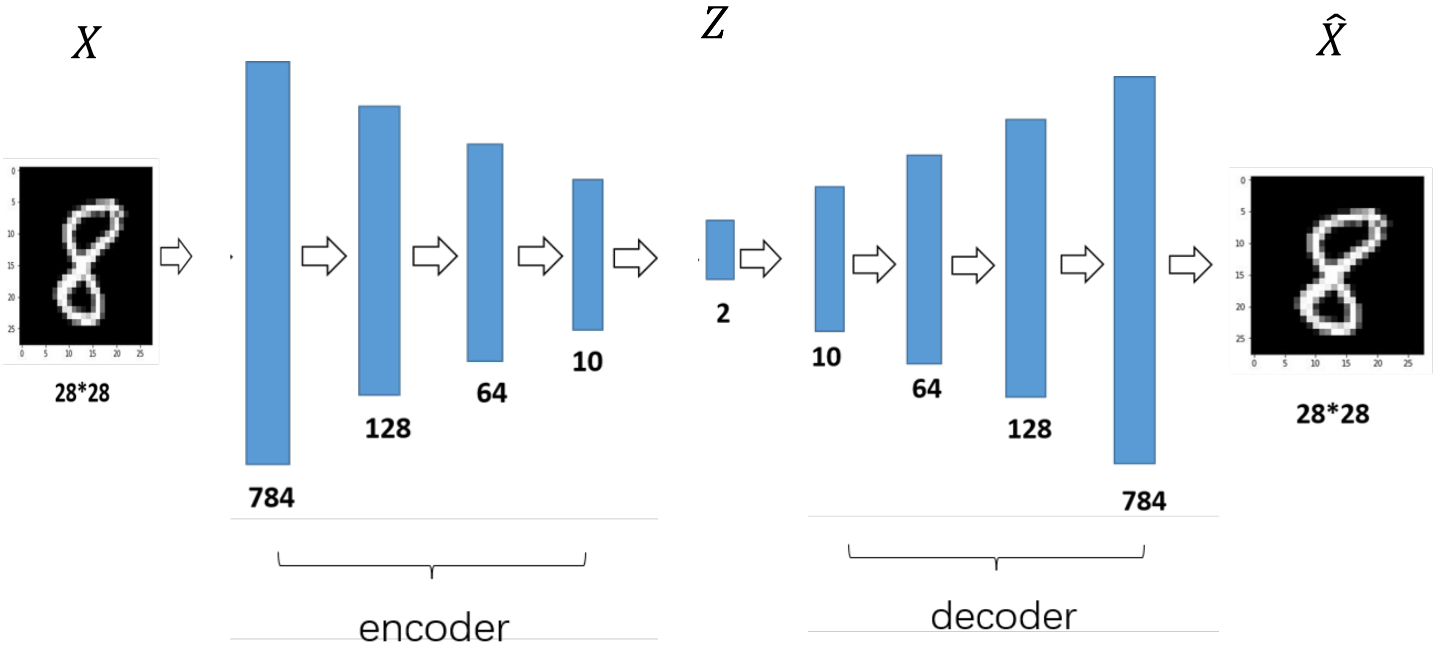
- Find a projection that captures the largest amount of variation in data.
- Project and reconstruct x_i with PCA
- Limitation of PCA method
  - **Can only model linear combination of original features**



Reconstruction with Autoencoder



Training objective



- encoder: map x to low dimensional z.

$$z = f_{\varphi}(x)$$

- decoder: reconstruct x based on z.

$$\hat{x} = g_{\theta}(z)$$

- Objective function: minimize reconstruction error.

$$L = \sum_{i=1}^{N} \|x_i - \tilde{x}_i\|_2^2$$

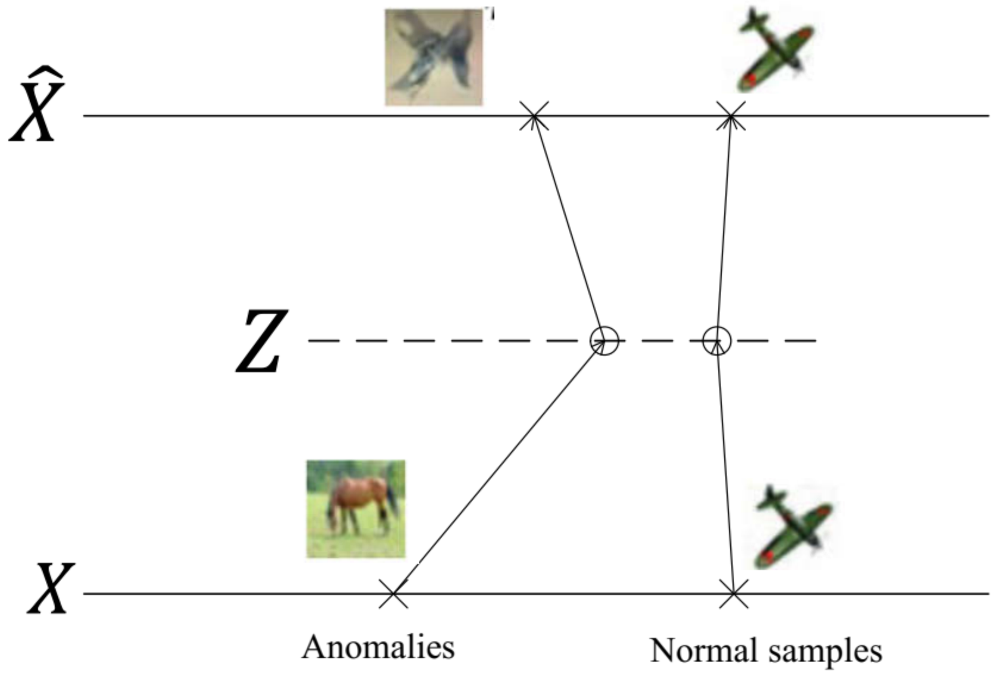
$$= \sum_{i=1}^{N} \left\|x_i - g_{\theta}(f_{\varphi}(x_i))\right\|_2^2$$

Outlier detection

- Core idea： compare reconstruction errors with given threshold to detect anomalies.

error> threshold **anomaly**

error< threshold **normal**



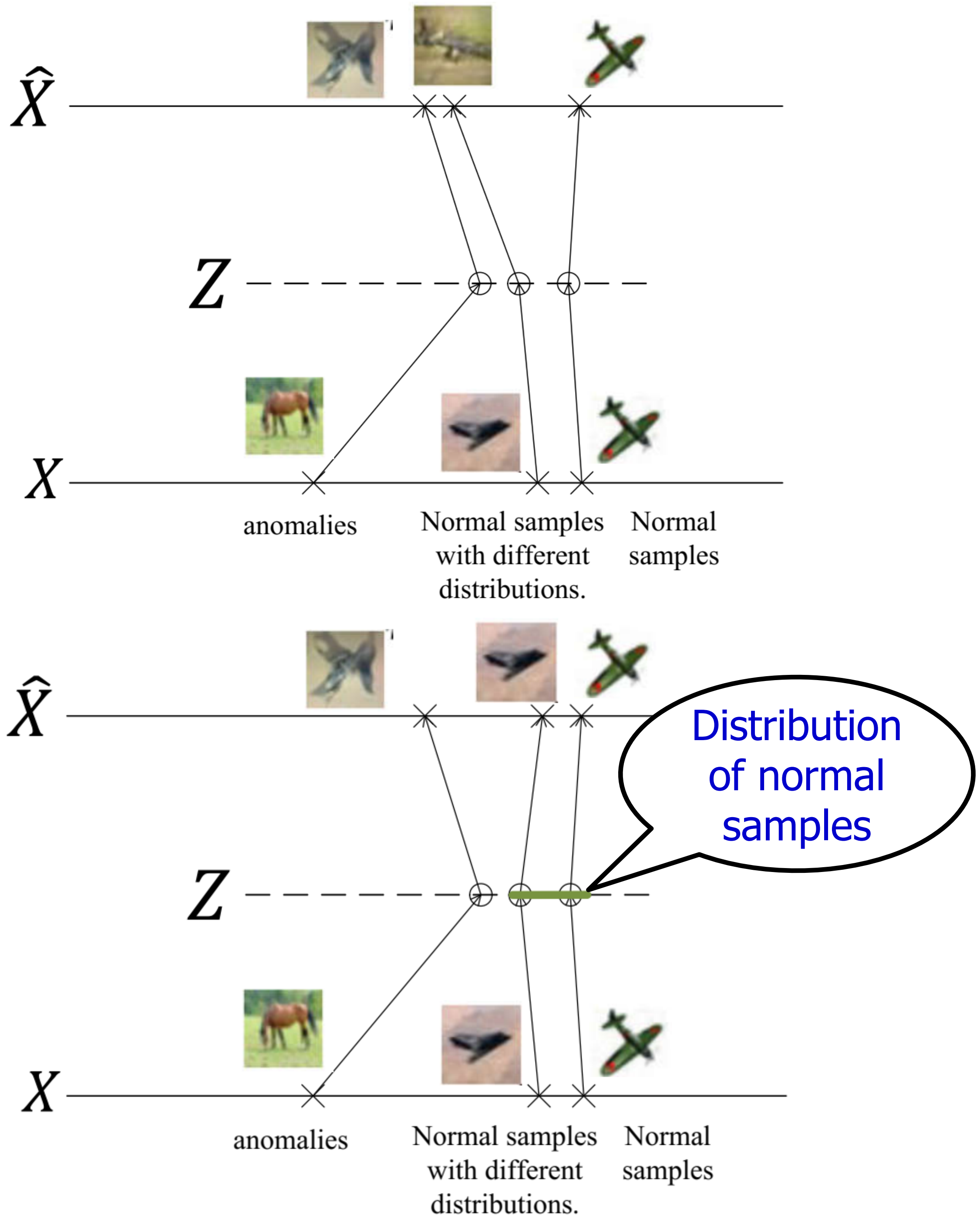
- **Shortages of AE**

$$z = f_\varphi(x)$$
$$\hat{x} = g_\theta(z)$$

➢ Learn one-on-one mapping between x and z.

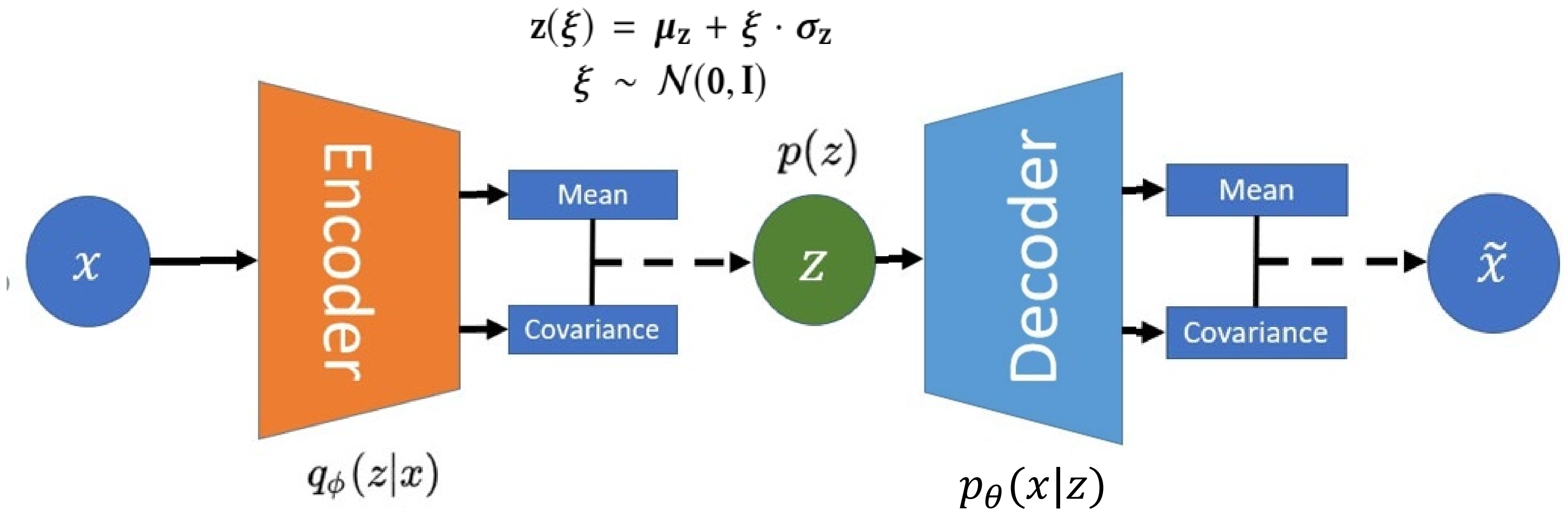
➢ Can not handle variance in normal samples. Low generalization capability.

➢ Normal samples may also be falsely judged as anomalies.



Reconstruction with Variational Autoencoder

- Core idea:
  model the parameters of distributions of z and $\tilde{x}$ rather than their values.
  ————main difference with autoencoders.



- Encoder: learn distributional parameters of z based on x.
- Decoder: learn distributional parameters of $\tilde{x}$ based on z.

Reading list

- An J, Cho S. Variational autoencoder based anomaly detection using reconstruction probability[J]. Special Lecture on IE, 2015, 2(1): 1-18.
- B. Zhou, S. Liu, B. Hooi, X. Cheng, J. Ye, Beatgan: Anomalous rhythm detection using adversarially generated time series, in: S. Kraus (Ed.), IJCAI 2019.
- Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, Kalyan Veeramachaneni. TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks. In proceedings of IEEE International Conference on Big Data, 2020

Acknowledgements

- Some text, figures and formulations are from WWW. Thanks for their sharing. If you have copyright claim please contact with me at yym@hit.edu.cn.
- This lecture is distributed for nonprofit purpose.

Thank You for Your Attention

Contact me at: yym@hit.edu.cn

Tel: 26033008, 13760196623

Address: Rm.1402, H# Building