

# Lecture 3: Novelty Detection

Pilsung Kang

School of Industrial Management Engineering

Korea University

# AGENDA

01 **Novelty Detection: Overview**

---

02 Density-based Novelty Detection

---

03 Distance/Reconstruction-based ND

---

04 Model-based Novelty Detection

---

# Machine Learning

- Definition
  - ✓ A computer program is said to **learn** from experience **E** with respect to some class of tasks **T** and performance measure **P**, if its performance at task in T, as measured by P, improves with experience E” – Mitchell (1997)

## Supervised Learning

- Goal: predict a single “target” or “outcome” variable
- Finds relations between X and Y
- Train (learn) data where target value is known
- Score data where target value is not known

## Unsupervised Learning

- Explores intrinsic characteristics
- Estimates underlying distribution
- Segment data into meaningful groups or detect patterns
- There is no target (outcome) variable to predict or classify

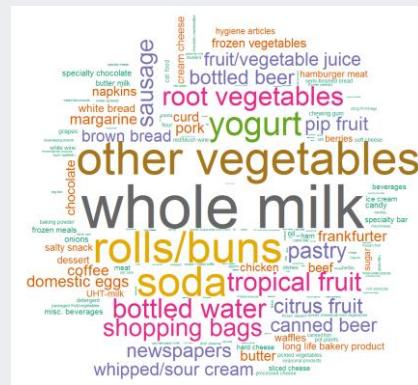
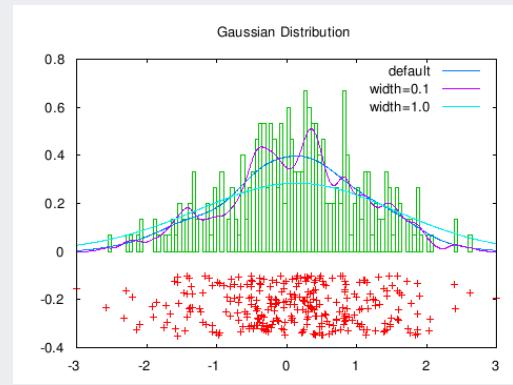
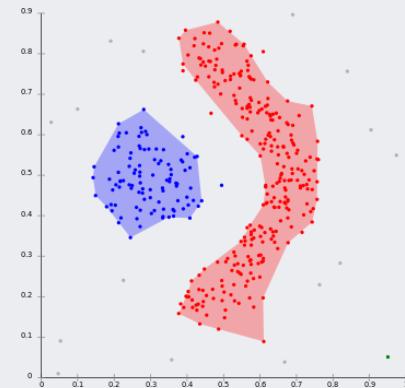
# Unsupervised Learning

A given dataset  $\mathbf{X}$

	Var. 1	Var. 2	...	Var. d
Ins. 1	..	..	...	..
Ins. 2	..	..	...	..
...	...	...	...	...
Ins. N	..	..	..	..

## Unsupervised learning

- Explores intrinsic characteristics
- Estimates underlying distribution
- Density estimation, clustering, association rule mining, network (graph) analysis, etc.



■ 이 책과 함께 구매한 도서

데이터마이닝 방법론(빅데 이터 분석을... 25,000원 [0%+1%P]	데이터마이닝 기초... 28,000원 [10%+10%P]	데이터마이닝 입문(SAS 대... 20,000원 [0%+1%P]
--	---------------------------------------	---

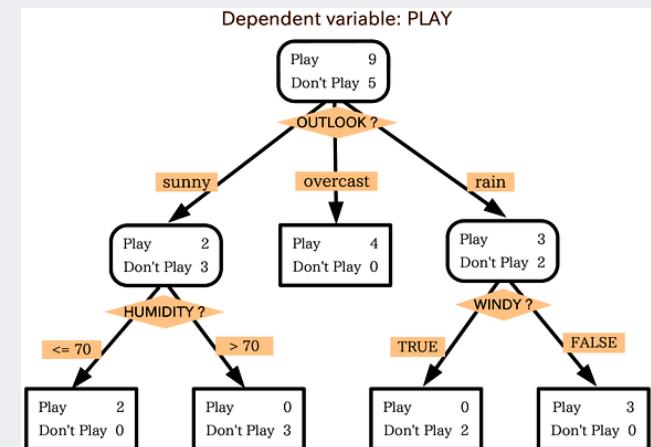


# Supervised Learning

A given dataset  $\mathbf{X}$  &  $\mathbf{Y}$

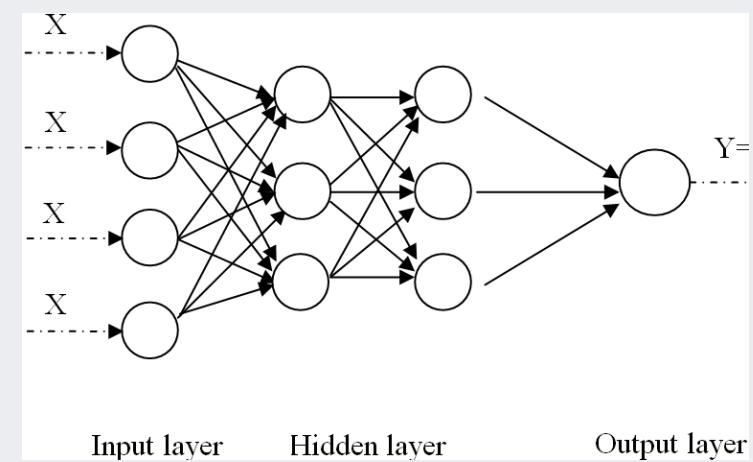
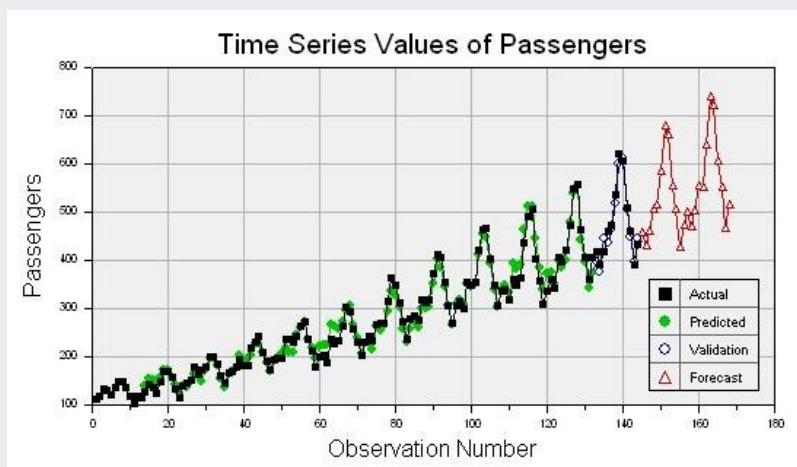
	Var. 1	Var. 2	...	Var. d	$\rightarrow$	$\mathbf{Y}$
Ins. 1	..	..	...	..		..
Ins. 2	..	..	...	..		..
...	...	...	...	...		...
Ins. N	..	..	..	..		..

$$y = f(x)$$



## Supervised learning

- Finds relations between  $\mathbf{X}$  and  $\mathbf{Y}$ : estimate the underlying function  $y = f(x)$
- Classification, regression, novelty detection



# Supervised Learning: Novelty Detection

- What is novel data (outliers)?

*“Observations that deviate so much from other observations as to arouse suspicions that they were generated by a different mechanism (Hawkins, 1980)”*  
*“Instances that their true probability density is very low (Harmeling et al., 2006)”*

- Outliers are different from noise data
  - ✓ Noise is random error or variance in a measured variable
  - ✓ Noise should be removed before outlier detection
- Outliers are interesting
  - ✓ It violates the mechanism that generates the normal data

# Supervised Learning: Novelty Detection

- Applications

- ✓ Industrial monitoring
- ✓ IT security
- ✓ Healthcare informatics
- ✓ Sensor network
- ✓ Computer vision
- ✓ Credit card fraud detection
- ✓ Telecom fraud detection
- ✓ Customer segmentation/Medical analysis

# Supervised Learning: Novelty Detection

- Classification vs. Novelty Detection

Signal Processing, Volume 99, June 2014, Pages 215-249

## A Review of Novelty Detection

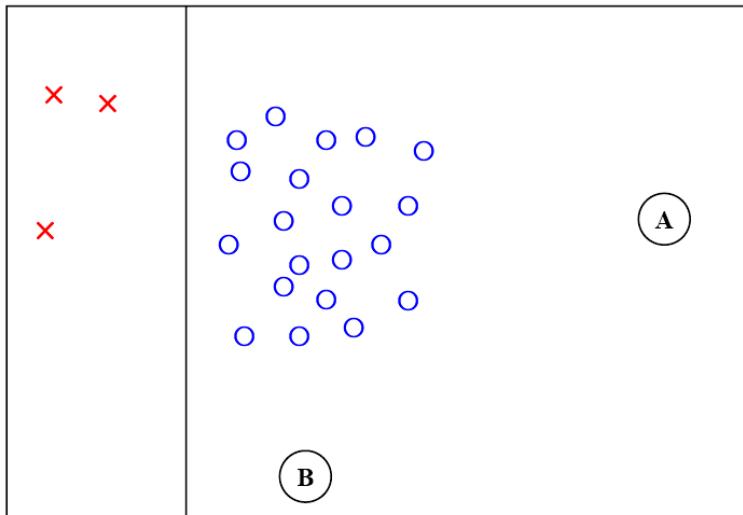
**Marco AF Pimentel, David A Clifton, Lei Clifton, Lionel Tarassenko**



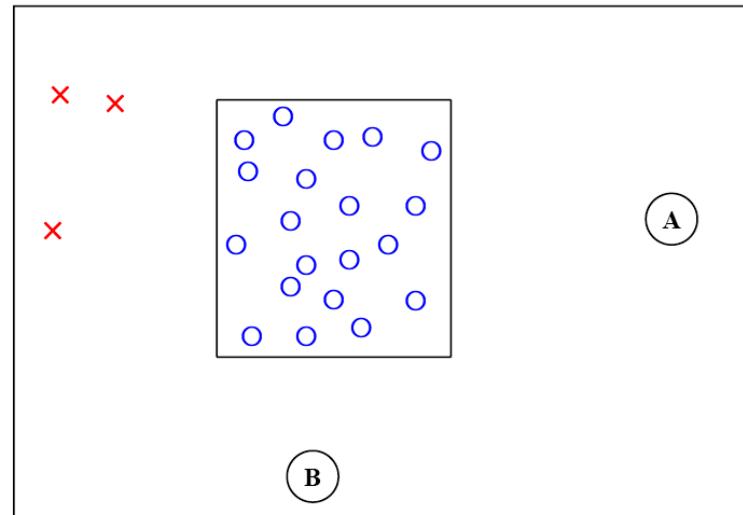
UNIVERSITY OF  
**OXFORD**

# Supervised Learning: Novelty Detection

- Classification vs. Novelty Detection

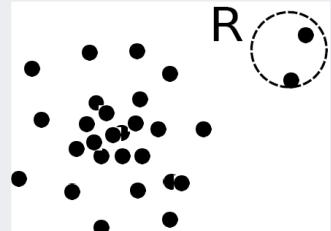
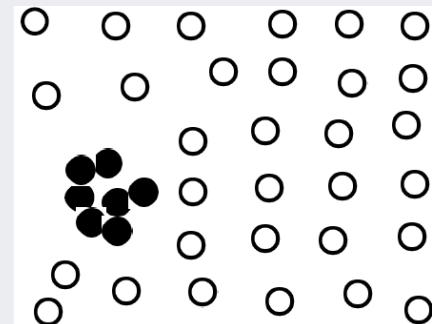


Binary classification



Novelty detection

# Type of Novel Data (Outliers)

- **Global outlier**
    - ✓ Object that significantly deviates from the rest of the data set
    - ✓ Ex) Credit card fraud detection
    - ✓ Issue: find an appropriate measurement of deviation
  - **Contextual outlier (local outlier)**
    - ✓ Object that deviates significantly based on a selected context
    - ✓ Ex) 30°C in Alaska vs. 30°C in Sahara
    - ✓ Issue: How to define or formulate meaningful context?
  - **Collective outlier**
    - ✓ A subset of data objects collectively deviate significantly from the whole data set, even if the individual data objects may not be outliers
    - ✓ Ex) Denial-of-service attack
- 
- A scatter plot with a light gray background. It contains several black dots representing data points. One point, located in the upper right quadrant, is circled with a dashed line and labeled with the letter 'R'. This illustrates a global outlier, which is an object that significantly deviates from the rest of the data set.
- 
- A scatter plot with a light gray background. The data points are represented by two types of symbols: small open circles and larger solid black circles. There is a dense cluster of small open circles in the lower-left area. In the upper-right area, there is a group of three large solid black circles. This illustrates a contextual outlier, where an object deviates significantly based on a selected context (the upper-right area).

# Challenges

- Modeling normal objects and outliers properly
  - ✓ The border between normal and outlier objects is often a gray area
- Application-specific outlier detection
  - ✓ Choice of distance measure among objects and the model of relationship among objects are often application-dependent
  - ✓ E.g., clinic data: a small deviation could be an outlier; while in marketing analysis, larger fluctuations
- Understandability
  - ✓ Understand why these are outliers: Justification of the detection
  - ✓ Specify the degree of an outlier: the unlikelihood of the object being generated by a normal mechanism

# Challenges

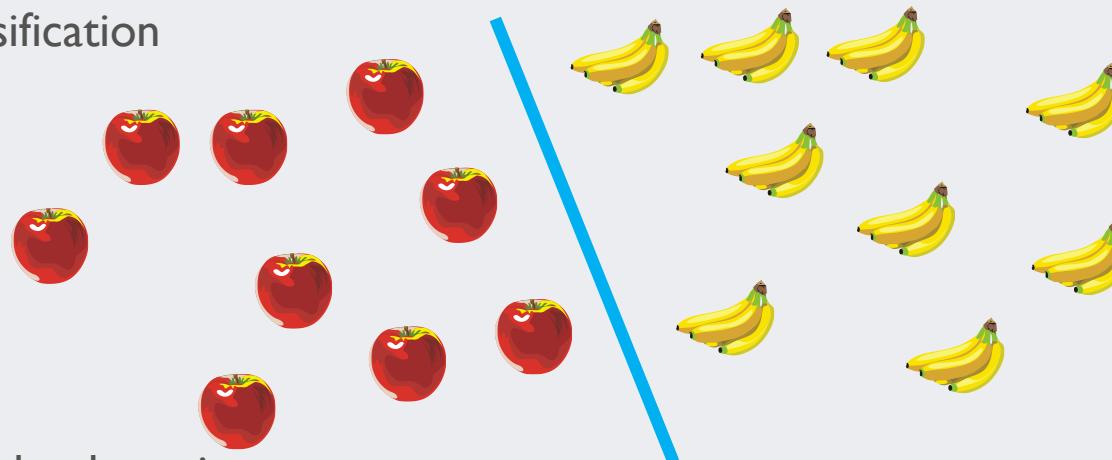
- Novelty detection actually matters

- 
- 분석목표 : 제품 원료품질 검사 데이터(X)로 제품불량(Y)을 예측
  - 데이터 분포
    - X 변수 : 재료 품질검사 데이터 560 개 항목
    - Y 범주 : 불량, 정상
    - 비중 : 정상 93%, 불량 7%
  - 분석기법
    - 지도학습 : KNN, Random Forest
    - 비지도학습 : Gaussian Mixture 모델
  - 분석과정 및 결과
    - 1) 전체 데이터로 학습
      - Accuracy 93%
      - 문제점 : 불량데이터에 대한 예측성능이 매우 낮음 (Precision 0.38, Recall 0.01, f1-score 0.02)
    - 2) 정상, 불량 학습데이터 비중을 맞춤 (1 : 1)
      - Accuracy 66%
      - 문제점 : 불량데이터 예측성능은 향상되었으나 (Precision 0.38, Recall 0.71), 전체 Accuracy 낮아짐 (66%)
    - 3) 이상치 탐지 분석 (가우시안 혼합모델)
      - 비정상 데이터에 대한 예측성능이 지도학습과 유사한 수준으로 낮음
    - 4) PolyNomial 방법
      - X 변수를 증가시키는 방법 : 분류분석이고, 이미 X변수가 많아서 시도하지 않음
  - 문의사항
    - 판단기준 : 편중된 데이터에 대해서 어느 정도의 수치가 나왔을 때 연관성이 있다고 판단해야 할지 판단기준은?
    - 분석방향 : 편중된 데이터에 대해서 불량(이상치)을 예측할 수 있도록 모델학습 시, 시도해 볼만한 방법은?

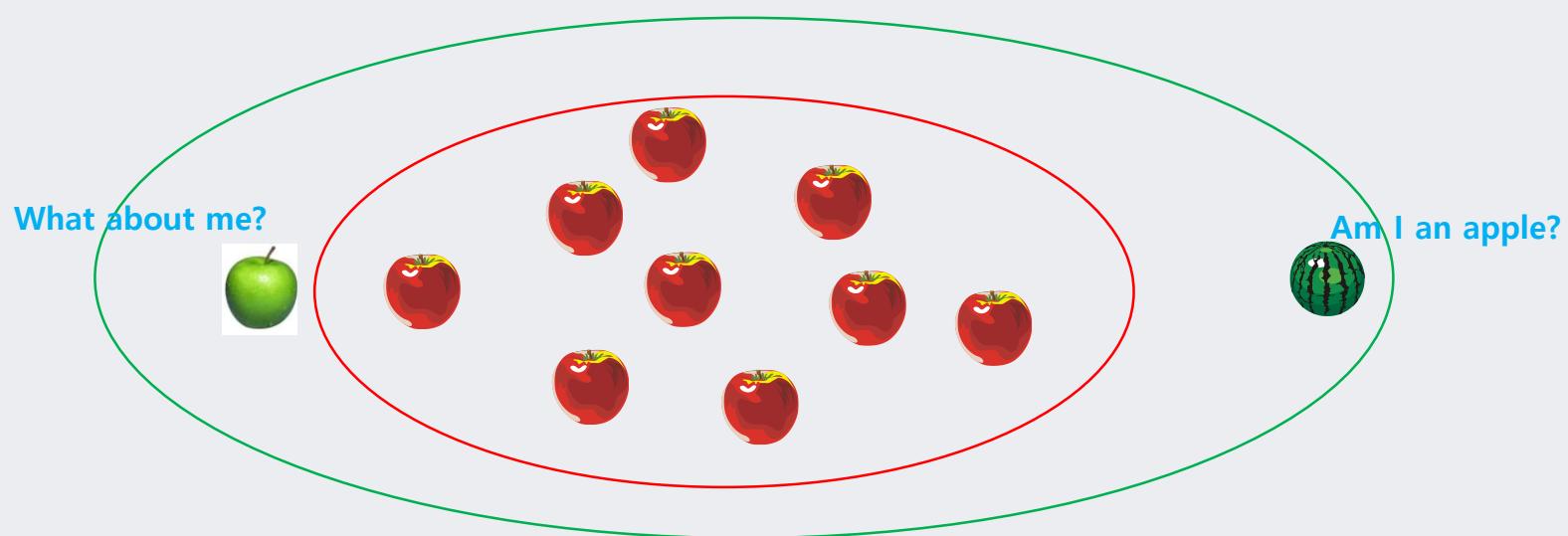
# Supervised Learning: Novelty Detection

- The way by which the classification and novelty detection learns from data

✓ Classification

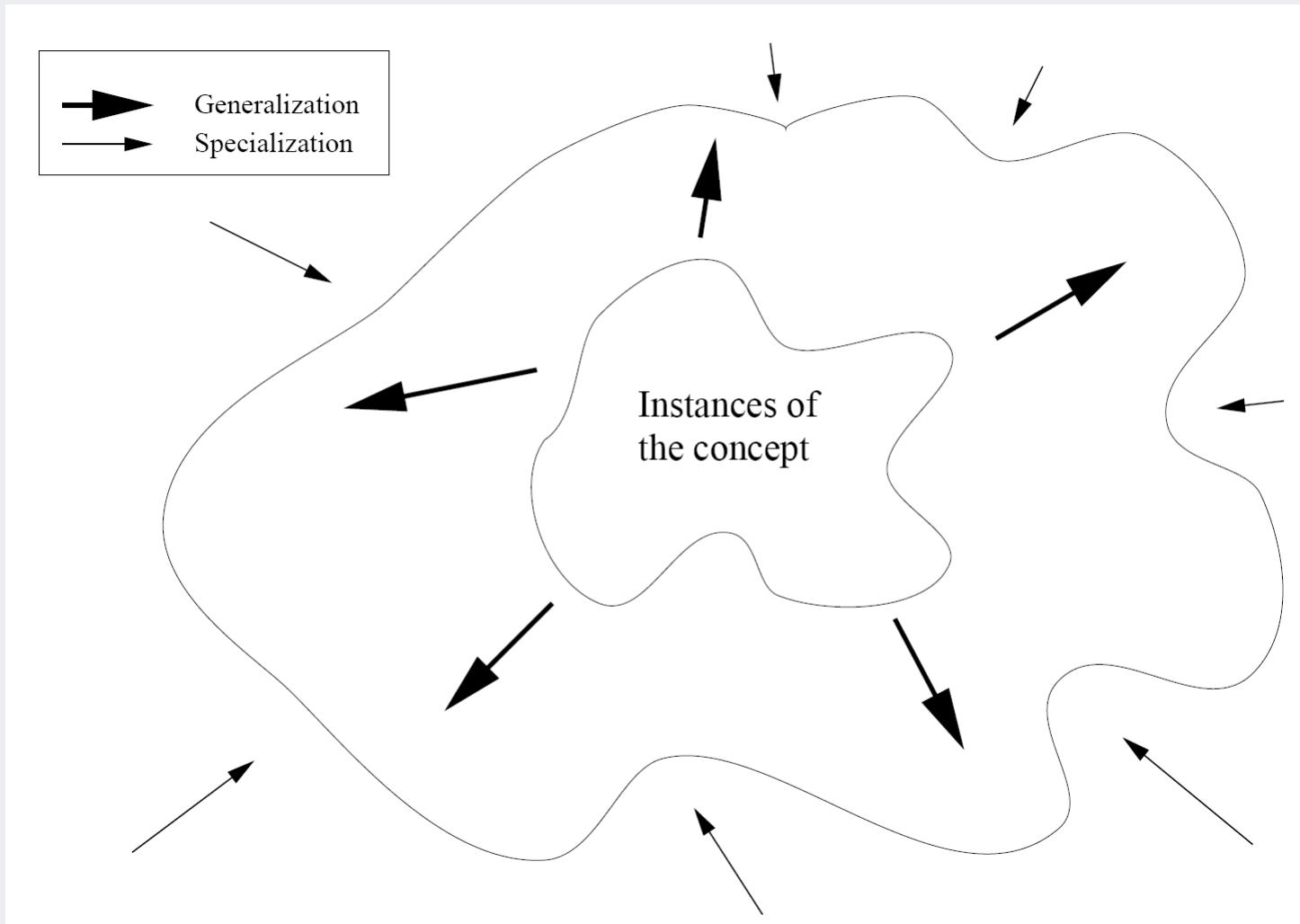


✓ Novelty detection



# Supervised Learning: Novelty Detection

- Generalization vs. Specialization

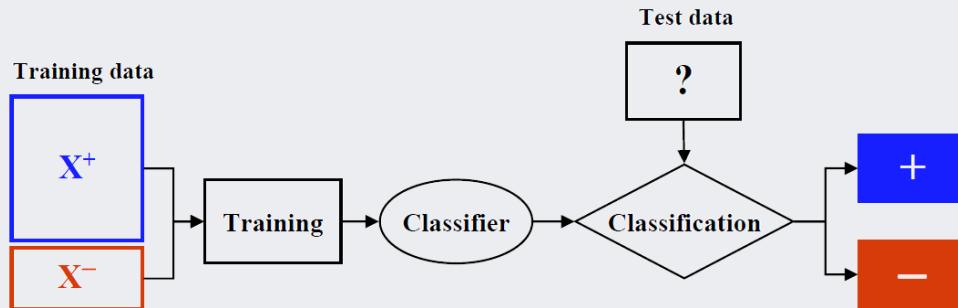


# ND Approach

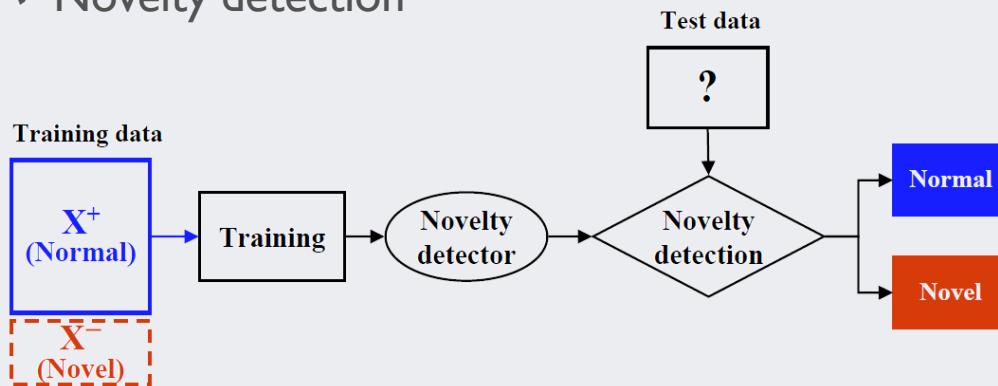
- Assumption

- ✓ There are considerably more “normal” observations than “abnormal” observations in the data

- ✓ Classification



- ✓ Novelty detection



# Performance Measures

- Performance Measures

- ✓ Confusion matrix for novelty detection

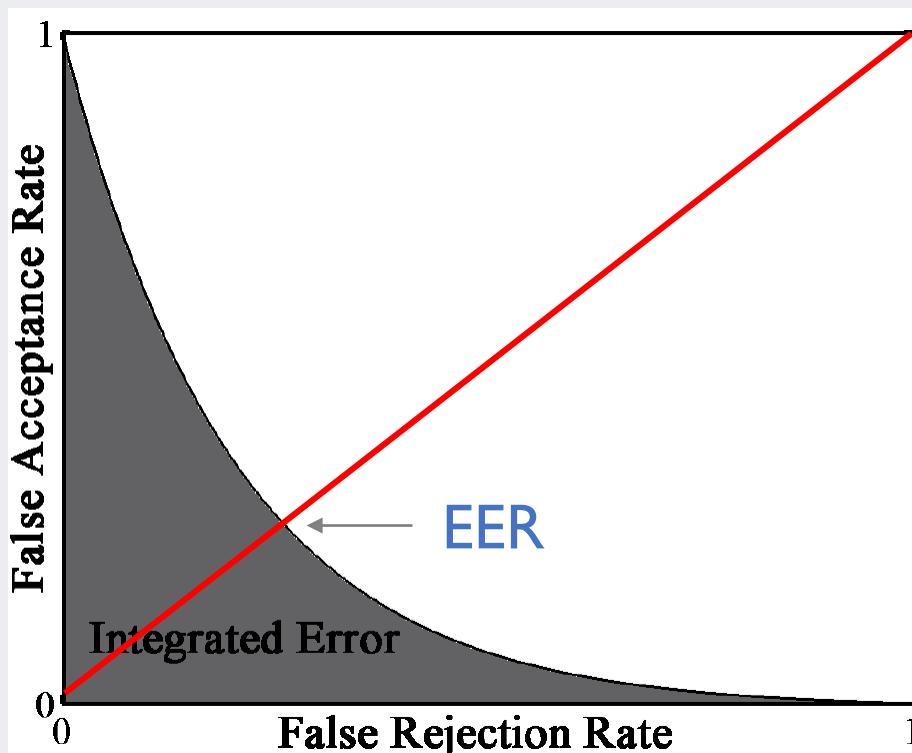
		Predicted class	
		Novel	Normal
Actual class	Novel	A	B
	Normal	C	D

- ✓ Performance measures when the cut-off (threshold) is set

Metric	Description
<b>Detection Rate</b>	$(\text{Identified as novel}) / (\text{Actually novel}) = A / (A+B)$
<b>False Rejection Rate (FRR)</b>	$(\text{Rejected as novel}) / (\text{Actually normal}) = C / (C+D)$
<b>False Acceptance Rate (FAR)</b>	$(\text{Accepted as normal}) / (\text{Actually novel}) = B / (A+B)$

# Performance Measures

- To evaluate an intrinsic performance of novelty detection algorithms
  - ✓ Equal error rate (EER): Error rate where the FAR and FRR are the same
  - ✓ Integrated Error (IE): the area under the FRR-FAR curve
    - AUROC for classification: the higher the better
    - IE for novelty detection: the lower the better



# AGENDA

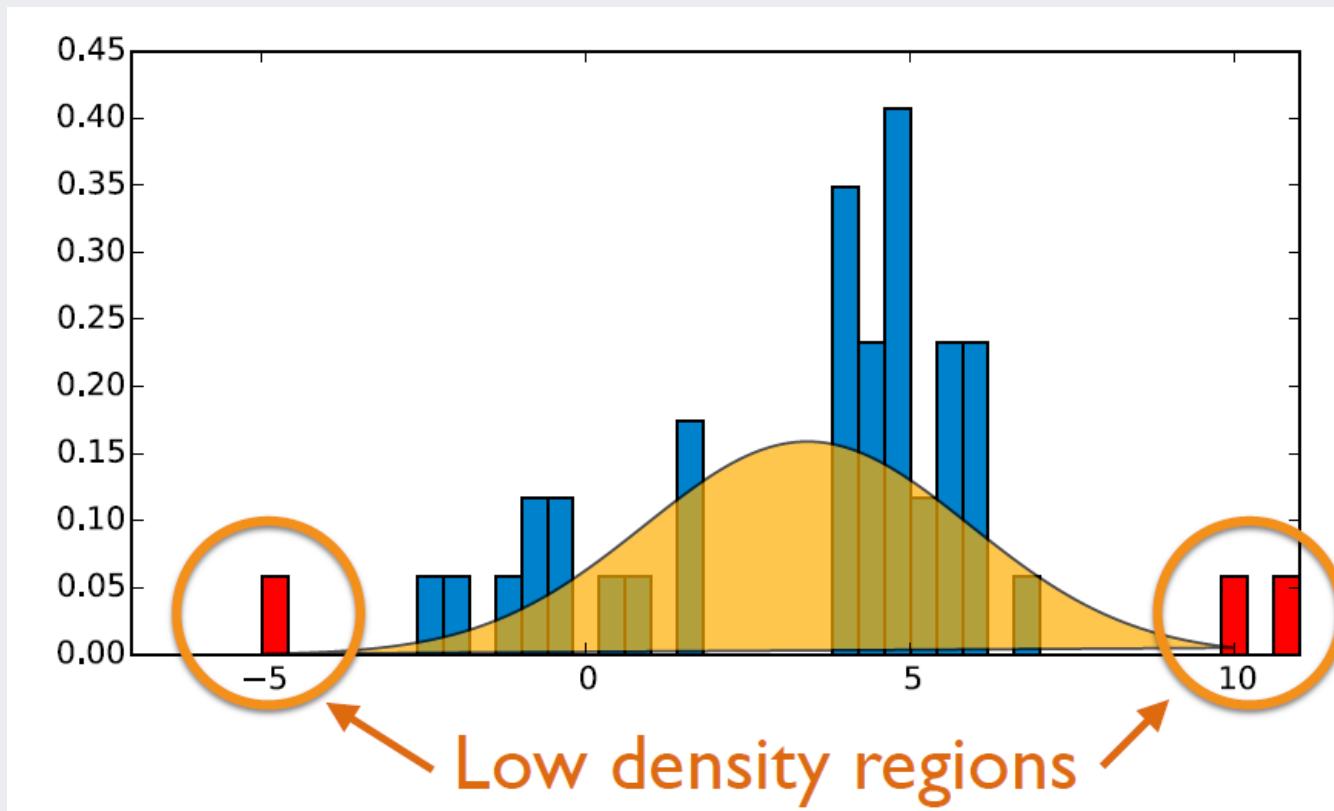
- 01 Novelty Detection: Overview
- 02 Density-based Novelty Detection
- 03 Distance/Reconstruction-based ND
- 04 Model-based Novelty Detection

# Density-based Novelty Detection

Gramfort (2006)

- Purpose

- ✓ Estimate the data-driven density function
- ✓ If a new instance has a low probability according the trained density function, it will be identified as novel

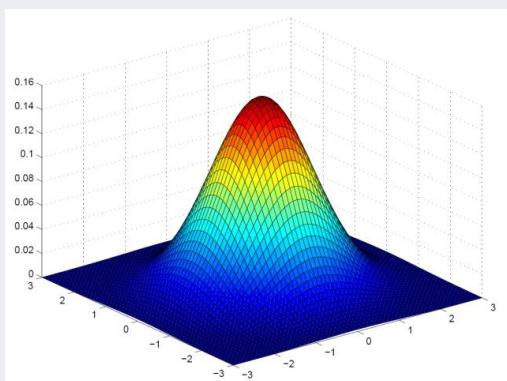


# Density-based Novelty Detection

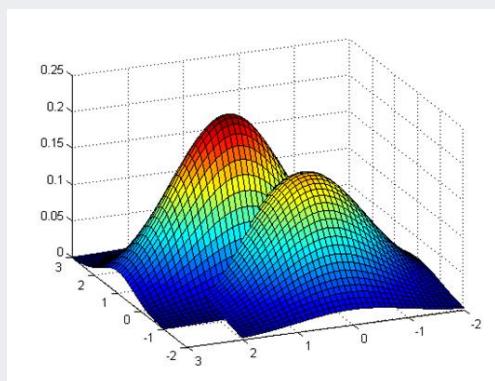
- Purpose

- ✓ Estimate the data-driven density function
- ✓ If a new instance has a low probability according the trained density function, it will be identified as novel

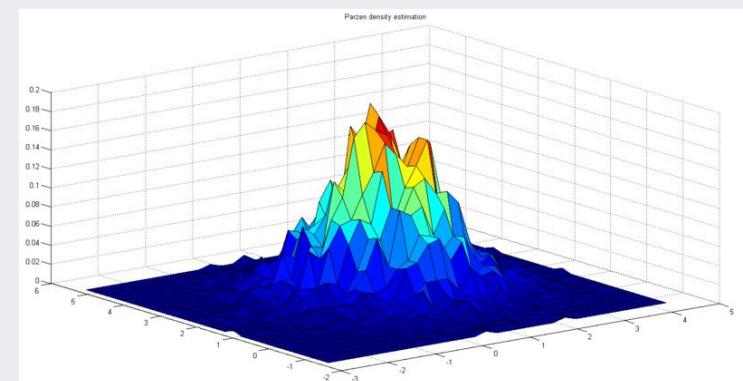
Gaussian Density Estimation



Mixture of Gaussian Density Estimation



Kernel Density Estimation



Number of modals  
= 1

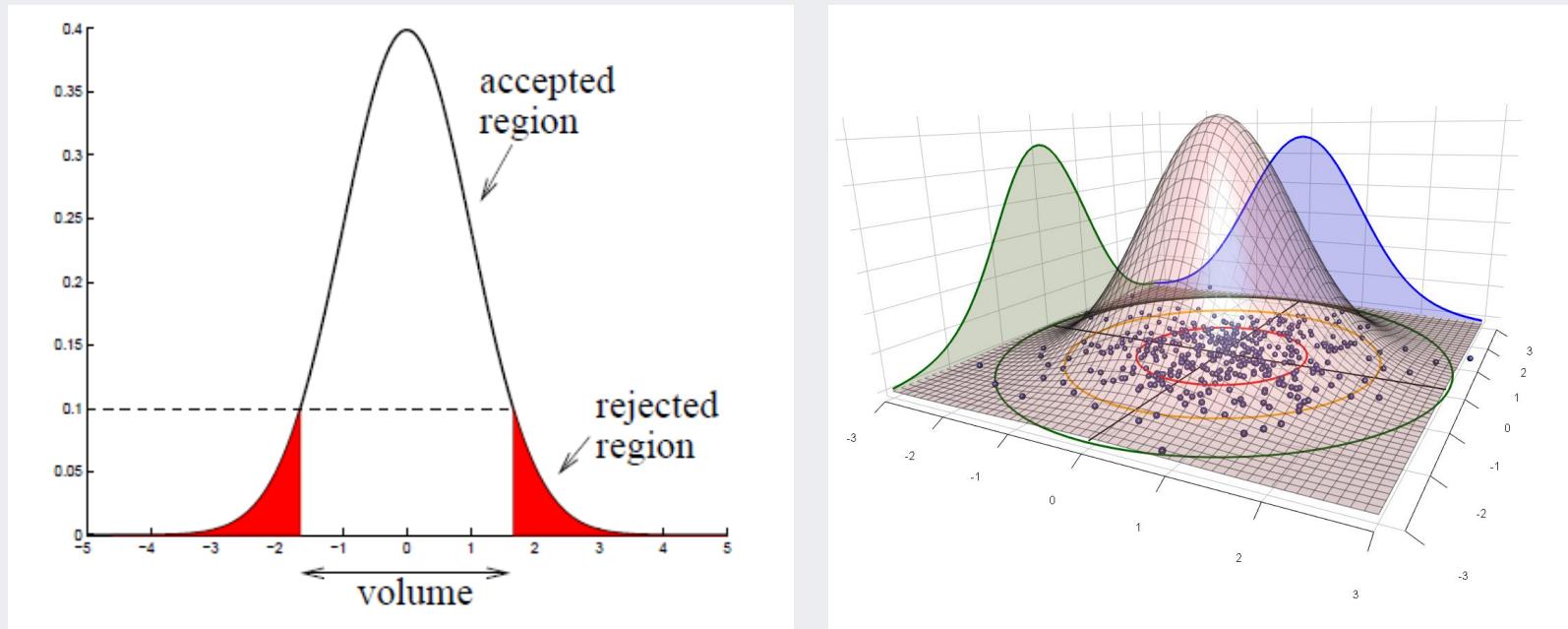
| <  
Number of modals  
< Number of instances

Number of modals  
= Number of instances

# Gaussian Density Estimation

- Gaussian Density Estimation

- ✓ Assume that the observed data are drawn from a Gaussian distribution



$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp \left[ \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{\mathbf{x}_i \in \mathbf{X}^+} \mathbf{x}_i \quad (\text{mean vector}), \quad \Sigma = \frac{1}{n} \sum_{\mathbf{x}_i \in \mathbf{X}^+} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T \quad (\text{covariance matrix})$$

# Gaussian Density Estimation

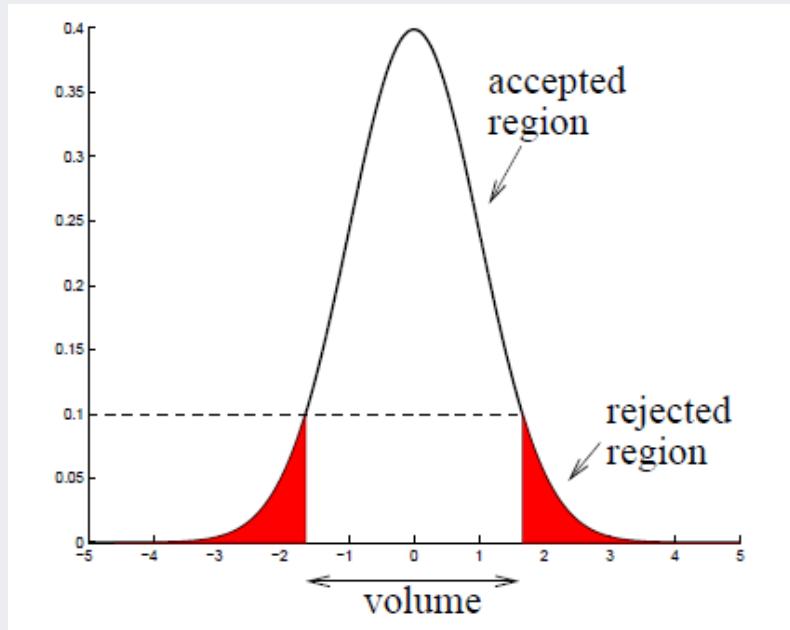
- Gaussian Density Estimation

- ✓ Advantages

- Insensitive to scaling of the data

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right]$$

- Possible to compute analytically the optimal threshold



# Gaussian Density Estimation

- Gaussian Density Estimation

- ✓ Parameter estimation:  $\mu$  and  $\sigma^2$

- ✓ For one-dimensional data,

$$L = \prod_{i=1}^N P(x_i | \mu, \sigma^2) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right)$$

$$\log L = -\frac{1}{2} \sum_{i=1}^N \frac{(x_i - \mu)^2}{\sigma^2} - \frac{N}{2} \log(2\pi\sigma^2)$$

# Gaussian Density Estimation

- Gaussian Density Estimation

✓ Maximum likelihood estimation, let's set  $\gamma = 1/\sigma^2$

$$\log L = -\frac{1}{2} \sum_{i=1}^N \gamma(x_i - \mu)^2 - \frac{N}{2} \log(2\pi) + \frac{N}{2} \log(\gamma)$$

$$\frac{\partial \log L}{\partial \mu} = \gamma \sum_{i=1}^N (x_i - \mu) = 0 \rightarrow \mu = \frac{1}{N} \sum_{i=1}^N x_i$$

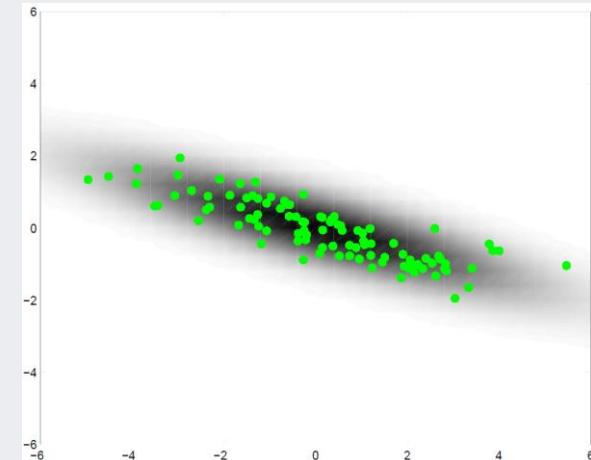
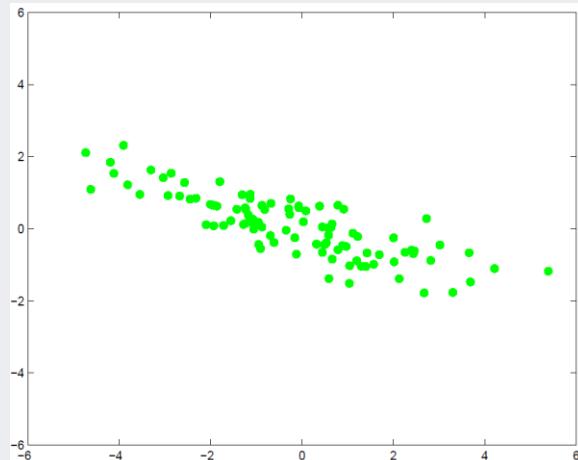
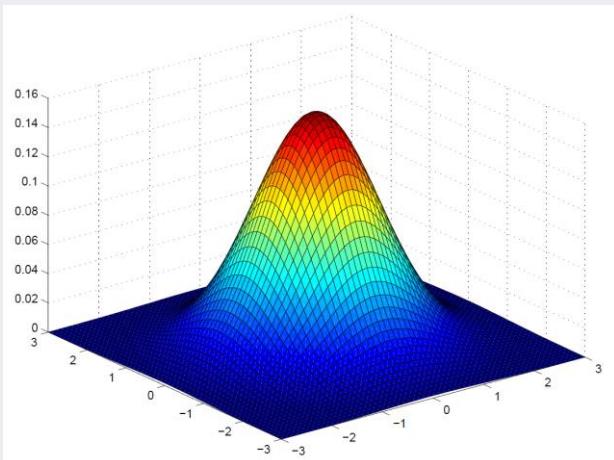
$$\frac{\partial \log L}{\partial \gamma} = -\frac{1}{2} \sum_{i=1}^N (x_i - \mu)^2 + \frac{N}{2\gamma} = 0 \rightarrow \sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$$

# Gaussian Density Estimation

- Gaussian Density Estimation

✓ In general,

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i, \quad \boldsymbol{\Sigma} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$$



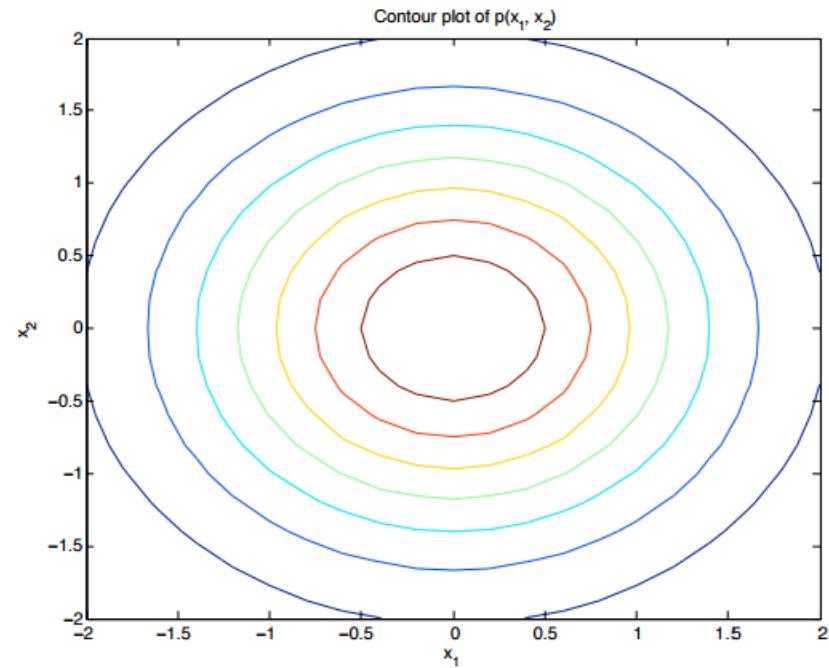
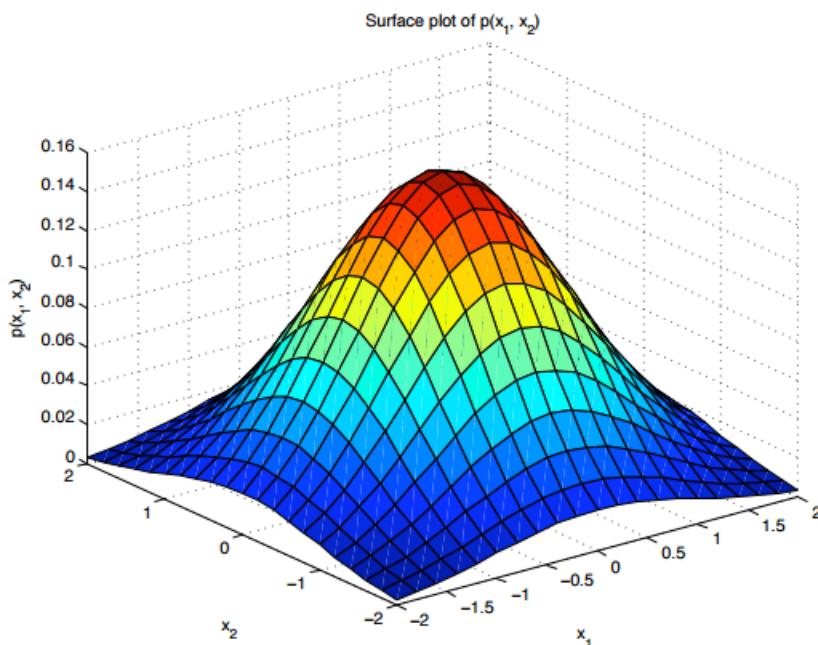
# Gaussian Density Estimation

- Gaussian Density Estimation

- ✓ The shape of Gaussian distribution according to the Covariance matrix type

Spherical

$$\Sigma = \sigma^2 \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$



(a) Spherical Gaussian (diagonal covariance, equal variances)

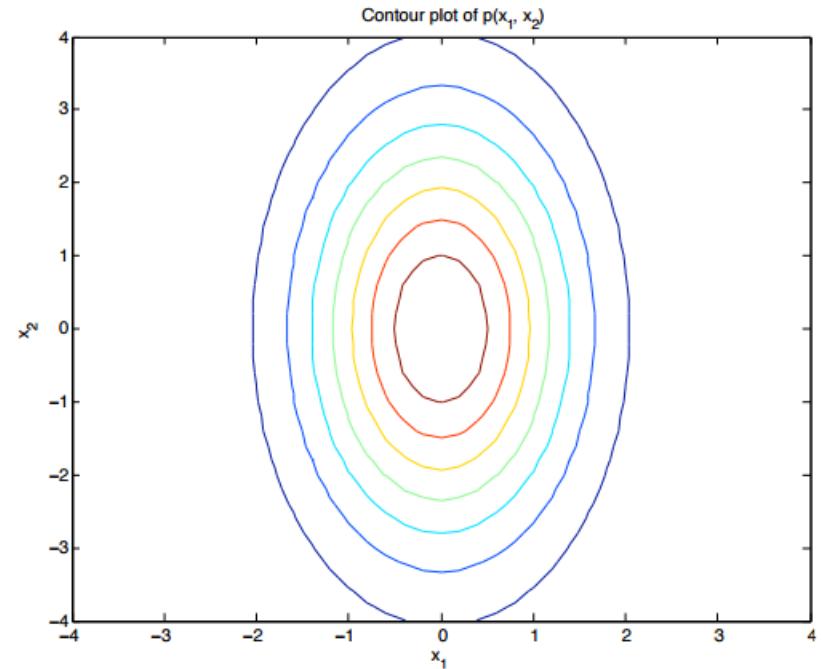
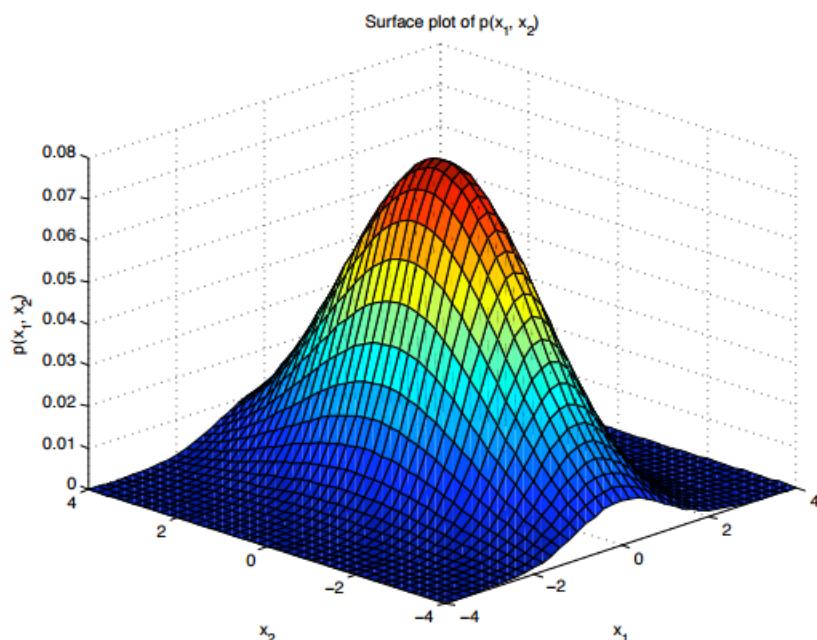
# Gaussian Density Estimation

- Gaussian Density Estimation

- ✓ The shape of Gaussian distribution according to the Covariance matrix type

Diagonal

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_d^2 \end{bmatrix}$$



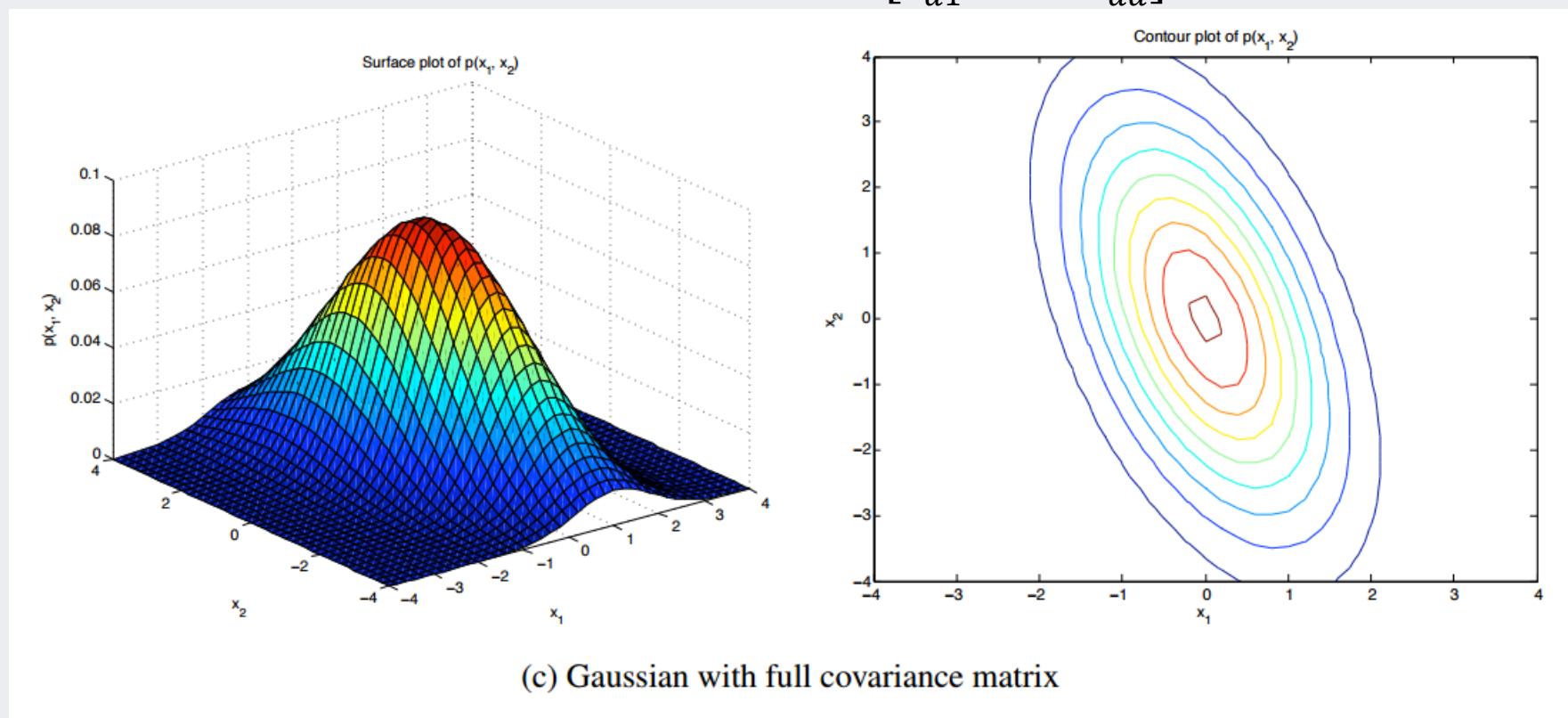
(b) Gaussian with diagonal covariance matrix

# Gaussian Density Estimation

- Gaussian Density Estimation

- ✓ The shape of Gaussian distribution according to the Covariance matrix type

Full       $\Sigma = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1d} \\ \vdots & \ddots & \vdots \\ \sigma_{d1} & \cdots & \sigma_{dd} \end{bmatrix}$



# Mixture of Gaussian Density Estimation

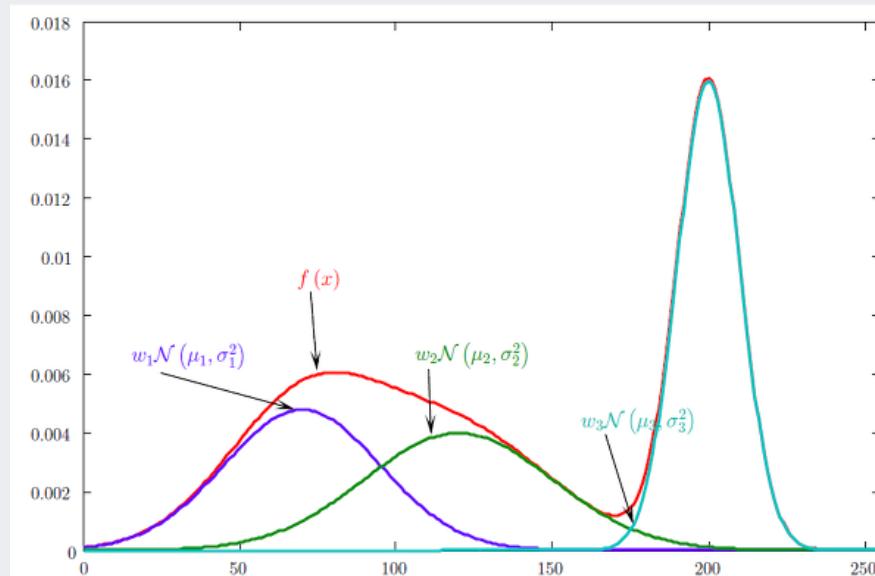
- Mixture of Gaussian (MoG) Density Estimation

- ✓ Gaussian Density Estimation

- assumes a very strong model of the data: unimodal and convex

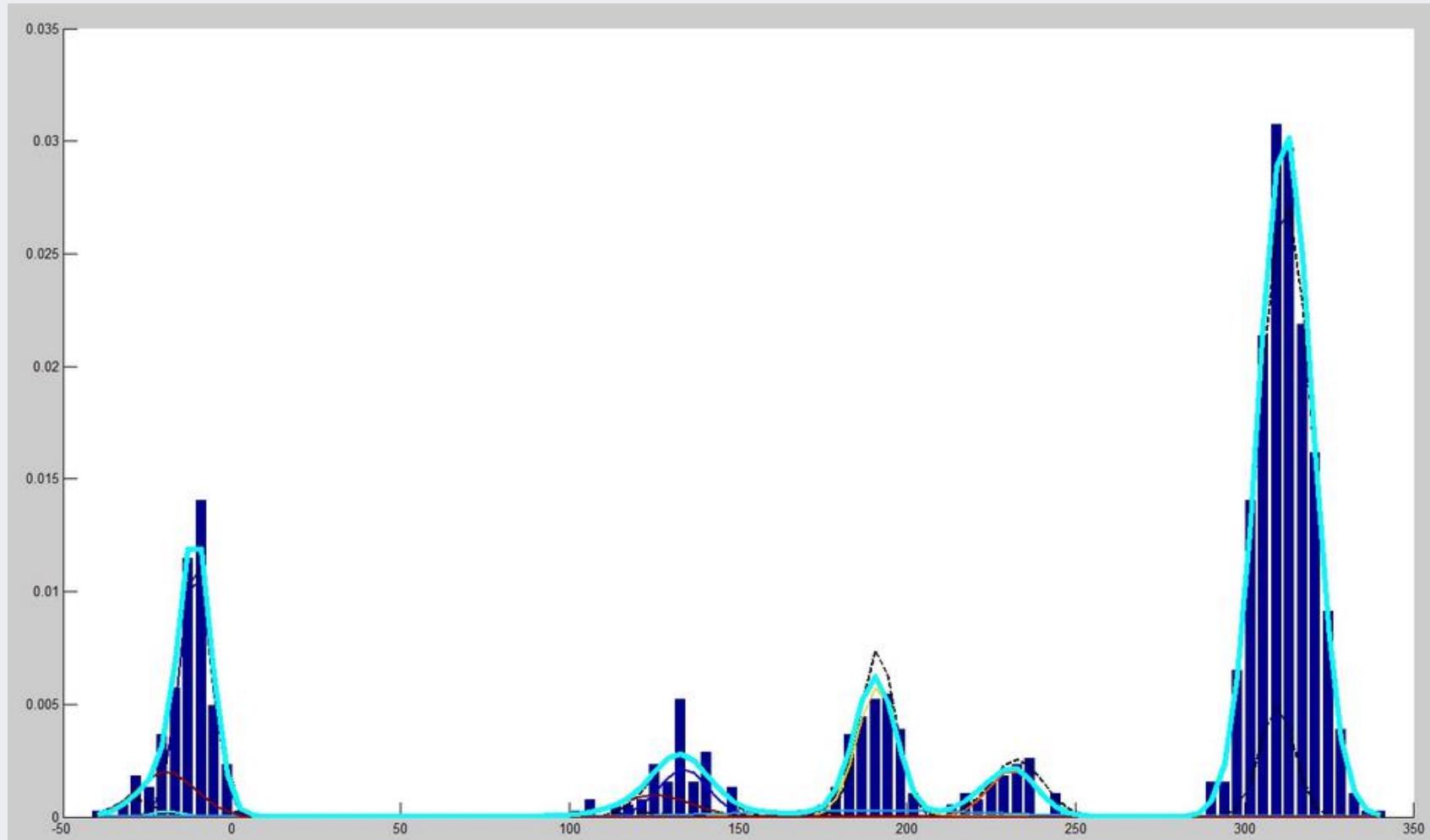
- ✓ MoG

- an extension of Gaussian that allows multi-modal distribution
    - a linear combination of normal distributions
    - Has a smaller bias than the single Gaussian distribution, but requires far more data for training



# Mixture of Gaussian Density Estimation

- MoG example



# Mixture of Gaussian Density Estimation

- Components of MoG

- ✓ Probability of an instance belonging to the normal class

$$p(\mathbf{x}|\lambda) = \sum_{m=1}^M w_m g(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

- ✓ Distribution of each Gaussian model

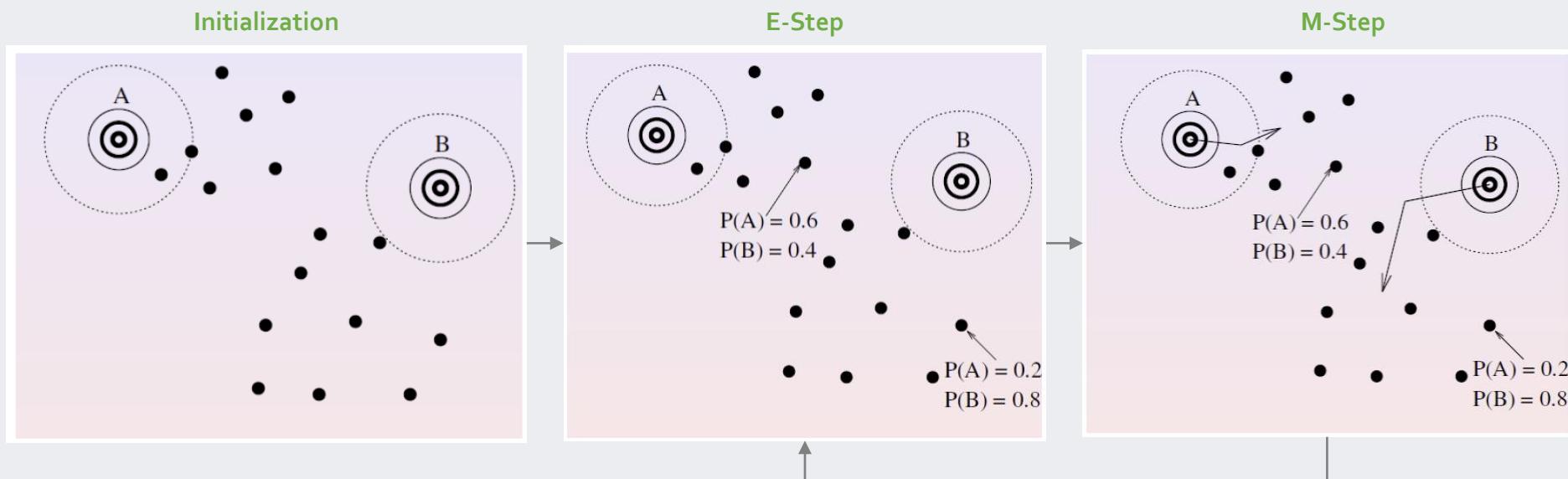
$$g(\mathbf{x}|\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_m|^{1/2}} \exp \left[ \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_m)^T \boldsymbol{\Sigma}_m^{-1} (\mathbf{x} - \boldsymbol{\mu}_m) \right]$$

$$\lambda = \{w_m, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}, \quad m = 1, \dots, M$$

# Mixture of Gaussian Density Estimation

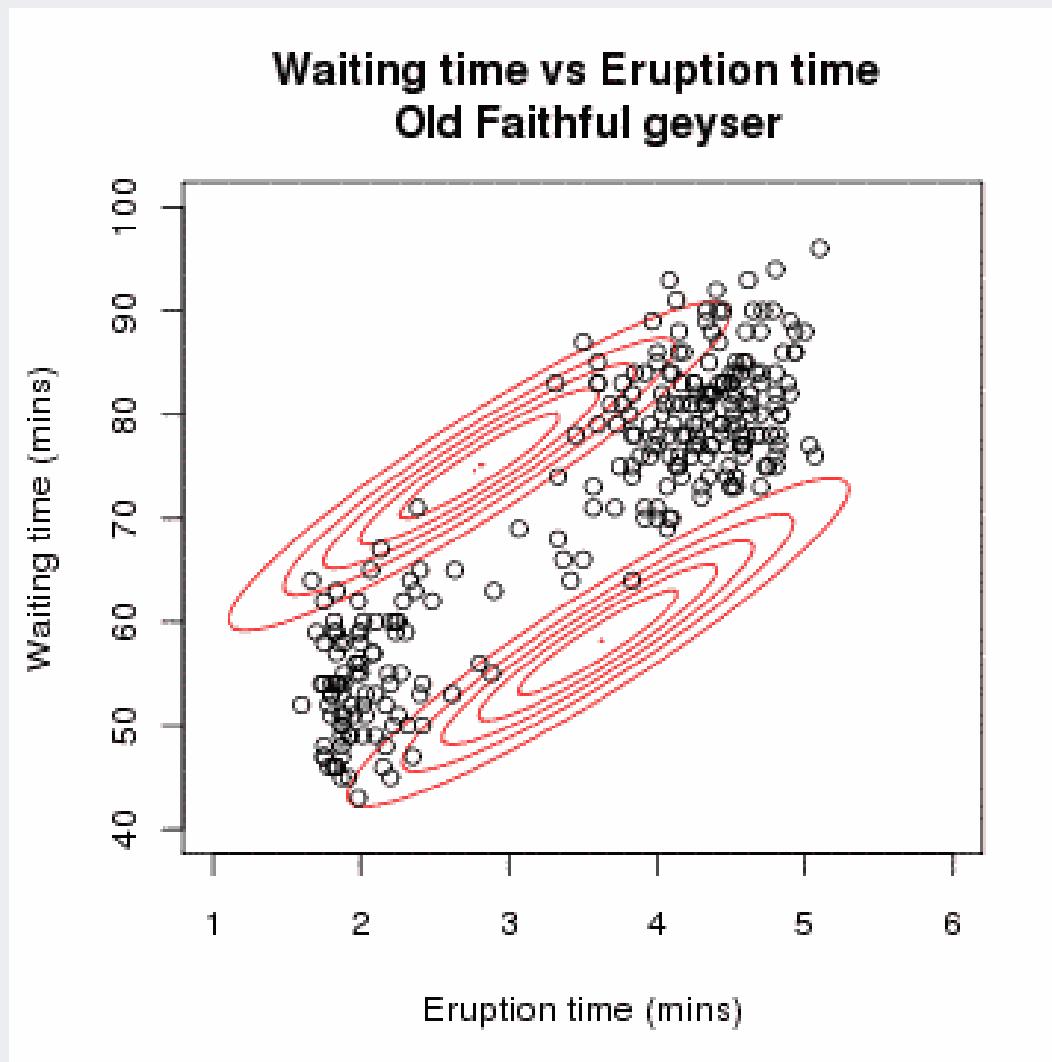
- Expectation-Maximization Algorithm

- ✓ E-Step: Given the current estimate of the parameters, compute the conditional probabilities
- ✓ M-Step: Update the parameters to maximize the expected likelihood found in the E-Step



# Mixture of Gaussian Density Estimation

- Expectation-Maximization Algorithm: Illustrative example



# Mixture of Gaussian Density Estimation

- EM algorithm for MoG

✓ Expectation

$$p(m|\mathbf{x}_i, \lambda) = \frac{w_m g(\mathbf{x}_i | \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{k=1}^M w_k g(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

✓ Maximization

- Mixture weight

$$w_m^{(new)} = \frac{1}{N} \sum_{i=1}^N p(m|\mathbf{x}_i, \lambda)$$

- Means and variances

$$\boldsymbol{\mu}_m^{(new)} = \frac{\sum_{i=1}^N p(m|\mathbf{x}_i, \lambda) \mathbf{x}_i}{\sum_{i=1}^N p(m|\mathbf{x}_i, \lambda)}, \quad \sigma_m^{2(new)} = \frac{\sum_{i=1}^N p(m|\mathbf{x}_i, \lambda) \mathbf{x}_i^2}{\sum_{i=1}^N p(m|\mathbf{x}_i, \lambda)} - \boldsymbol{\mu}_m^{2(new)}$$

# Mixture of Gaussian Density Estimation

- The shape of MoG according to the covariance matrix

Spherical

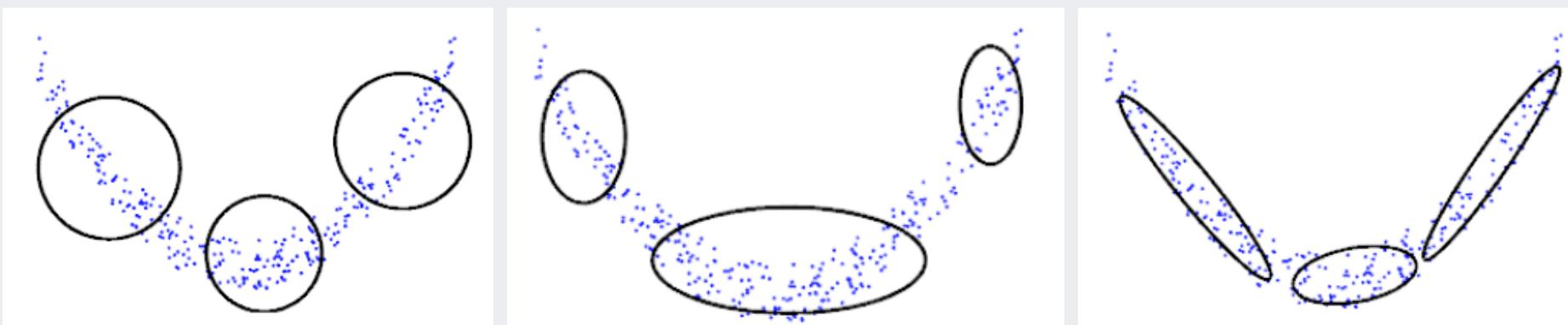
$$\Sigma = \sigma^2 \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$$

Diagonal

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_d^2 \end{bmatrix}$$

Full

$$\Sigma = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1d} \\ \vdots & \ddots & \vdots \\ \sigma_{d1} & \cdots & \sigma_{dd} \end{bmatrix}$$



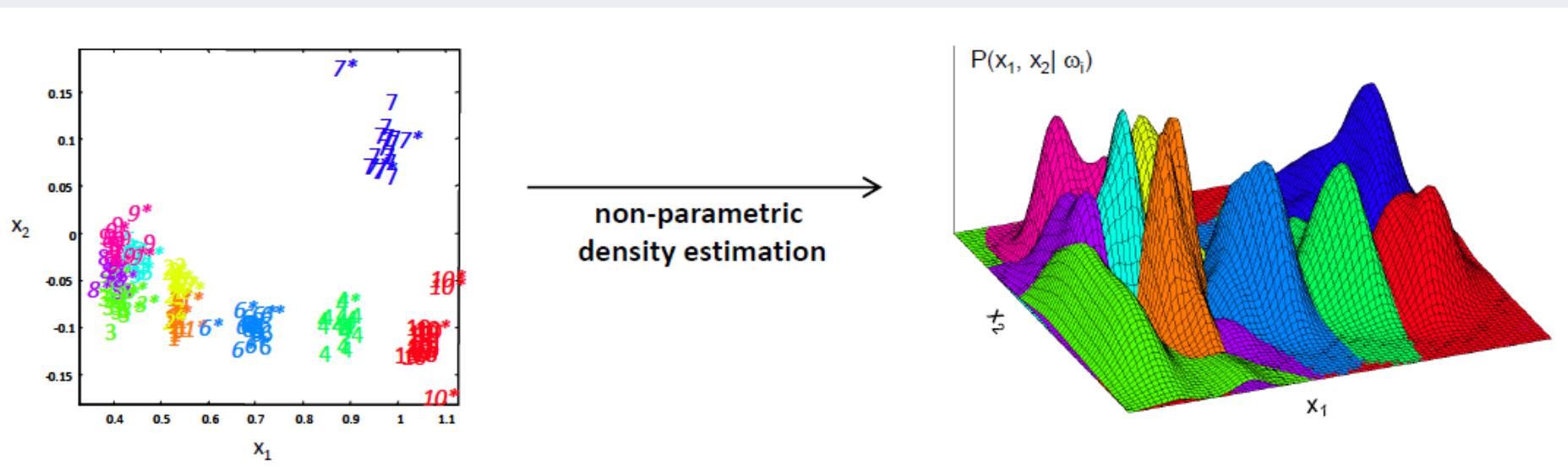
- Less precise
- Very efficient to compute

- More precise
- Efficient to compute

- Very precise
- Less efficient to compute

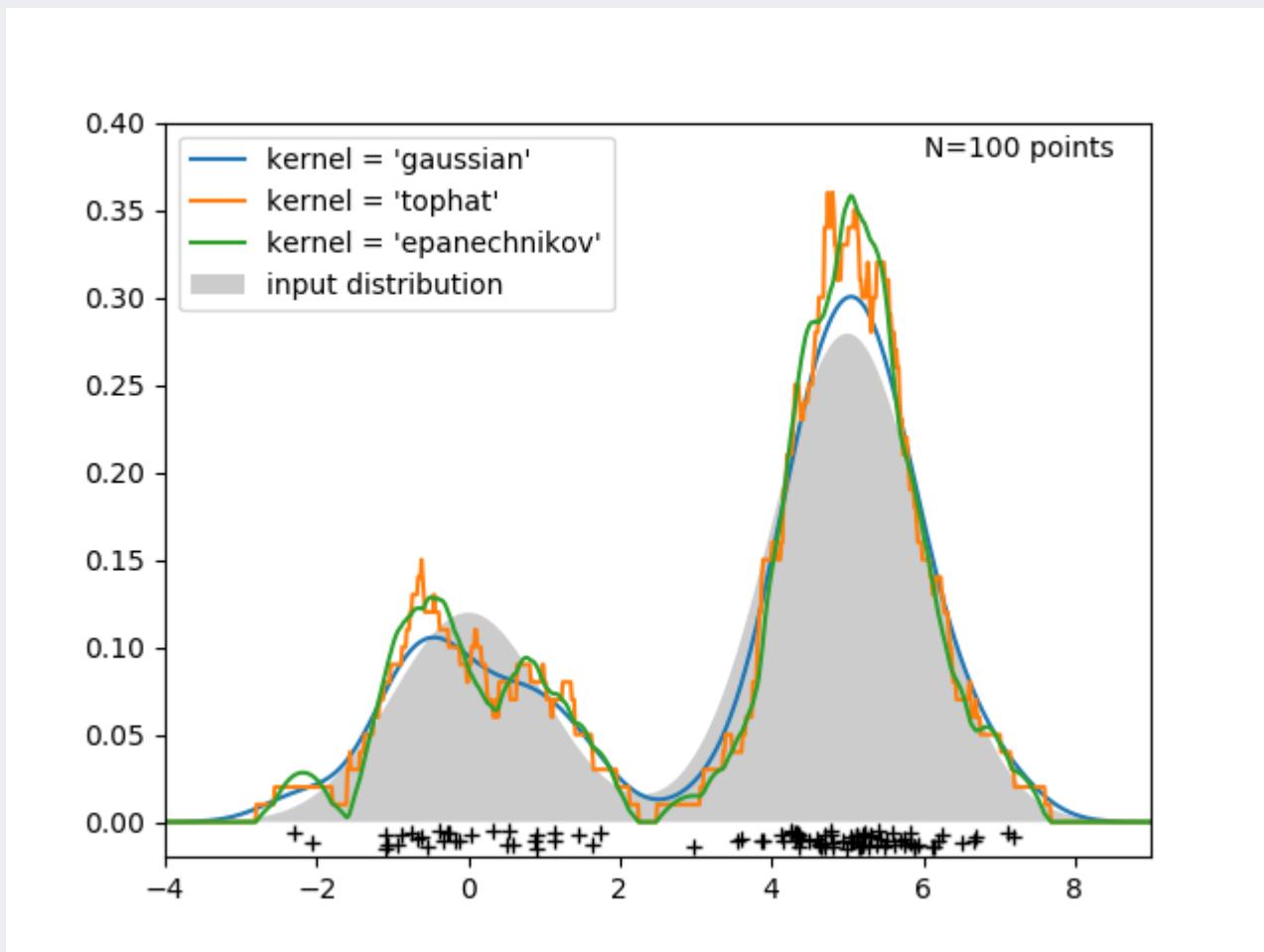
# Kernel-density Estimation

- Kernel-density Estimation
  - ✓ Attempts to estimate the density directly from the data **without assuming a particular form** for the underlying distribution



# Kernel-density Estimation

- Kernel-density Estimation: 1-D example



# Kernel-density Estimation

- Kernel-density Estimation
  - ✓ The probability that a vector  $x$ , drawn from a distribution  $p(x)$ , will fall in a given region  $R$  of the sample space

$$P = \int_R p(x')dx'$$

- ✓ Suppose that  $N$  vectors  $\{x^1, x^2, \dots, x^n\}$  are drawn from the distribution; the probability that  $k$  of these  $N$  vectors fall in  $R$  is given by

$$P(k) = \binom{N}{k} P^k (1 - P)^{N-k}$$

- ✓ It can be shown that (from the binomial distribution) the mean and variance of the ratio  $k/N$  are

$$E\left[\frac{k}{N}\right] = P, \quad Var\left[\frac{k}{N}\right] = \frac{P(1 - P)}{N}$$

# Kernel-density Estimation

- Kernel-density Estimation

- ✓ As  $N \rightarrow \infty$ , the distribution becomes sharper (the variance gets smaller), so we can expect that a good estimate of the probability  $P$  can be obtained from the mean fraction of the points that fall within  $R$

$$P \cong \frac{k}{N}$$

- ✓ If we assume that  $R$  is so small that  $p(x)$  does not vary appreciably within it, then

$$P = \int_R p(x')dx' \cong p(x)V$$

- ✓ where  $V$  is the volume enclosed by region  $R$

- ✓ Merging the two previous results

$$P = \int_R p(x')dx' \cong p(x)V = \frac{k}{N}, \quad p(x) = \frac{k}{NV}$$

# Kernel-density Estimation

- Kernel-density Estimation

$$p(x) = \frac{k}{NV}, \quad \text{where} \begin{cases} V: \text{volume surrounding } x \\ N: \text{the total number of examples} \\ k: \text{the number of examples inside } V \end{cases}$$

- ✓ Estimation becomes more accurate as we increase the number of sample points  $N$  and shrink the volume  $V$
- ✓ In practice, the total number of examples is fixed so that we have to find a compromise for  $V$ 
  - Large enough to include enough examples within  $R$
  - Small enough to support the assumption that  $p(x)$  is constant within  $R$
- ✓ Fix  $V$  and determine  $k$  from the data: Kernel-density estimation
- ✓ Fix  $k$  and determine  $V$  from the data:  $k$ -nearest neighbor density estimation

# Parzen Window Density Estimation

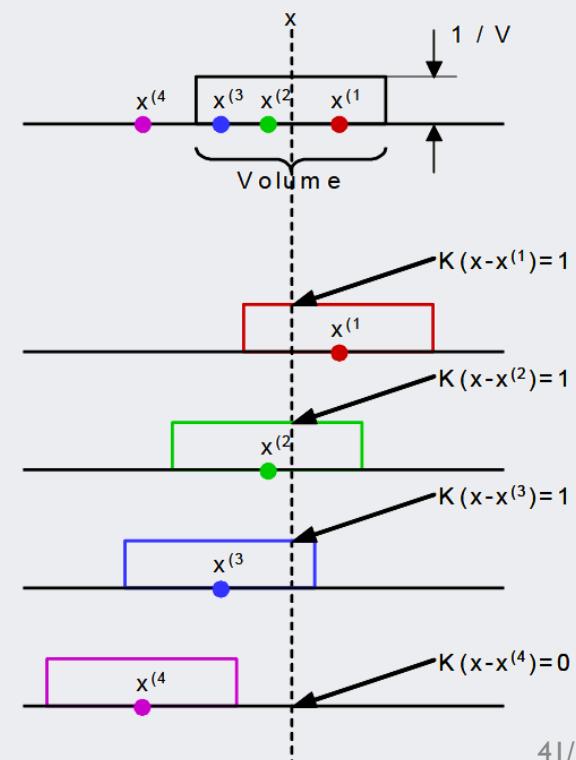
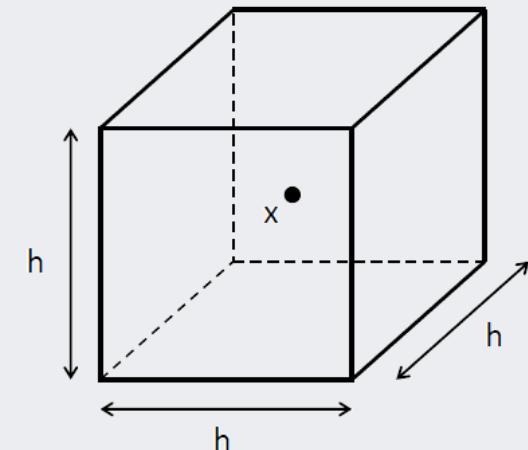
- Parzen Window Density Estimation

- ✓ Assume that the region  $R$  that encloses the  $k$  examples is a hypercube with sides of length  $h$  centered at  $x$ 
  - Its volume is given by  $V = h^d$ ,  $d$ : N. dimensions

- ✓ Define a kernel function  $K(u)$

$$K(u) = \begin{cases} 1 & |u_j| < \frac{1}{2} \forall j = 1, \dots, d \\ 0 & \text{otherwise} \end{cases}$$

$$k = \sum_{i=1}^N K\left(\frac{\mathbf{x}^i - \mathbf{x}}{h}\right) \quad p(x) = \frac{1}{Nh^d} \sum_{i=1}^N K\left(\frac{\mathbf{x}^i - \mathbf{x}}{h}\right)$$



# Parzen Window Density Estimation

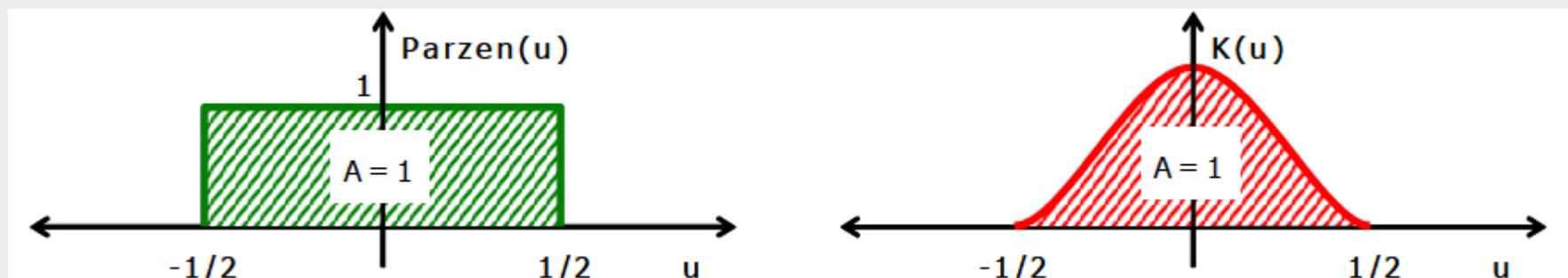
- Drawbacks of  $K(u)$

- ✓ Yields density estimate that have discontinuities
- ✓ Weights equally all points  $\mathbf{x}^i$ , regardless of their distance to the estimation point  $\mathbf{x}$

- Smooth kernel function

$$P = \int_R K(x) dx = 1$$

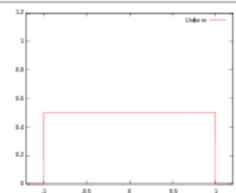
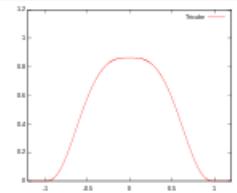
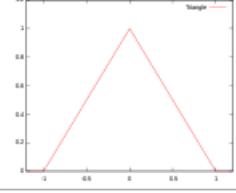
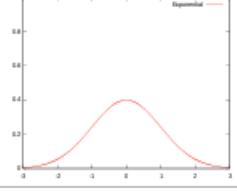
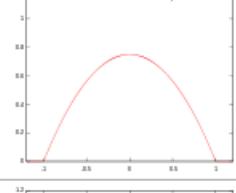
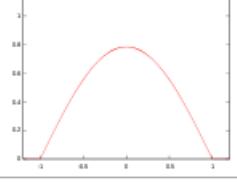
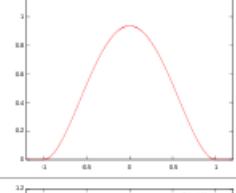
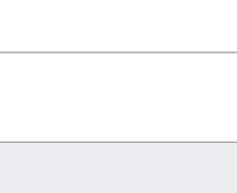
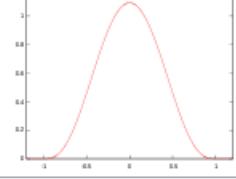
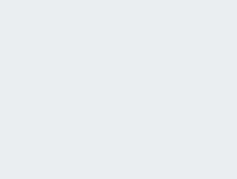
- ✓ Commonly use a radially symmetric and unimodal pdf, such as Gaussian



$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N K\left(\frac{\mathbf{x}^i - \mathbf{x}}{h}\right)$$

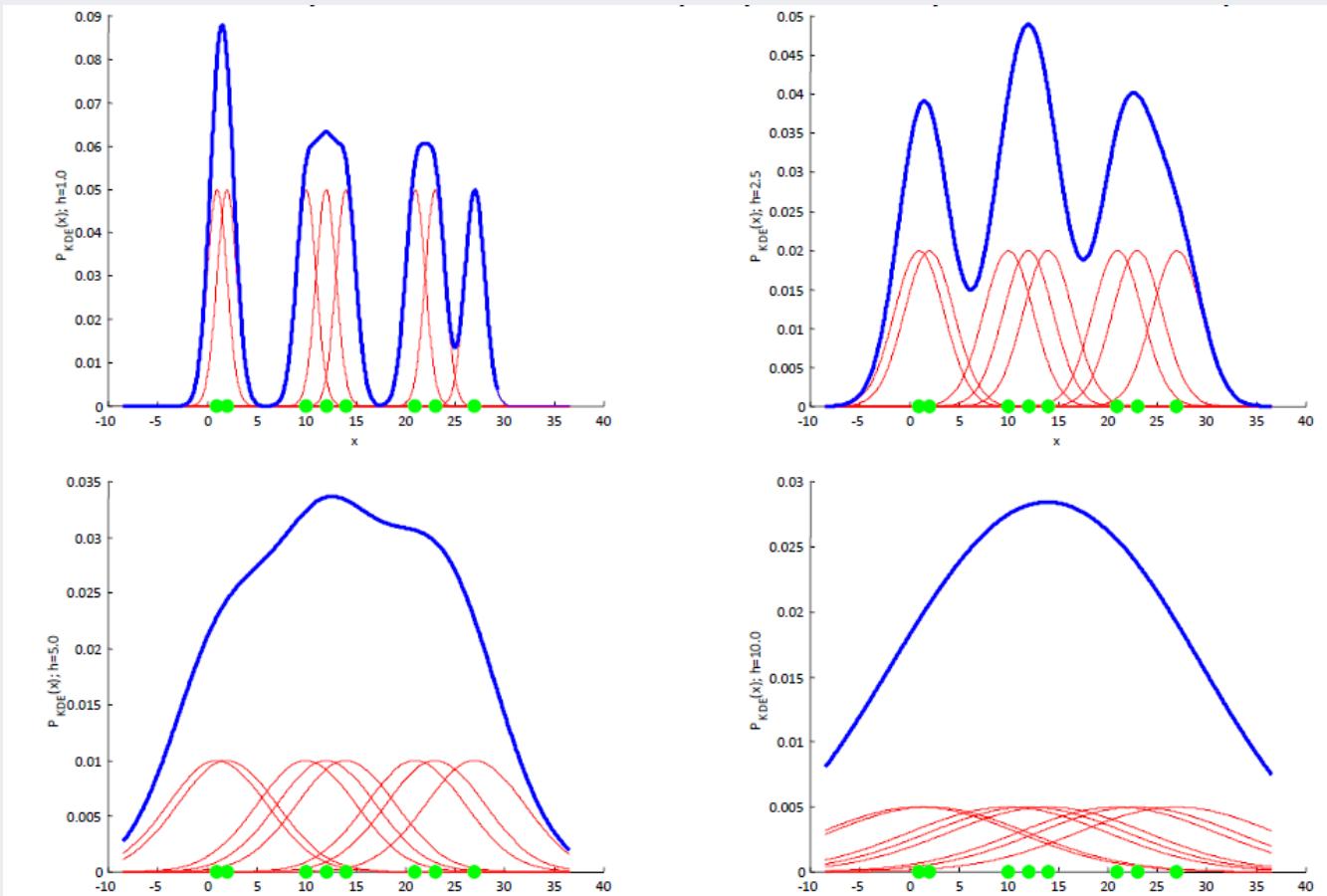
# Parzen Window Density Estimation

- Example of smooth kernels

<b>Uniform</b>	$K(u) = \frac{1}{2} \mathbf{1}_{\{ u  \leq 1\}}$	 A step function plot labeled "Uniform". The x-axis ranges from -1 to 1, and the y-axis ranges from 0 to 1.2. The function is constant at 0.5 for  u  ≤ 1 and drops to 0 outside this range.	<b>Tricube</b>	$K(u) = \frac{70}{81}(1 -  u ^3)^3 \mathbf{1}_{\{ u  \leq 1\}}$	 A smooth bell-shaped curve labeled "Tricube". The x-axis ranges from -3 to 3, and the y-axis ranges from 0 to 1.2. The curve is zero at  u  > 1 and reaches a maximum value of approximately 0.9 at u=0.
<b>Triangular</b>	$K(u) = (1 -  u ) \mathbf{1}_{\{ u  \leq 1\}}$	 A triangular plot labeled "Triangular". The x-axis ranges from -1 to 1, and the y-axis ranges from 0 to 1.2. The function is zero at  u  > 1 and reaches a maximum value of 1 at u=0.	<b>Gaussian</b>	$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$	 A smooth bell-shaped curve labeled "Gaussian". The x-axis ranges from -3 to 3, and the y-axis ranges from 0 to 1.2. The curve is centered at 0 and has a standard deviation of approximately 1.57.
<b>Epanechnikov</b>	$K(u) = \frac{3}{4}(1 - u^2) \mathbf{1}_{\{ u  \leq 1\}}$	 A smooth bell-shaped curve labeled "Epanechnikov". The x-axis ranges from -1 to 1, and the y-axis ranges from 0 to 1.2. The curve is zero at  u  > 1 and reaches a maximum value of 0.75 at u=0.	<b>Cosine</b>	$K(u) = \frac{\pi}{4} \cos\left(\frac{\pi}{2}u\right) \mathbf{1}_{\{ u  \leq 1\}}$	 A smooth bell-shaped curve labeled "Cosine". The x-axis ranges from -1 to 1, and the y-axis ranges from 0 to 1.2. The curve is zero at  u  > 1 and reaches a maximum value of 0.707 at u=0.
<b>Quartic (biweight)</b>	$K(u) = \frac{15}{16}(1 - u^2)^2 \mathbf{1}_{\{ u  \leq 1\}}$	 A smooth bell-shaped curve labeled "Quartic". The x-axis ranges from -1 to 1, and the y-axis ranges from 0 to 1.2. The curve is zero at  u  > 1 and reaches a maximum value of 0.9375 at u=0.	<b>Logistic</b>	$K(u) = \frac{1}{e^u + 2 + e^{-u}}$	 A smooth bell-shaped curve labeled "Logistic". The x-axis ranges from -3 to 3, and the y-axis ranges from 0 to 1.2. The curve is zero at  u  > 1 and reaches a maximum value of 0.27 at u=0.
<b>Triweight</b>	$K(u) = \frac{35}{32}(1 - u^2)^3 \mathbf{1}_{\{ u  \leq 1\}}$	 A smooth bell-shaped curve labeled "Triweight". The x-axis ranges from -1 to 1, and the y-axis ranges from 0 to 1.2. The curve is zero at  u  > 1 and reaches a maximum value of 1.05 at u=0.	<b>Silverman kernel<sup>[4]</sup></b>	$K(u) = \frac{1}{2} e^{-\frac{ u }{\sqrt{2}}} \cdot \sin\left(\frac{ u }{\sqrt{2}} + \frac{\pi}{4}\right)$	 A smooth bell-shaped curve labeled "Silverman kernel". The x-axis ranges from -3 to 3, and the y-axis ranges from 0 to 1.2. The curve is zero at  u  > 1 and reaches a maximum value of approximately 0.82 at u=0.

# Parzen Window Density Estimation

- Smoothing parameter (bandwidth)  $h$ 
  - ✓ A large  $h$  will over-smooth the density distribution
  - ✓ A small  $h$  will result in a spiky density distribution

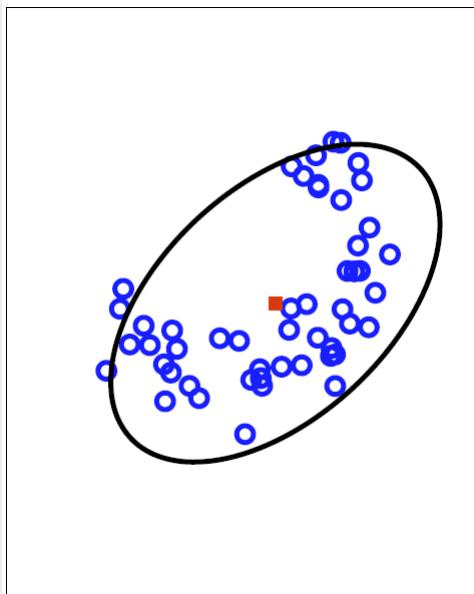


# Parzen Window Density Estimation

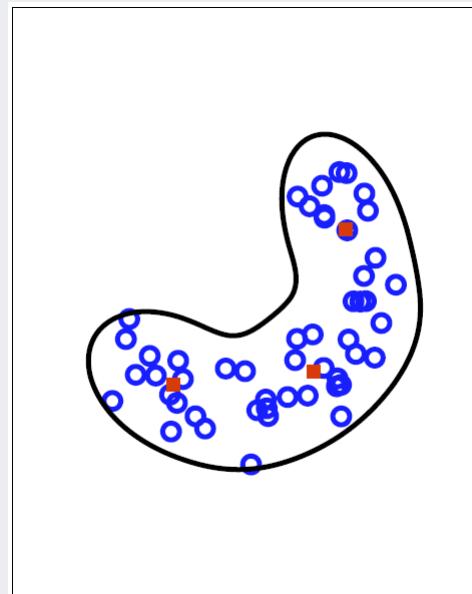
- Kernel Density Estimation

- ✓ The smoothing parameter  $h$  can be optimized through EM algorithm

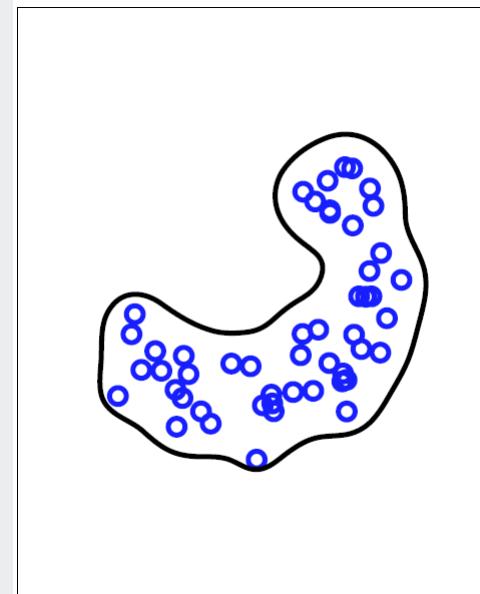
Gaussian density estimation



Mixture of Gaussian

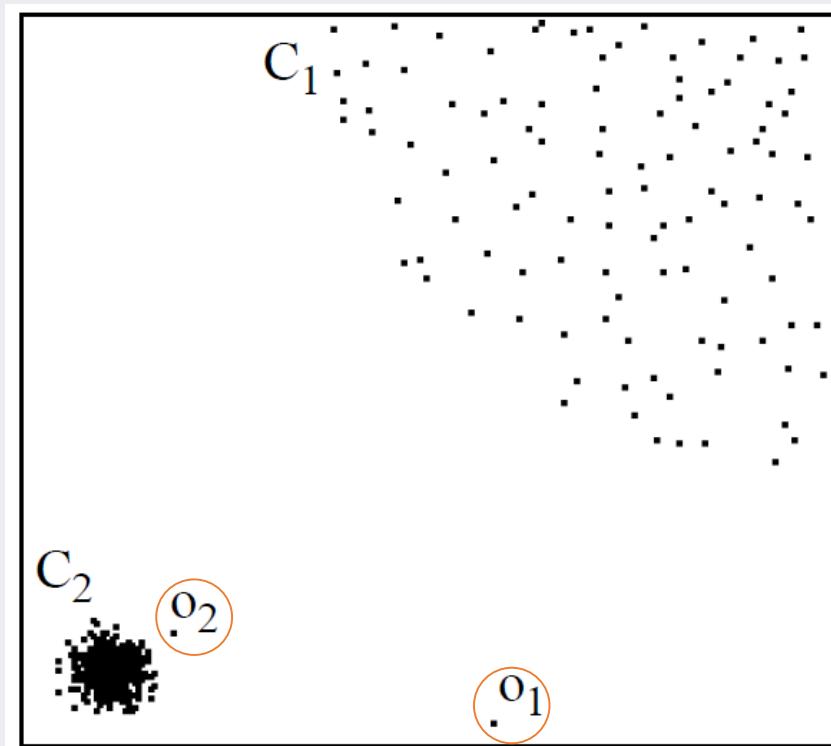


Parzen window with Gaussian Kernel



# Local Outlier Factors (LOF)

- Motivation
  - ✓ Compute the novelty score of an instance by considering local density around it



# Local Outlier Factors (LOF)

- Local Outlier Factors (LOF)

✓ **Definition I:** k-distance of an object p

- For any positive integer k, the k-distance of object p, denoted as k-distance(p), is defined as the distance  $d(p,o)$  between p and an object o in D such that
- for at least k objects o' in  $D \setminus \{p\}$  it holds that  $d(p, o') \leq d(p, o)$
- for at most k-1 objects o' in  $D \setminus \{p\}$  it holds that  $d(p, o') < d(p, o)$
- Simply it is the distance to the k-th nearest neighbor considering ties.

1st	2nd	3rd	4th	5th	6th	7th	3-distance
1	2	3	3	3	4	5	3
1	2	2	2	3	4	5	2
1	1	1	1	2	3	4	1

# Local Outlier Factors (LOF)

- Local Outlier Factors (LOF)

✓ Definition 2: k-distance neighborhood of an object p

- Given the k-distance of p, the k-distance neighborhood of p contains every object whose distance from p is not greater than the k-distance

$$N_k(p) = \{q \in D \setminus \{p\} \mid d(p, q) \leq k - \text{distance}(p)\}$$

- Example

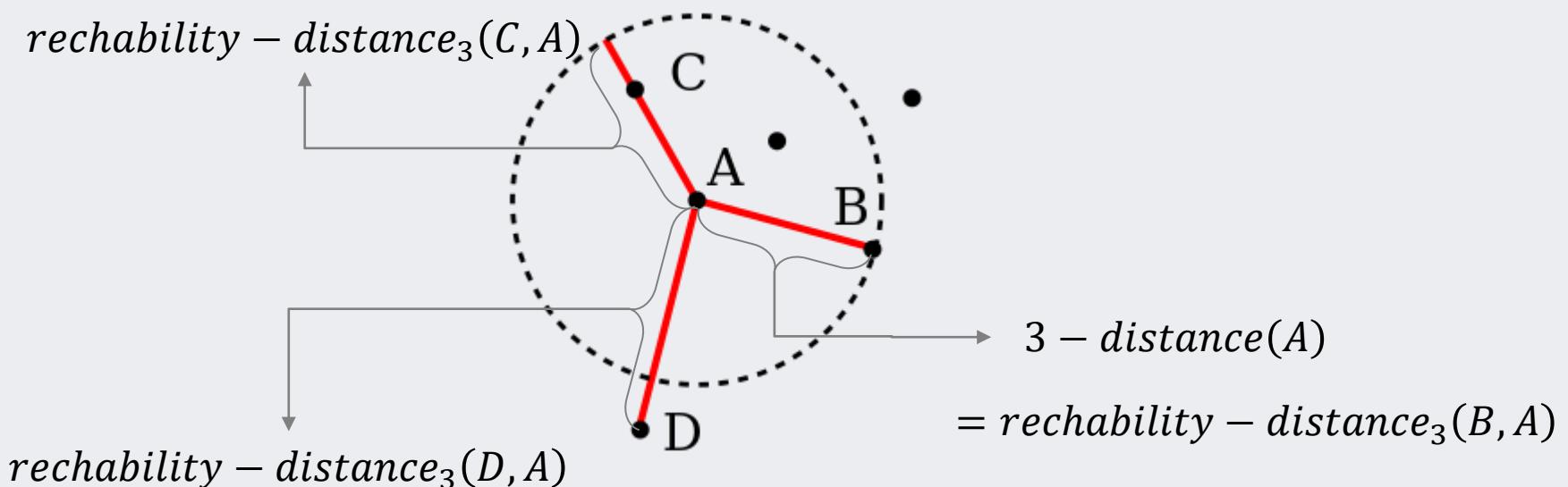
1st	2nd	3rd	4th	5th	6th	7th	$k - \text{distance}(3)$	$N_3(p)$
1	2	3	3	3	4	5	3	5
1	2	2	2	3	4	5	2	4
1	1	1	1	2	3	4	1	4

# Local Outlier Factors (LOF)

- Local Outlier Factors (LOF)

- ✓ **Definition 3:** reachability distance

- $\text{reachability-distance}_k(p, o) = \max\{k - \text{distance}(o), d(p, o)\}$
  - Examples

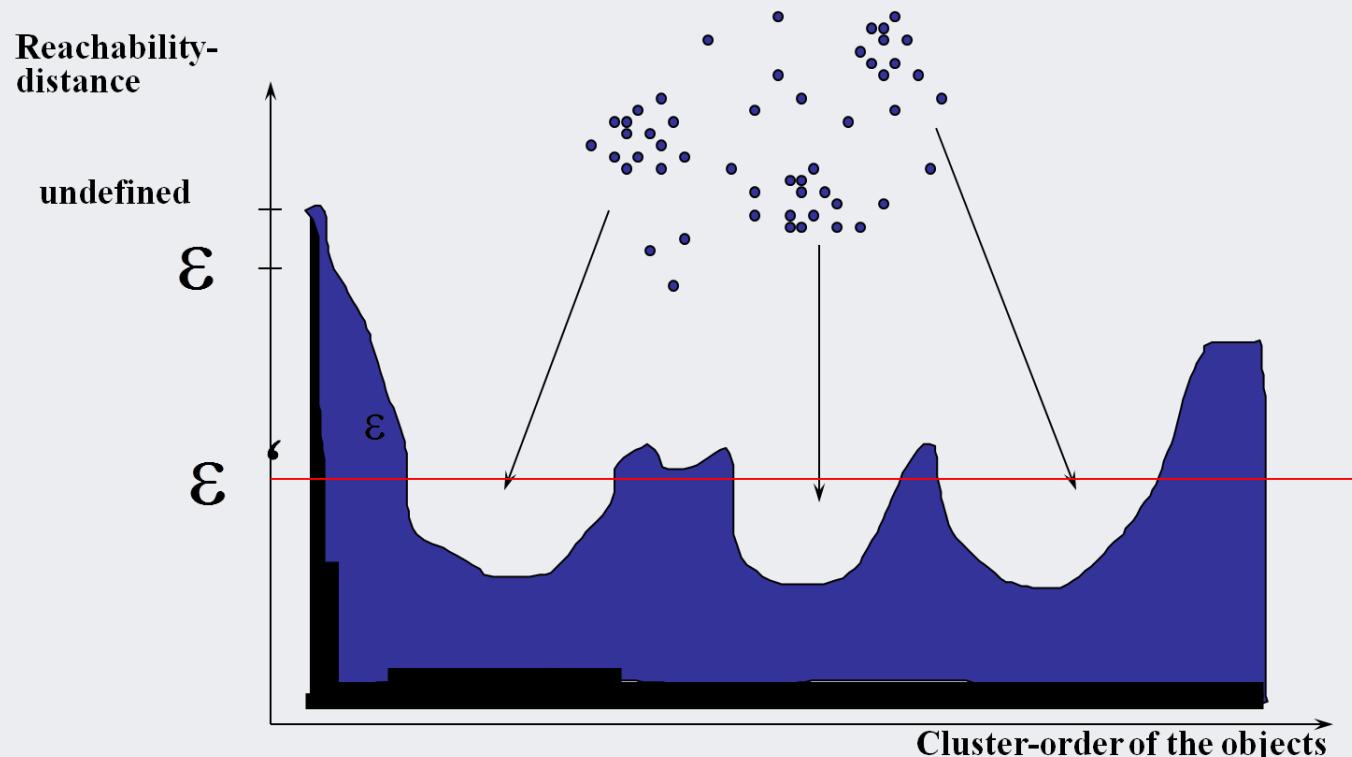


# Local Outlier Factors (LOF)

- Local Outlier Factors (LOF)

- ✓ **Definition 3:** reachability distance

- $\text{reachability-distance}_k(p, o) = \max\{k - \text{distance}(o), d(p, o)\}$
  - Distribution of reachability-distance



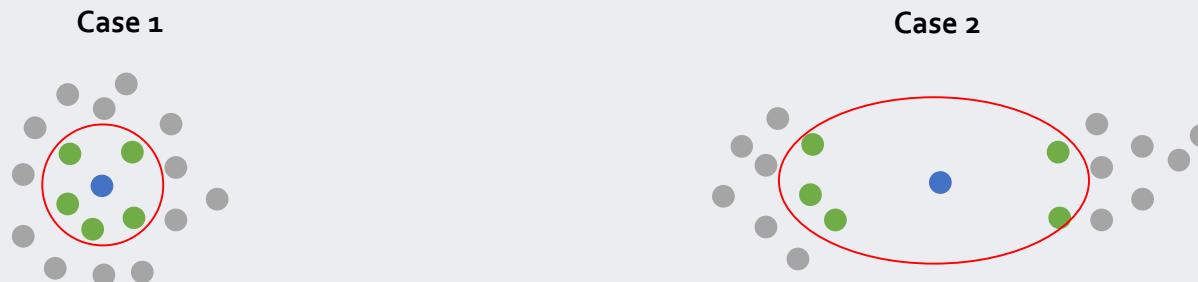
# Local Outlier Factors (LOF)

- Local Outlier Factors (LOF)

✓ Definition 4: local reachability density of an object p

$$lrd_k(p) = \frac{|N_k(p)|}{\sum_{o \in N_k(p)} \text{reachability} - \text{distance}_k(p, o)}$$

- Case 1: p is located in the middle of a denser area: the denominator of  $lrd_k(p)$  becomes small, which results in a large  $lrd_k(p)$
- Case 2: p is located in a spare area between two dense data clusters: the denominator of  $lrd_k(p)$  becomes large, which results in a small  $lrd_k(p)$

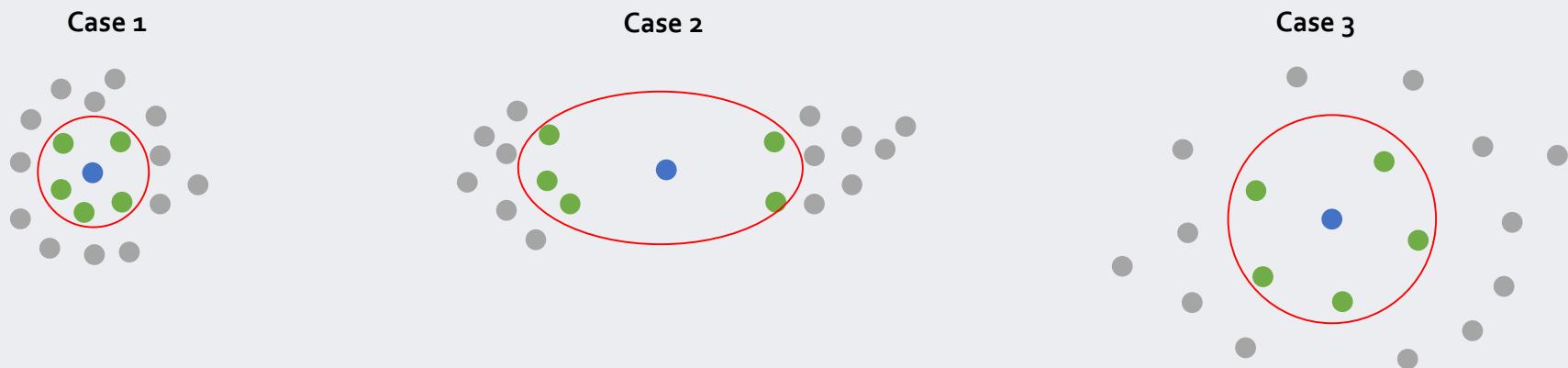


# Local Outlier Factors (LOF)

- Local Outlier Factors (LOF)

✓ Definition 5: local outlier factor of an object  $p$

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|} = \frac{1}{lrd_k(p)} \frac{\sum_{o \in N_k(p)} lrd_k(o)}{|N_k(p)|}$$

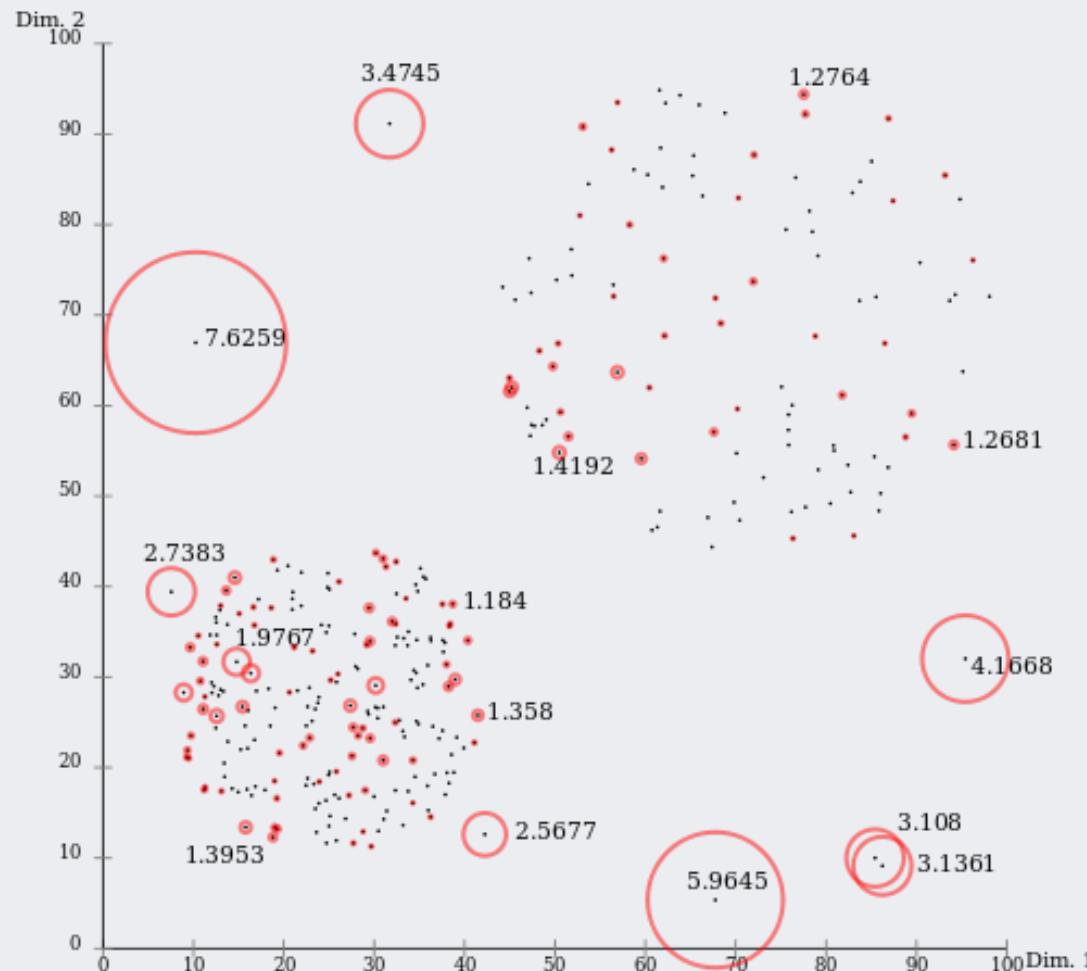


Case	$lrd_k(p)$	$lrd_k(o)$	$LOF_k(p)$
Case 1	Large	Large	Small
Case 2	Small	Large	Large
Case 3	Small	Small	Small

# Local Outlier Factors (LOF)

- Local Outlier Factors (LOF)

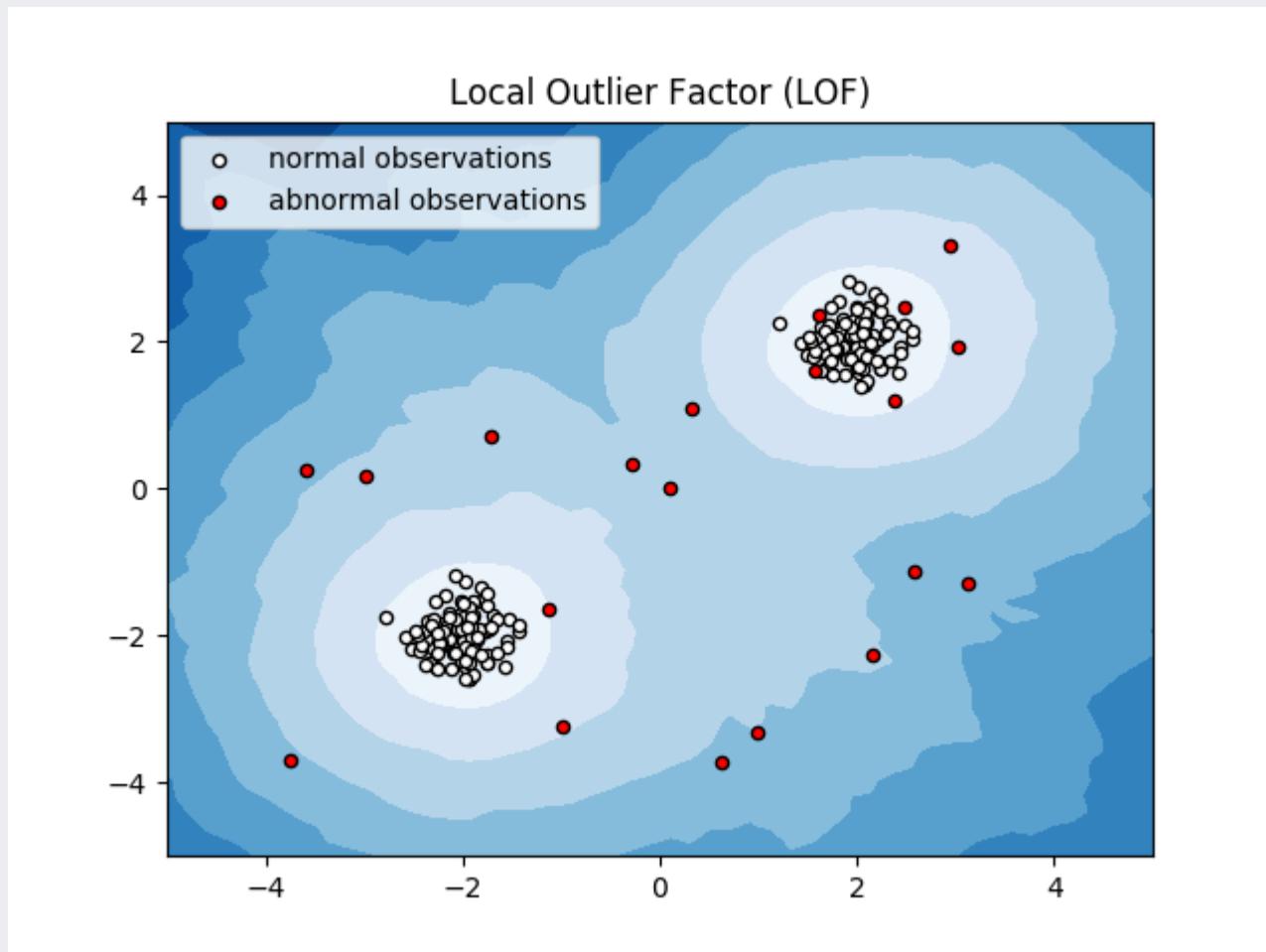
- ✓ For each point, compute the density of its local neighborhood



# Local Outlier Factors (LOF)

- Local Outlier Factors (LOF)

- ✓ LOF contour plot



# AGENDA

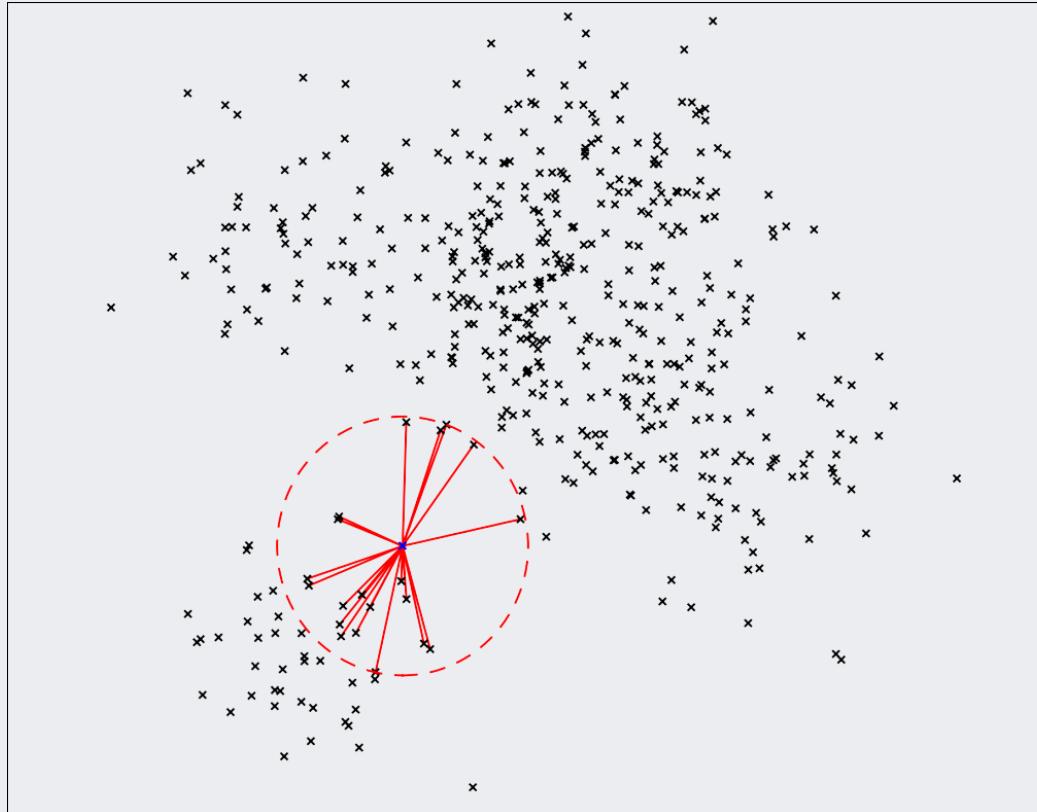
- 01 Novelty Detection: Overview
- 02 Density-based Novelty Detection
- 03 Distance/Reconstruction-based ND
- 04 Model-based Novelty Detection

# k-Nearest Neighbor-based Novelty Detection

Harmeling et al. (2006)

- k-Nearest Neighbor-based Approach

- ✓ Novelty score of an instance is computed based on the distance information to k nearest neighbors
- ✓ Does not assume any prior probability distribution for the normal class



# k-Nearest Neighbor-based Novelty Detection

- Various distance information used for novelty score

- ✓ Maximum distance to the k-th nearest neighbor

$$d_{max}^k = \kappa(\mathbf{x}) = \|\mathbf{x} - z_k(\mathbf{x})\|$$

- ✓ Average distance to the k-nearest neighbors

$$d_{avg}^k = \gamma(\mathbf{x}) = \frac{1}{k} \sum_{j=1}^k \|\mathbf{x} - z_j(\mathbf{x})\|$$

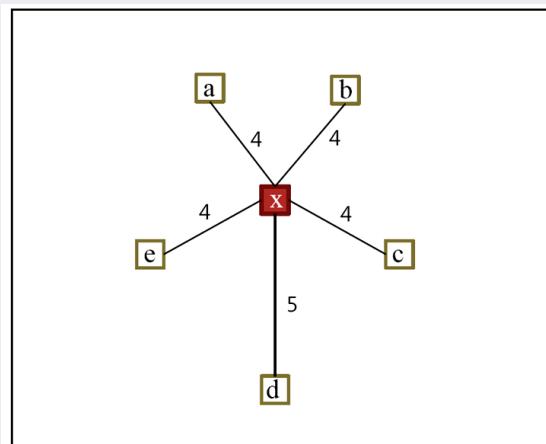
- ✓ Distance to the mean of the k-nearest neighbors

$$d_{mean}^k = \delta(\mathbf{x}) = \left\| \mathbf{x} - \frac{1}{k} \sum_{j=1}^k z_j(\mathbf{x}) \right\|$$

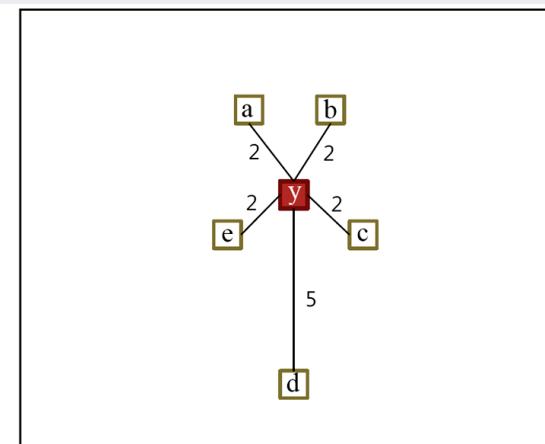
# k-Nearest Neighbor-based Novelty Detection

Kang and Cho (2009)

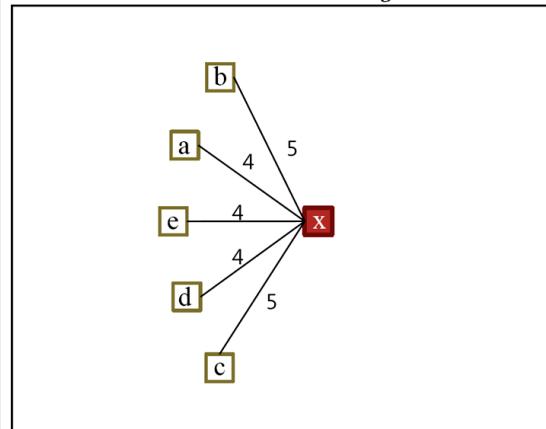
- Various distance information used for novelty score
  - ✓ Comparison among the maximum, average, and mean distance



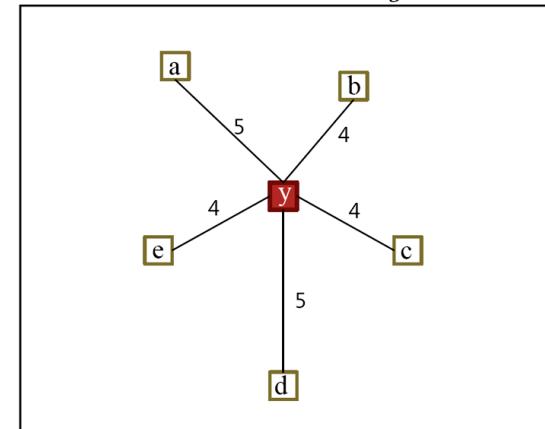
$$(a) d_{max}^5 = 5.0, d_{avg}^5 = 4.2.$$



$$(b) d_{max}^5 = 5.0, d_{avg}^5 = 2.6.$$



$$(c) d_{avg}^5 = 4.4, d_{mean}^5 = 3.3.$$



$$(d) d_{avg}^5 = 4.4, d_{mean}^5 = 2.1.$$

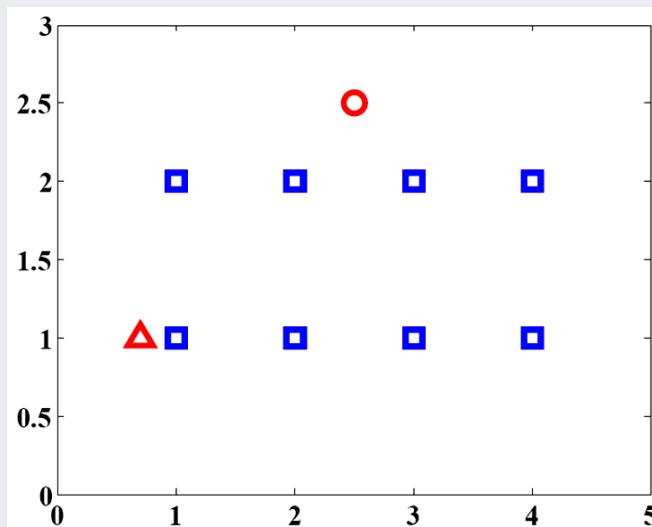
# k-Nearest Neighbor-based Novelty Detection

Kang and Cho (2009)

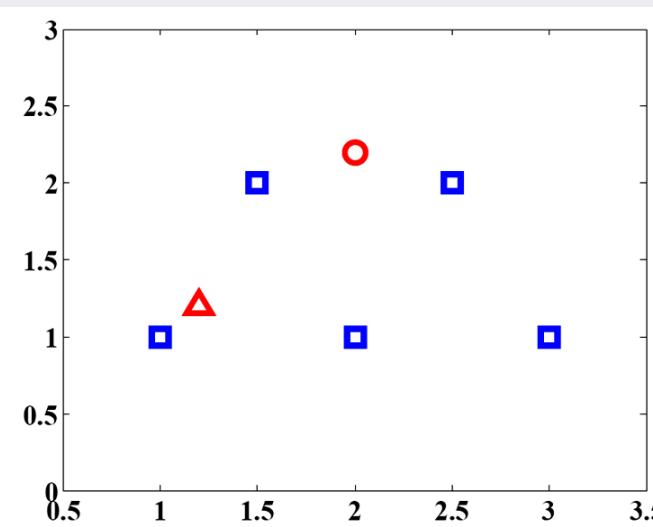
- Counter example of the previous novelty scores

✓ Which one should be identified as novel?

A



B



		$d^k_{\max}$	$d^k_{\text{avg}}$	$d^k_{\text{mean}}$
A (k=4)	Circle	1.58	1.14	0.50
	Triangle	1.64	1.07	0.94
B (k=5)	Circle	1.56	1.08	0.80
	Triangle	1.86	1.09	0.88

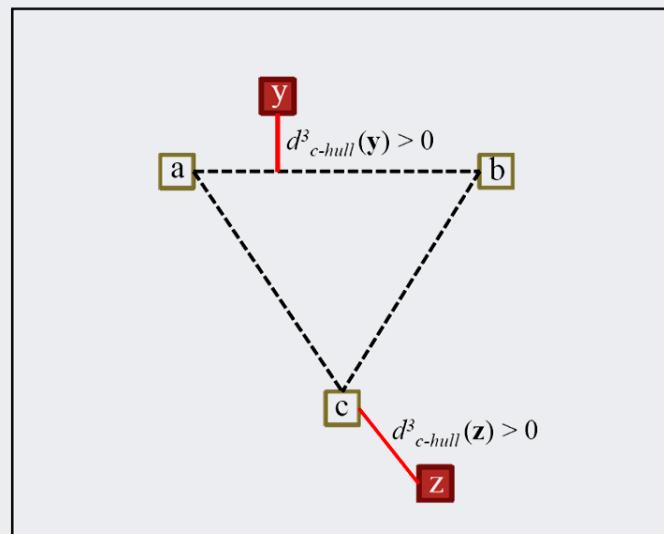
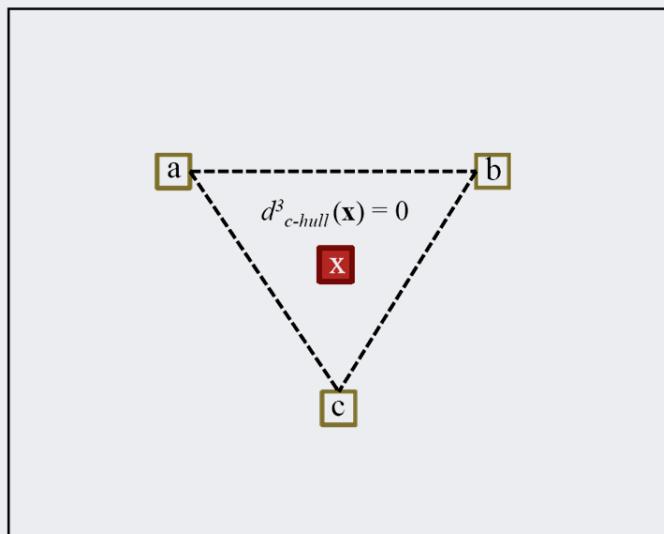
# k-Nearest Neighbor-based Novelty Detection

- Consider additional factor

✓ whether the new instance is located inside the convex hull of its neighbors

$$\min_{\mathbf{w}} \left( d_{c-hull}^k(\mathbf{x}) \right)^2 = \left\| \mathbf{x}_{new} - \sum_{j=1}^k \mathbf{w}_i z_j(\mathbf{x}) \right\|^2$$

$$s.t. \quad \sum_{i=1}^k \mathbf{w}_i = 1, \quad \mathbf{w}_i \geq 0, \quad \forall i.$$



# k-Nearest Neighbor-based Novelty Detection

- Combine the average distance and convex distance

- ✓ Average distance to the k-nearest neighbors

$$d_{avg}^k = \frac{1}{k} \sum_{j=1}^k \|\mathbf{x} - z_j(\mathbf{x})\|$$

- ✓ Convex distance to its k-nearest neighbors

$$d_{c-hull}^k = \left\| \mathbf{x} - \sum_{j=1}^k \mathbf{w}_i z_j(\mathbf{x}) \right\|$$

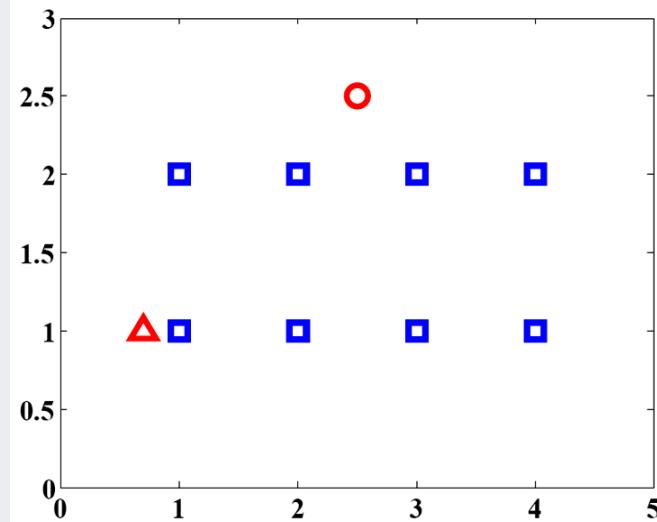
- ✓ Put the penalty term using the convex distance for those instances located outside the convex hull of its k-nearest neighbors

$$d_{hybrid}^k = d_{avg}^k \times \left( \frac{2}{1 + \exp(-d_{c-hull}^k)} \right)$$

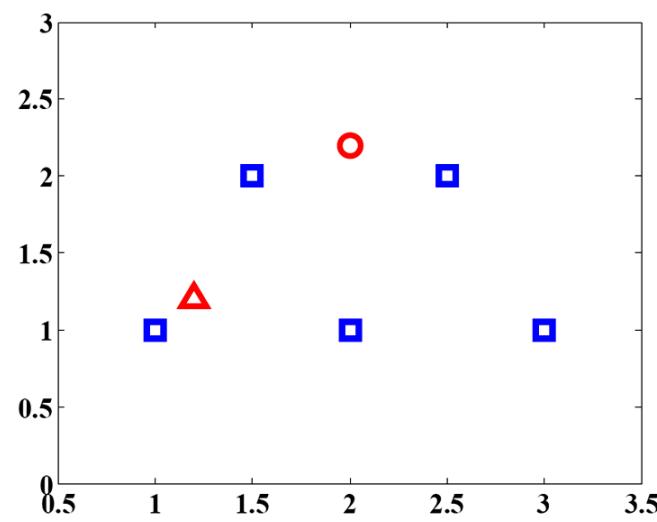
# k-Nearest Neighbor-based Novelty Detection

- Counter example revisited

A

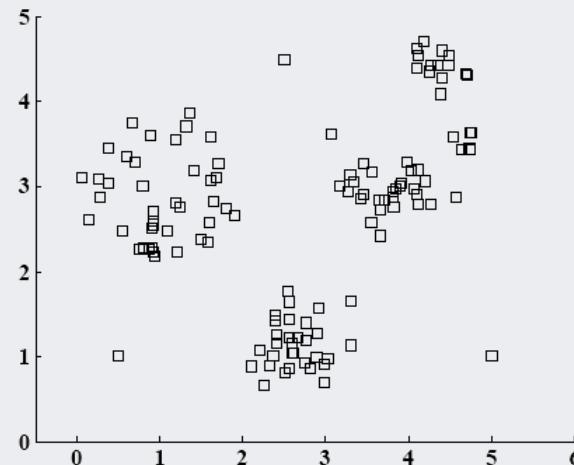


B

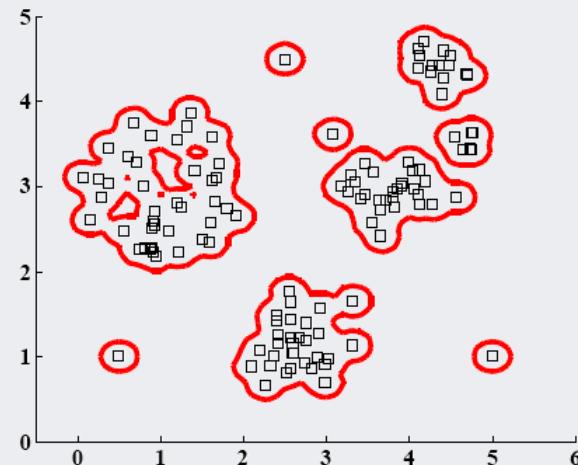


		$d^k_{\max}$	$d^k_{\text{avg}}$	$d^k_{\text{mean}}$	$d^k_{\text{hybrid}}$
A (k=4)	Circle	1.58	1.14	0.50	1.42
	Triangle	1.64	1.07	0.94	1.18
B (k=5)	Circle	1.56	1.08	0.80	1.18
	Triangle	1.86	1.09	0.88	1.09

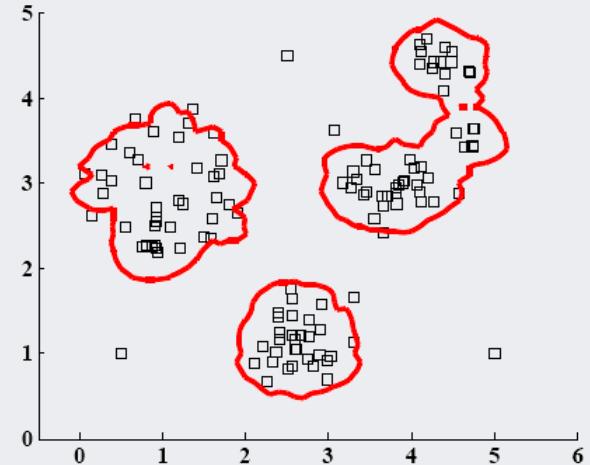
# k-Nearest Neighbor-based Novelty Detection



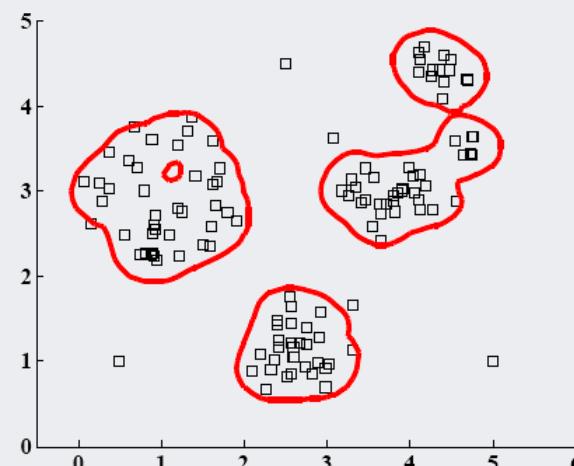
(a) Normal instances



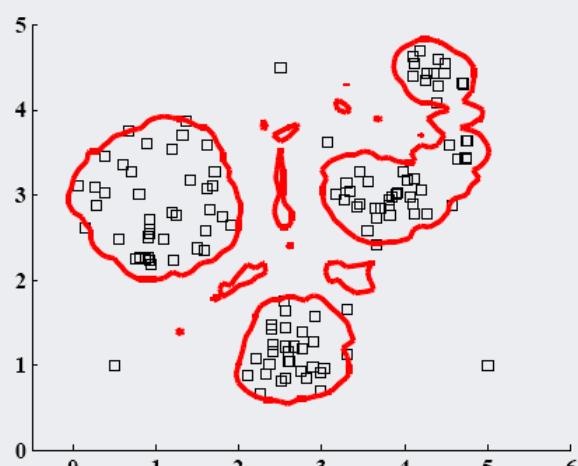
(b) 1-NN



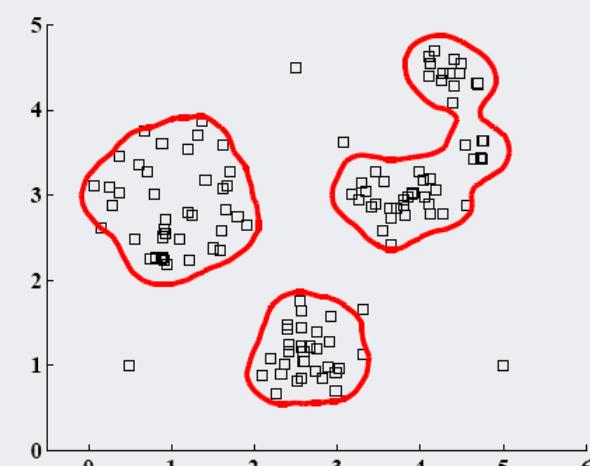
(c)  $d_{max}^5$



(d)  $d_{avg}^5$



(e)  $d_{mean}^5$



(f)  $d_{hybrid}^5$

# k-Nearest Neighbor-based Novelty Detection

- Experiment
- ✓ Datasets

No.	Name	Source	Class	Dim.	TrN <sub>n</sub>	TsN <sub>n</sub>	TsN <sub>o</sub>
1	Banana	Rätsch	-1	2	216	2,708	271
2	Titanic	Rätsch	-1	3	100	1,390	139
3	Liver	UCI	healthy	6	73	72	7
4	Ecoli	UCI	cp	7	72	71	7
5	Yeast	UCI	0	8	232	231	23
6	Pima	UCI	0	8	250	250	25
7	Diabetes	Rätsch	-1	8	304	196	20
8	Glass	UCI	1	9	35	35	4
9	Breast	Rätsch	-1	9	142	54	5
10	Flare	Rätsch	-1	9	300	178	18
11	Heart	Rätsch	-1	13	94	56	6
12	Image	Rätsch	-1	18	554	436	44
13	Twonorm	Rätsch	-1	20	198	3,499	350
14	German	Rätsch	-1	20	489	211	21
15	Waveform	Rätsch	-1	21	268	3,085	308
16	Parkinsons	UCI	parkinsons	22	74	73	7
17	Ionosphere	UCI	0	33	113	112	11
18	Spectf	UCI	0	44	28	27	3
19	Sonar	UCI	mine	60	56	55	6
20	Ozone	UCI	non-ozone	72	29	28	3
21	Arrhythmia	UCI	normal	258	119	118	12

# k-Nearest Neighbor-based Novelty Detection

- Performance (in terms of the Integrated Error)

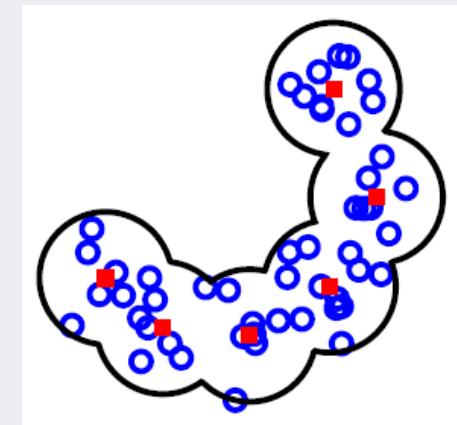
Data	Dim.	TrN <sub>n</sub>	Gauss	MoG	Parzen	1-SVM	KMC	KCC	HC	PCA	$d_{max}^k$	$d_{avg}^k$	$d_{mean}^k$	1-NN	MST-CD	$d_{hybrid}^k$
Titanic	3	100	19.12	19.12	18.50	17.27	21.12	21.26	22.80	22.48	3.64	3.53	10.00	8.73	<b>1.33*</b>	<b>1.33*</b>
Liver	6	73	44.41	45.04	41.11	38.82	41.90	43.41	40.91	40.68	40.00	38.85	38.96	39.81	39.14	<b>38.18</b>
Ecoli	7	72	3.61	2.58	3.14	2.35	2.45	3.38	3.88	20.12	2.22	2.13	2.84	3.94	2.67	<b>2.11</b>
Glass	9	35	18.82	18.29	22.25	17.43	18.61	22.61	24.42	22.21	13.86	12.25	12.36	18.93	11.54	<b>11.39</b>
Breast	9	142	35.83	31.51	32.04	29.64	34.68	40.68	32.80	31.24	29.58	28.84	29.53	34.62	33.18	<b>26.99*</b>
Banana	2	216	54.91	8.56	7.96	8.30	16.93	12.53	<b>7.65</b>	42.55	8.51	8.08	9.87	10.57	11.67	7.89
Yeast	8	232	31.55	28.98	27.99	26.58	28.78	33.25	27.32	31.47	25.81	24.54	25.87	27.79	26.50	<b>23.40</b>
Pima	8	250	29.86	33.55	26.04	27.50	29.60	32.69	28.46	33.28	24.82	24.57	27.68	28.17	27.72	<b>24.45</b>
Diabetes	8	304	30.61	34.68	27.35	26.60	28.80	35.31	26.45	31.32	23.70	23.29	25.68	26.66	28.21	<b>23.64</b>
Flare	9	300	23.19	23.19	24.82	15.40	29.67	26.65	25.57	26.76	10.49	9.74	17.09	5.62	<b>5.47</b>	6.14
Spectf	44	28	28.33	16.42	21.11	14.75	15.00	28.02	16.36	26.30	14.20	13.40	12.96	17.16	13.33	<b>11.67*</b>
Sonar	60	56	41.59	37.27	34.32	33.80	41.55	40.06	40.07	42.32	39.65	34.67	32.12	33.73	<b>31.30</b>	32.62
Ozone	72	29	23.99	14.64	13.15	12.50	13.51	19.11	16.49	34.46	11.07	10.71	<b>9.76</b>	14.11	12.86	10.30
Arrhythmia	258	119	28.01	40.25	28.03	25.79	25.38	28.62	27.42	28.17	26.10	26.01	25.93	26.16	24.57	<b>23.93</b>
Heart	13	94	21.13	19.05	20.58	18.61	19.84	20.79	18.22	20.21	15.23	14.75	15.72	23.08	23.24	<b>14.30</b>
Image	18	554	13.11	13.61	11.79	10.38	23.22	30.13	31.21	15.87	13.80	11.99	10.32	10.53	<b>9.19*</b>	11.04
Twonorm	20	198	9.83	11.80	11.62	9.48	8.82	12.38	<b>6.13*</b>	9.62	9.62	10.02	10.79	12.87	12.62	10.11
German	20	489	38.16	36.99	37.24	35.72	39.23	42.79	40.56	37.73	34.72	33.95	34.59	37.88	36.35	<b>33.77</b>
Waveform	21	268	42.14	30.25	26.33	<b>22.65*</b>	27.56	31.26	25.44	43.07	23.75	24.58	27.69	29.05	27.09	25.24
Parkinsons	22	74	32.71	46.35	32.14	<b>29.16</b>	32.65	32.66	32.79	34.60	36.95	32.54	31.63	34.22	30.68	30.30
Ionosphere	33	113	4.44	4.93	4.16	2.80	3.54	4.40	4.39	3.68	2.70	2.76	2.75	4.00	<b>2.70</b>	2.72

# Clustering-based Approach

- K-Means clustering-based novelty detection
  - ✓ Novelty score of an instance is computed based on the distance information to the nearest centroid
  - ✓ Does not assume any prior probability distribution for the normal class

$$\mathcal{X} = C_1 \cup C_2 \dots \cup C_K, \quad C_i \cap C_j = \emptyset, \quad i \neq j.$$

$$\arg \min_{\mathbf{C}} \sum_{i=1}^K \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \mathbf{c}_i\|^2$$



- ✓ EM algorithm for K-Means clustering

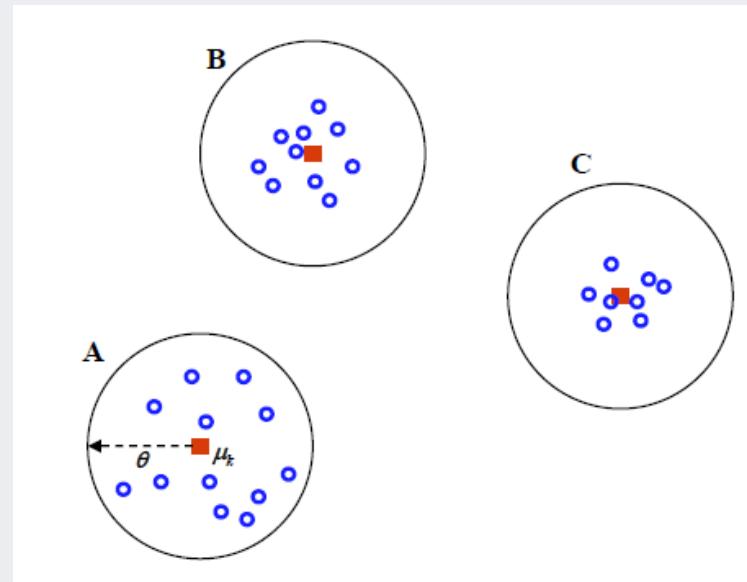
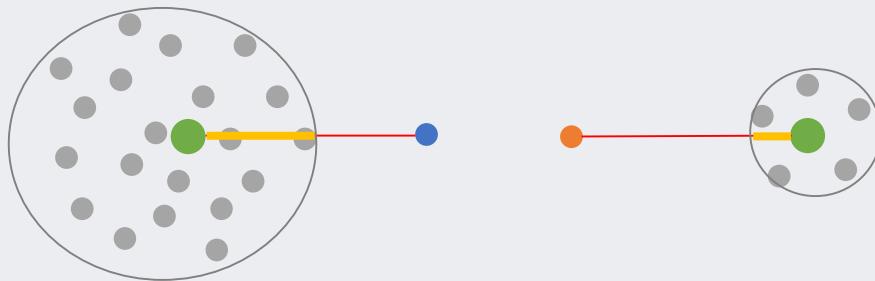
- 1: Select  $K$  points as the initial centroids.
- 2: **repeat**
- 3:   Form  $K$  clusters by assigning all points to the closest centroid.
- 4:   Recompute the centroid of each cluster.
- 5: **until** The centroids don't change

# Clustering-based Approach

- Clustering-based Approach

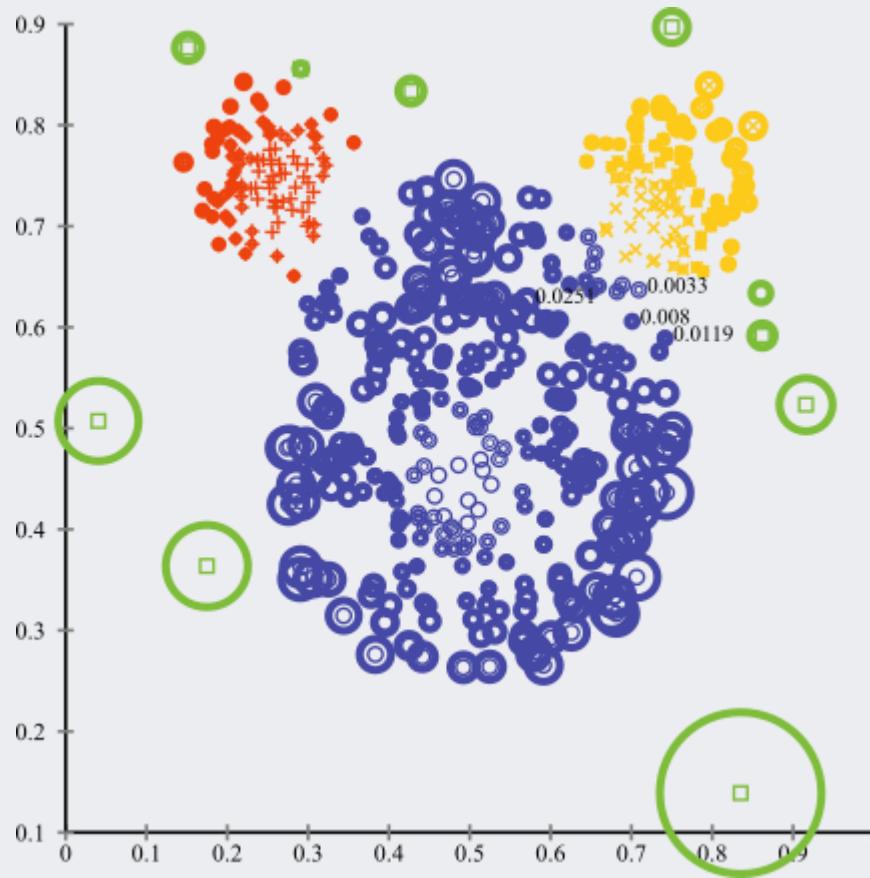
- ✓ Two novelty scores by KMC

- Absolute distance to the nearest centroid
- Relative distance to the nearest centroid



# Clustering-based Approach

- KMC-based novelty score: Example



# Principal Component Analysis-based Novelty Detection

- PCA revisited

✓ Purpose: maximize the variance after projection

$$\max \mathbf{w}^T \mathbf{S} \mathbf{w}$$

$$s.t. \quad \mathbf{w}^T \mathbf{w} = 1$$

✓ Solution

$$L = \mathbf{w}^T \mathbf{S} \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{w} - 1)$$

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{S} \mathbf{w} - \lambda \mathbf{w} = 0 \Rightarrow (\mathbf{S} - \lambda \mathbf{I}) \mathbf{w} = 0$$

# Principal Component Analysis-based Novelty Detection

- PCA as a novelty detector

- ✓ Novelty score: the amount of reconstruction loss from the projected space into the original space



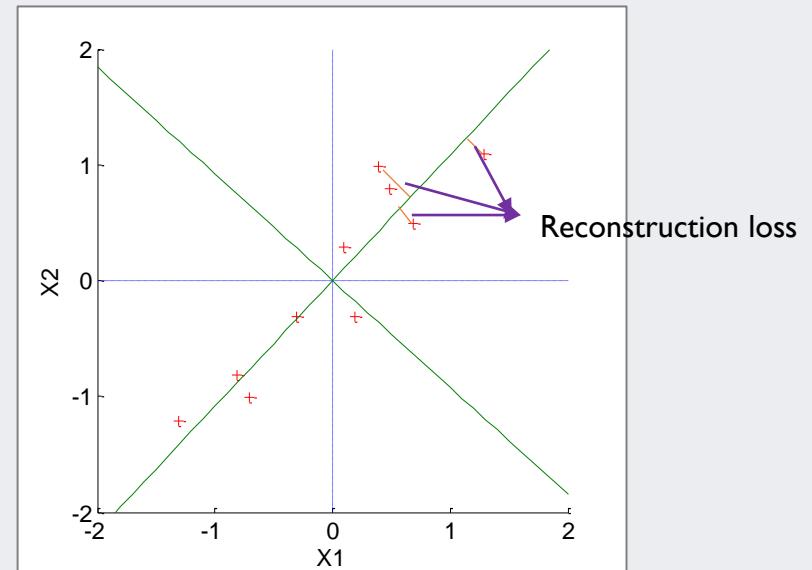
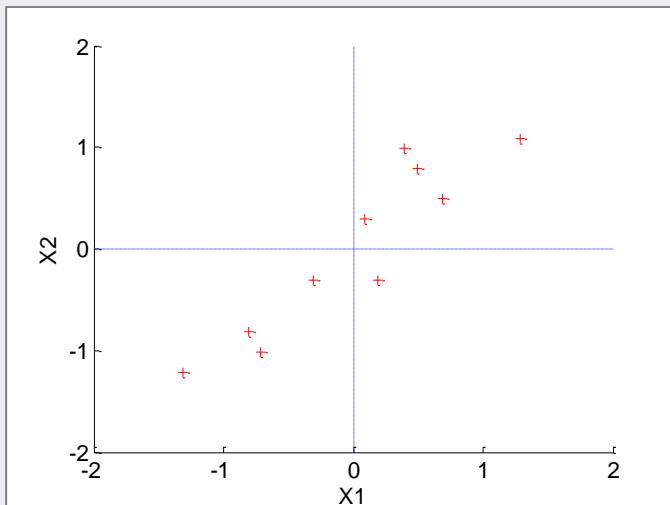
# Principal Component Analysis-based Novelty Detection

- PCA as a novelty detector
  - ✓ Compute the reconstruction loss

$$\text{error}(\mathbf{x}) = \|\mathbf{x} - \mathbf{w}\mathbf{w}^T\mathbf{x}\|^2 = (\mathbf{x} - \mathbf{w}\mathbf{w}^T\mathbf{x})^T(\mathbf{x} - \mathbf{w}\mathbf{w}^T\mathbf{x})$$

$$= \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{w}\mathbf{w}^T\mathbf{x} - \mathbf{x}^T\mathbf{w}\mathbf{w}^T\mathbf{x} + \mathbf{x}^T\mathbf{w}\mathbf{w}^T\mathbf{w}\mathbf{w}^T\mathbf{x}$$

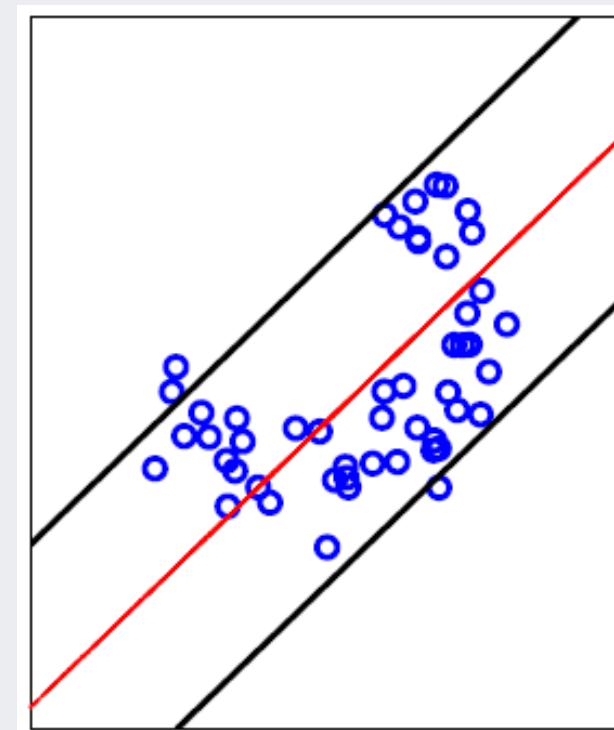
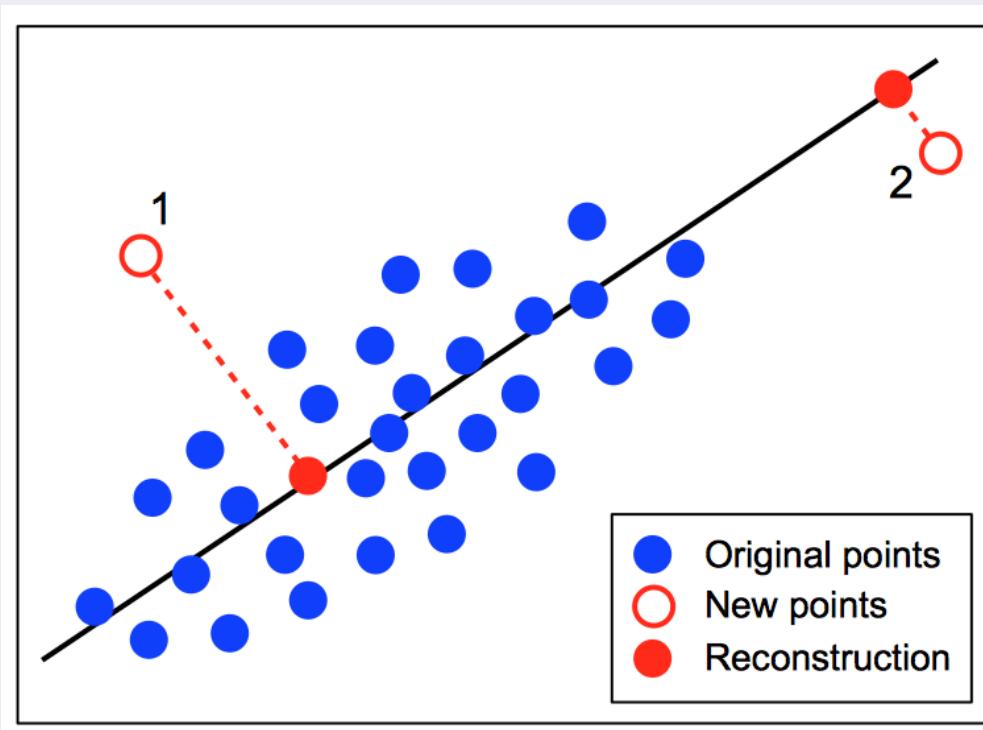
$$= \mathbf{x}^T\mathbf{x} - \mathbf{x}^T\mathbf{w}\mathbf{w}^T\mathbf{x} = \|\mathbf{x}\|^2 - \|\mathbf{w}^T\mathbf{x}\|^2$$



# Principal Component Analysis-based Novelty Detection

- PCA as a novelty detector

- ✓ Graphical interpretation



<https://stats.stackexchange.com/questions/259806/anomaly-detection-using-pca-reconstruction-error>

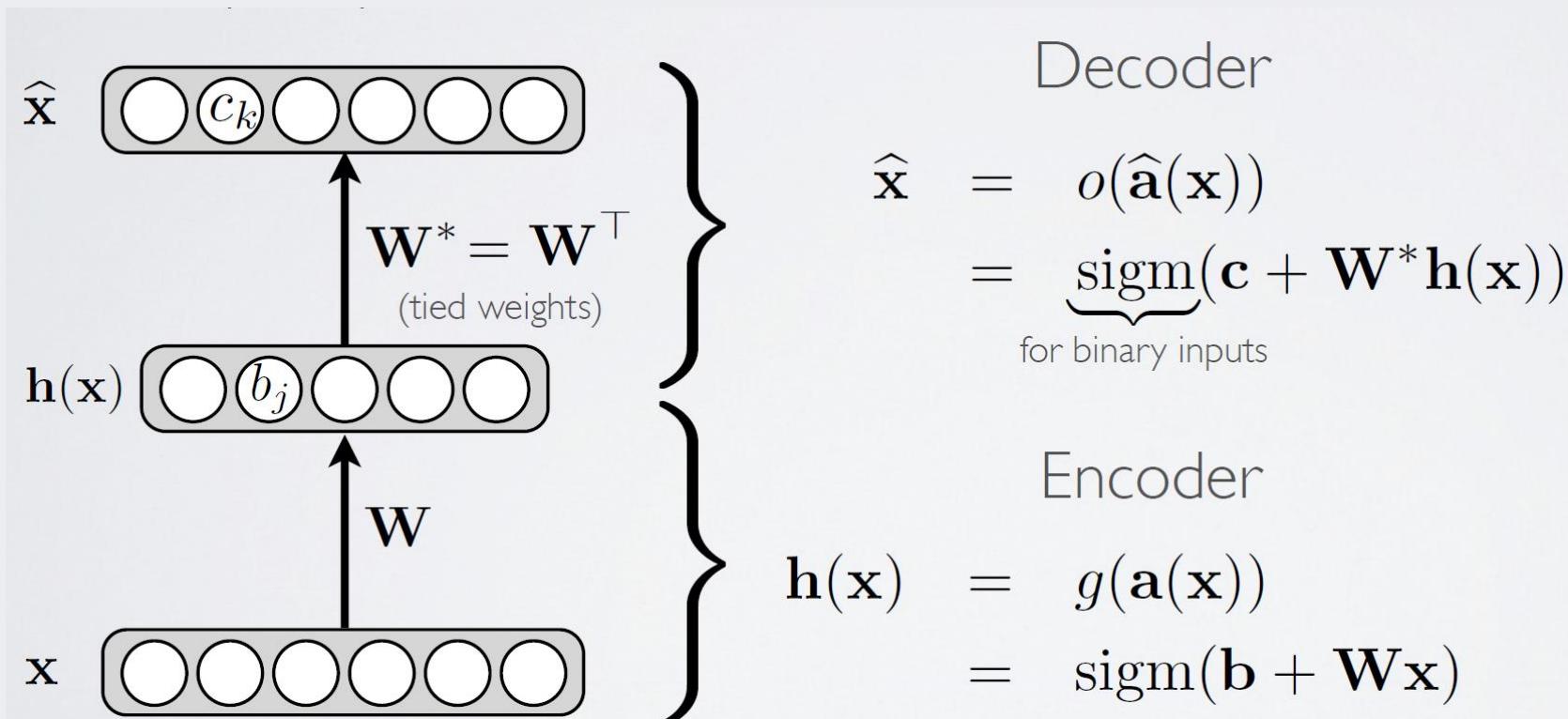
# AGENDA

- 01 Novelty Detection: Overview
- 02 Density-based Novelty Detection
- 03 Distance/Reconstruction-based ND
- 04 Model-based Novelty Detection

# Auto-Encoder for Novelty Detection

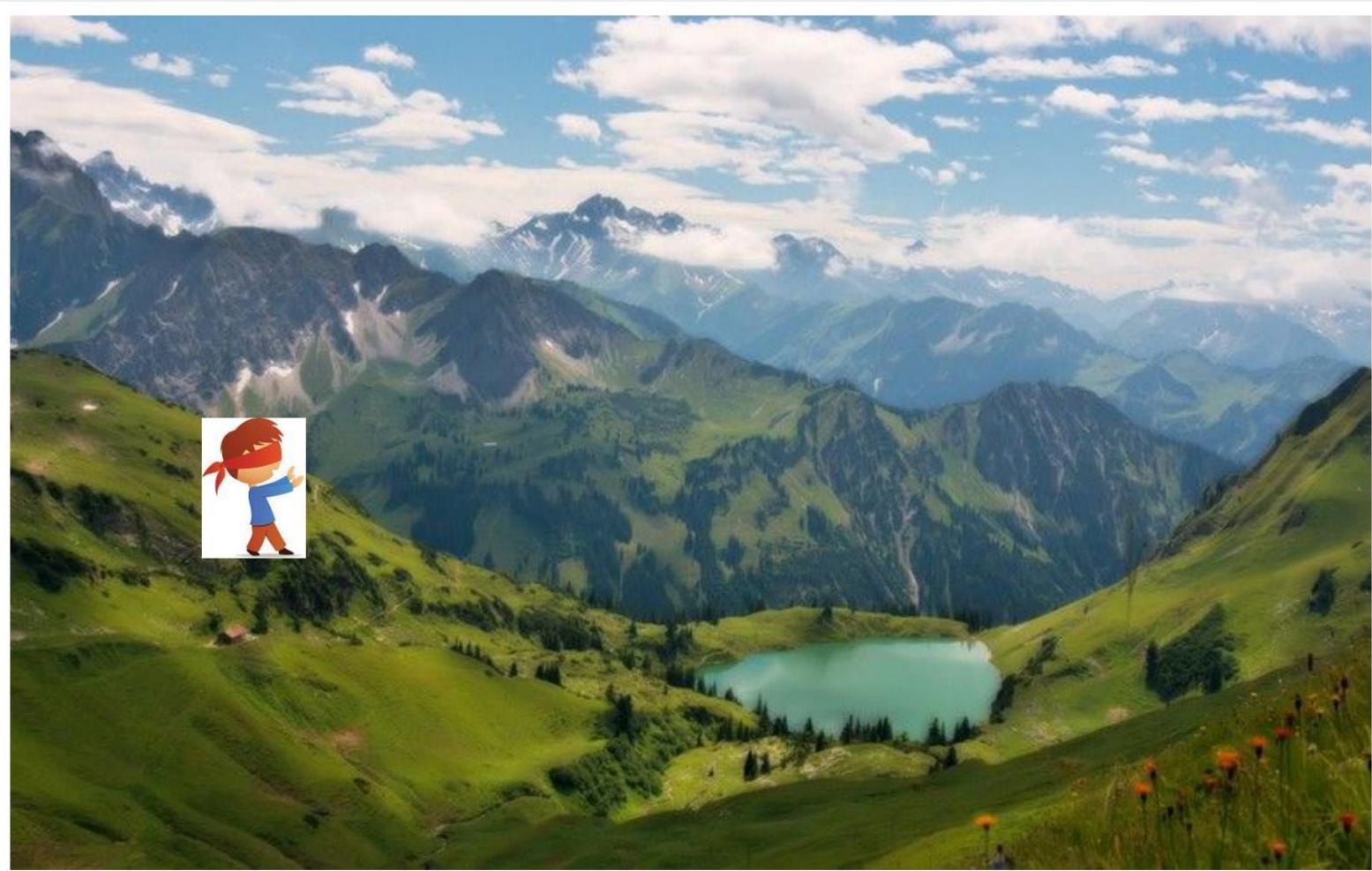
- Auto-Encoder (Auto-Associative Neural Network)
  - ✓ Feed-forward neural network trained to **reproduce** its input at the output layer

- Loss function:  $l(f(\mathbf{x})) = \frac{1}{2} \sum_k (\hat{x}_k - x_k)^2$



# Auto-Encoder for Novelty Detection

- Auto-Encoder (Auto-Associative Neural Network)
  - ✓ Training: Gradient Descent

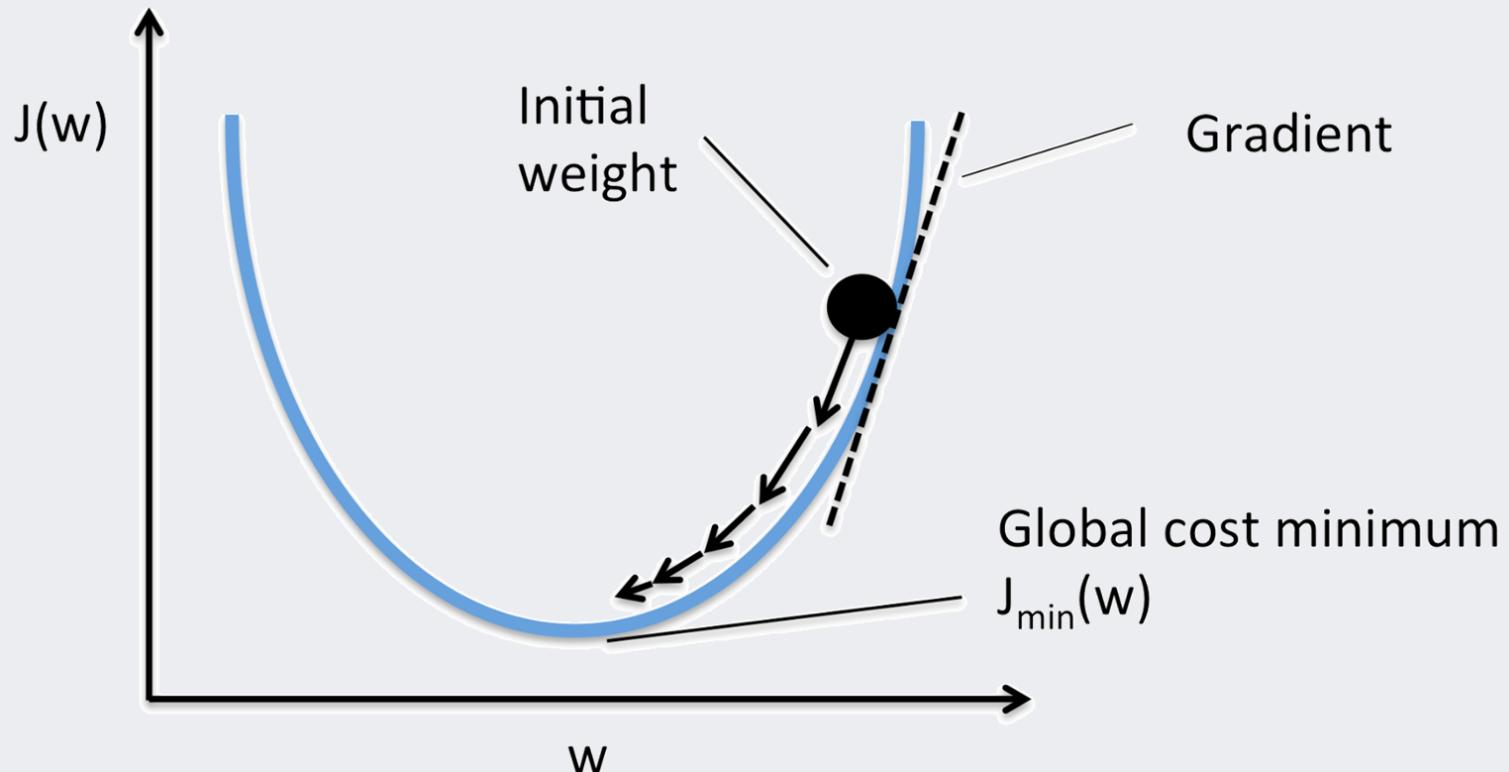


# Auto-Encoder for Novelty Detection

- Auto-Encoder (Auto-Associative Neural Network)

- ✓ Training: Gradient Descent

- Blue line: the value of the objective function w.r.t.  $w$
- Black point: current position
- Arrow: the direction that  $w$  should move toward to minimize the objective function



# Auto-Encoder for Novelty Detection

- Gradient Descent

- ✓ Taylor expansion of a function

$$f(x + \Delta x) = f(x) + \frac{f'(x)}{1!} \Delta x + \frac{f''(x)}{2!} (\Delta x)^2 + \dots$$

- ✓ For the minimization, the objective function decreases by moving the  $x$  toward the opposite direction of the first derivative if the value of first derivative is not zero.

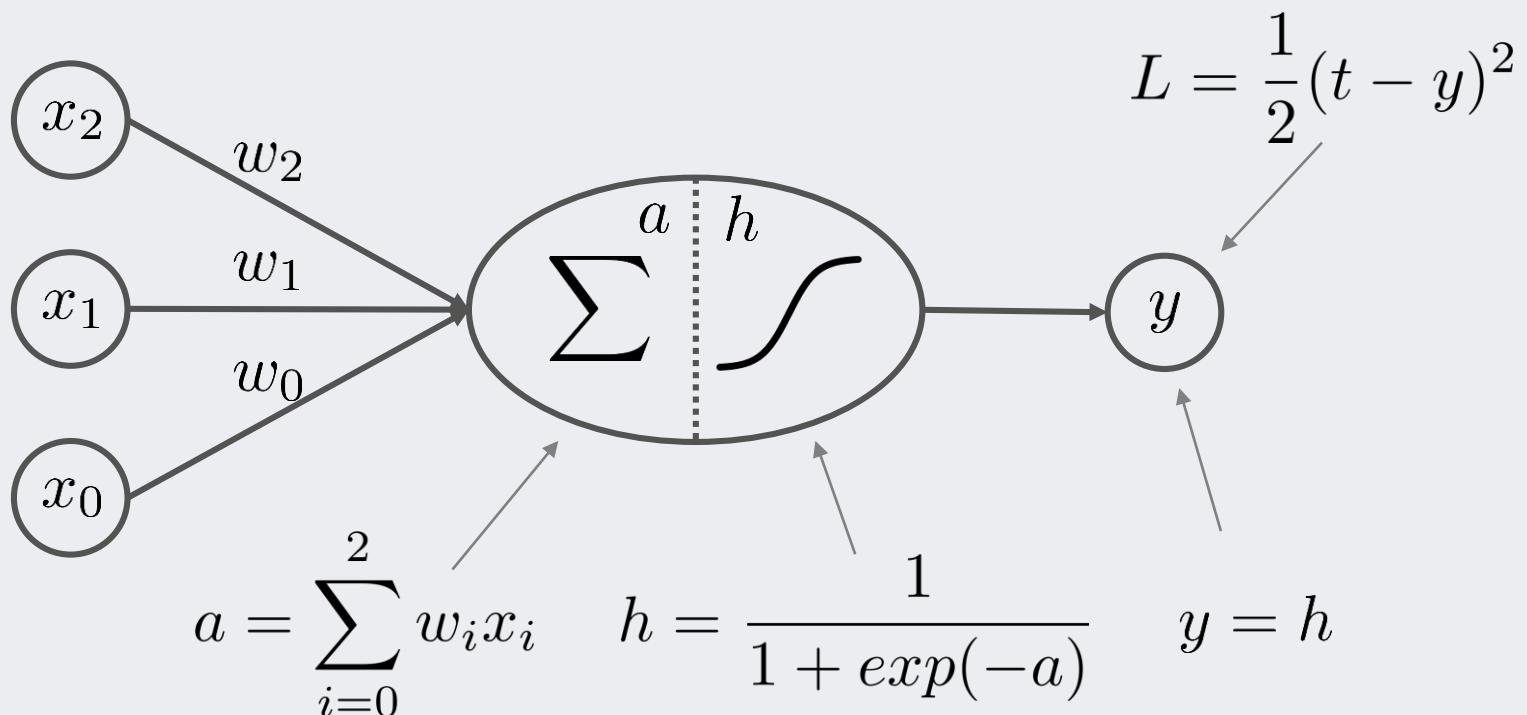
$$x_{new} = x_{old} - \eta f'(x), \quad \text{where } 0 < \eta < 1.$$

$$f(x_{new}) = f(x_{old} - \eta f'(x_{old})) \cong f(x_{old}) - \eta |f'(x)|^2 < f(x_{old})$$

# Auto-Encoder for Novelty Detection

- Gradient Descent Example with Perceptron

원하는 값( $t$ )과 예측값( $y$ )의 차이를 손실함수로 정의



여러 변수들의 정보를  
나름대로 취합해서

얼마만큼 다음 단계로  
전달할지 결정한다  
(활성화)

# Auto-Encoder for Novelty Detection

- Gradient Descent Example with Perceptron

✓ Chain rule

$$\frac{\partial L}{\partial y} = y - t \quad \frac{\partial y}{\partial h} = 1$$

$$\frac{\partial h}{\partial a} = \frac{exp(-a)}{(1 + exp(-a))^2} = \frac{1}{1 + exp(-a)} \cdot \frac{exp(-a)}{1 + exp(-a)} = h(1 - h)$$

$$\frac{\partial a}{\partial w_i} = x_i$$

✓ Gradient for w

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial h} \cdot \frac{\partial h}{\partial a} \cdot \frac{\partial a}{\partial w_i} = (y - t) \cdot 1 \cdot h(1 - h) \cdot x_i$$

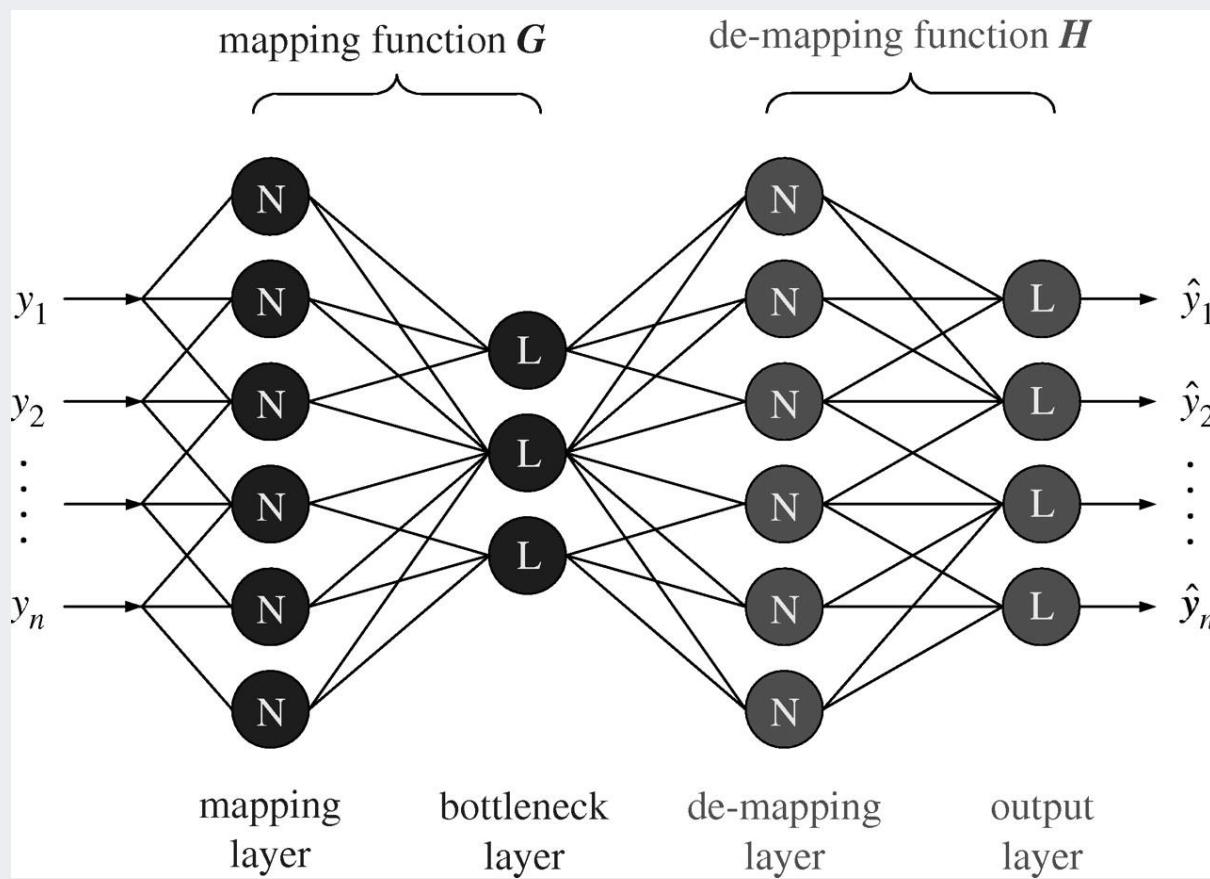
✓ Update w

$$w_i^{new} = w_i^{old} - \eta \times \frac{\partial L}{\partial w_i^{old}} = w_i^{old} - \eta \times (y - t) \cdot 1 \cdot h(1 - h) \cdot x_i$$

# Auto-Encoder for Novelty Detection

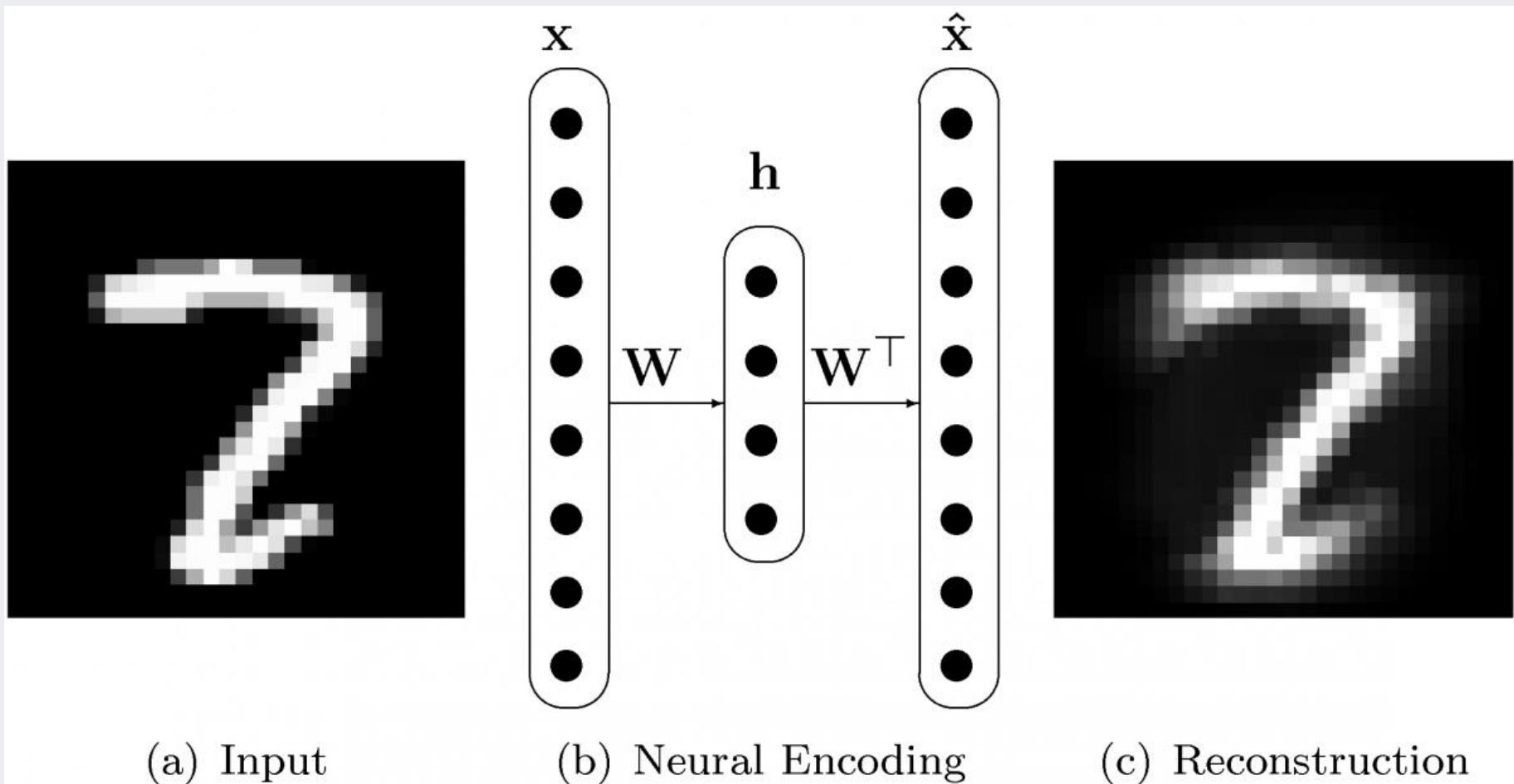
- Auto Encoder (Auto-Associative Neural Network)

- ✓ Feed-forward neural network trained to **reproduce** its input at the output layer
- ✓ Overcomplete and Undercomplete hidden layers for AE



# Auto-Encoder for Novelty Detection

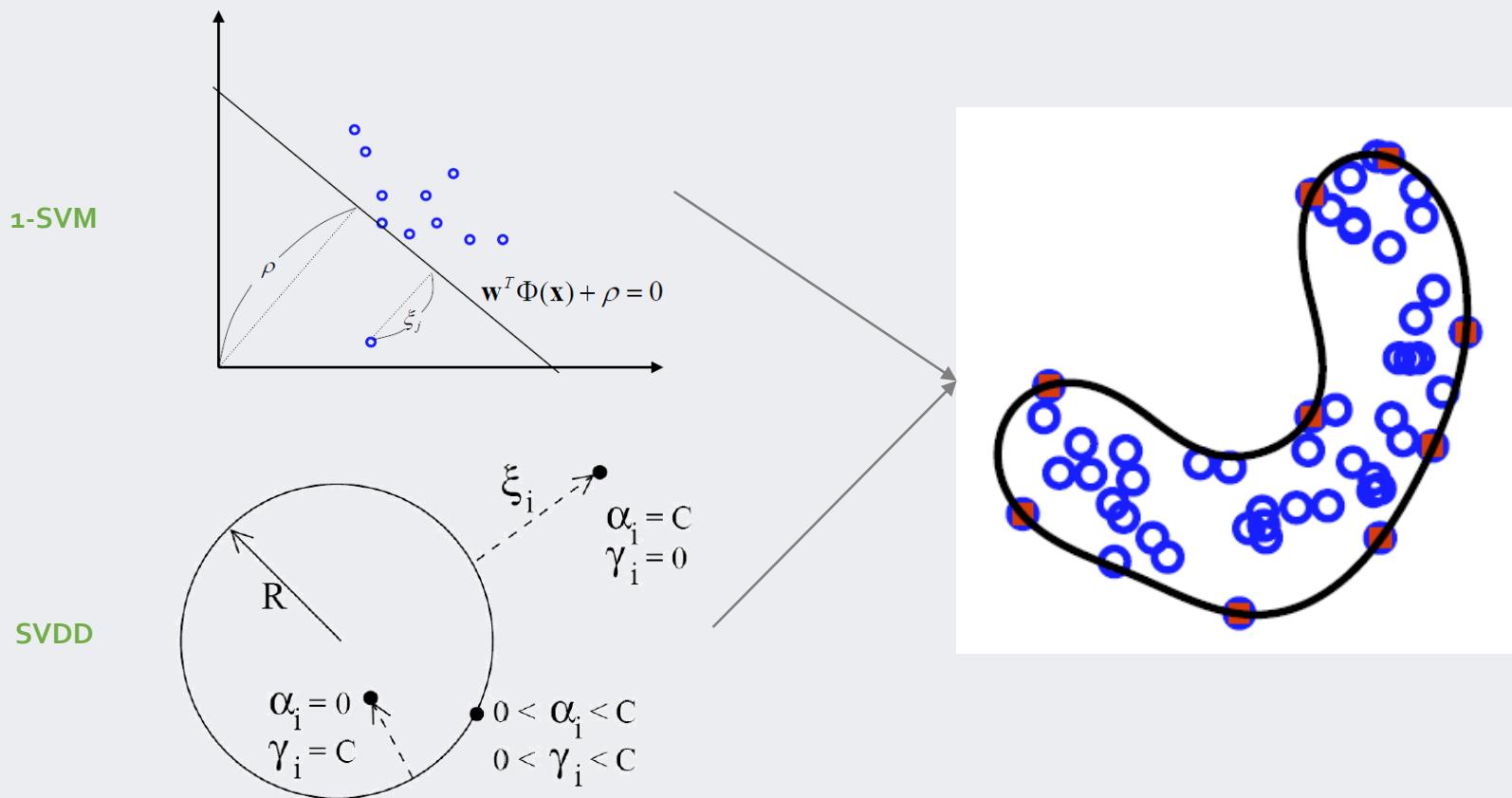
- Auto Encoder (Auto-Associative Neural Network)
  - ✓ Example



# Support Vector-based Novelty Detection

- Support vector-based novelty detection

- ✓ Define boundaries of normal regions directly by finding function that separates the normal and abnormal observations



# One-Class Support Vector Machine

Scholkopf et al. (2001)

- One-class support vector machine (1-SVM)

- ✓ Map the data into the feature space corresponding to the kernel and to separate them from the origin with maximum margin

- Optimization problem

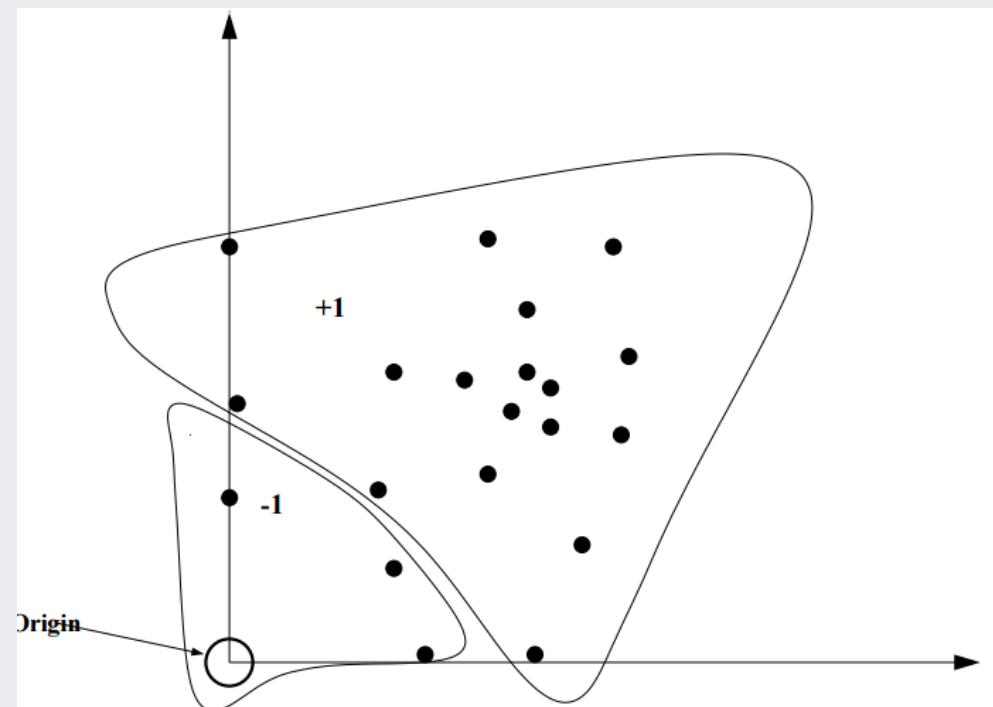
$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu l} \sum_{i=1}^l \xi_i - \rho$$

$$s.t. \quad \mathbf{w} \cdot \Phi(\mathbf{x}_i) \geq \rho - \xi_i$$

$$i = 1, 2, \dots, l, \quad \xi_i \geq 0$$

- Decision function

$$f(\mathbf{x}_i) = \text{sign}(\mathbf{w} \cdot \Phi(\mathbf{x}_i) - \rho)$$



# One-Class Support Vector Machine

- One-class support vector machine (1-SVM)

✓ Primal Lagrangian problem (Minimize)

$$L = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu l} \sum_{i=1}^l \xi_i - \rho - \sum_{i=1}^l \alpha_i (\mathbf{w} \cdot \Phi(\mathbf{x}_i) - \rho + \xi_i) - \sum_{i=1}^l \beta_i \xi_i$$

✓ KKT condition

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^l \alpha_i \Phi(\mathbf{x}_i) = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^l \alpha_i \Phi(\mathbf{x}_i)$$

$$\frac{\partial L}{\partial \xi_i} = \frac{1}{\nu l} - \alpha_i - \beta_i = 0 \quad \Rightarrow \quad \alpha_i = \frac{1}{\nu l} - \beta_i$$

$$\frac{\partial L}{\partial \rho} = -1 + \sum_{i=1}^l \alpha_i = 0 \quad \Rightarrow \quad \sum_{i=1}^l \alpha_i = 1$$

# One-Class Support Vector Machine

- One-class support vector machine (1-SVM)

✓ Dual Lagrangian problem (Maximize)

$$L = \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j) + \frac{1}{\nu l} \sum_{i=1}^l \xi_i - \rho$$

$$- \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j) + \rho \sum_{i=1}^l \alpha_i - \sum_{i=1}^l \alpha_i \xi_i - \sum_{i=1}^l \beta_i \xi_i$$

✓ We should solve

$$\min L = \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j)$$

$$s.t. \quad \sum_{i=1}^l \alpha_i = 1, \quad 0 \leq \alpha_i \leq \frac{1}{\nu l}$$

# One-Class Support Vector Machine

- One-class support vector machine (1-SVM)

✓ Employ Kernel Trick for a non-linear mapping

$$\min L = \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j)$$

$$s.t. \quad \sum_{i=1}^l \alpha_i = 1, \quad 0 \leq \alpha_i \leq \frac{1}{\nu l}$$



$$\min L = \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$s.t. \quad \sum_{i=1}^l \alpha_i = 1, \quad 0 \leq \alpha_i \leq \frac{1}{\nu l}$$

Some possible kernels  $K(\cdot, \cdot)$ :

$$K(x, x_i) = x_i^T x \text{ (linear SVM)}$$

$$K(x, x_i) = (x_i^T x + \tau)^d \text{ (polynomial SVM of degree } d\text{)}$$

$$K(x, x_i) = \exp(-\|x - x_i\|_2^2 / \sigma^2) \text{ (RBF kernel)}$$

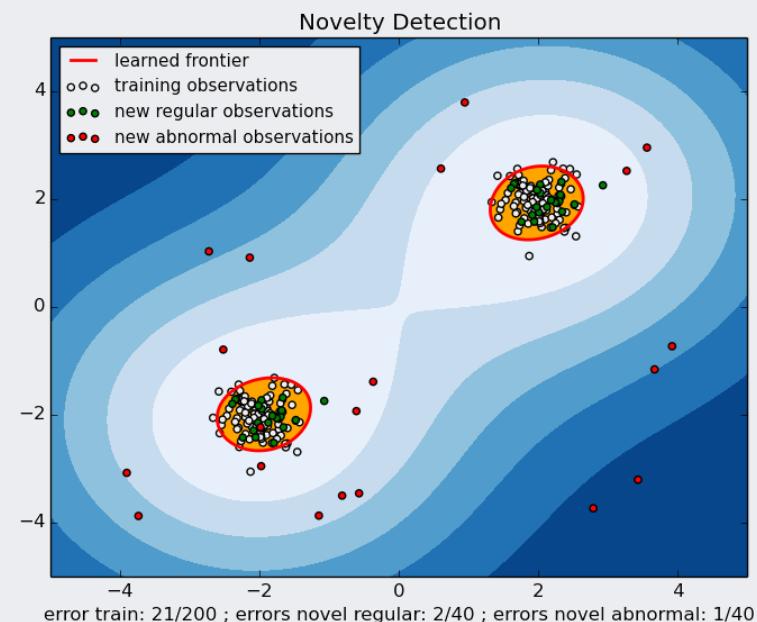
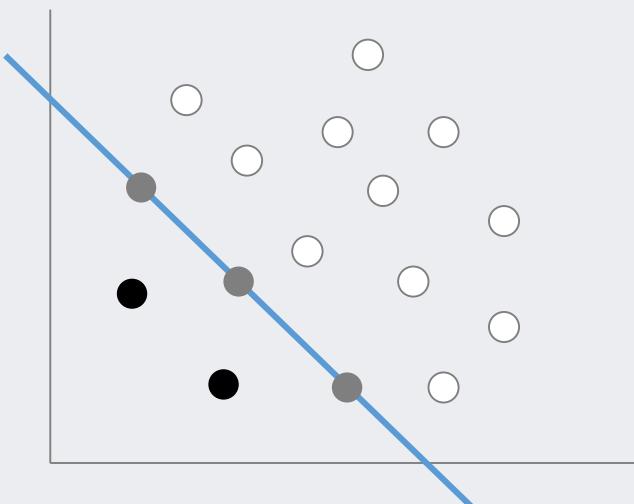
$$K(x, x_i) = \tanh(\kappa x_i^T x + \theta) \text{ (MLP kernel)}$$

# One-Class Support Vector Machine

- One-class support vector machine (1-SVM)

- ✓ Location of a point w.r.t.  $\alpha_i$

- ▪ Case 1:  $\alpha_i = 0 \Rightarrow$  a non-support vector
- ▪ Case 2:  $\alpha_i = \frac{1}{\nu l} \Rightarrow \beta_i = 0 \Rightarrow \xi_i > 0 \Rightarrow$  Support vector (outsider the hyperplane)
- ▪ Case 3:  $0 < \alpha_i < \frac{1}{\nu l} \Rightarrow \beta_i > 0 \Rightarrow \xi_i = 0 \Rightarrow$  Support vector (on the hyperplane)



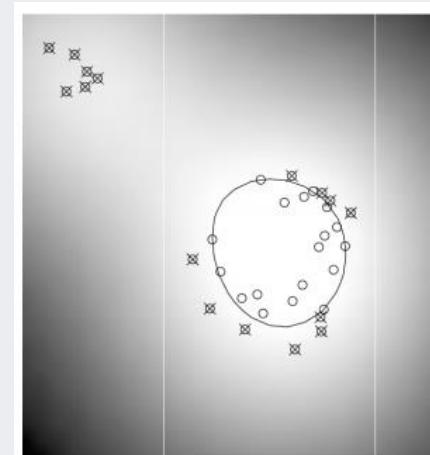
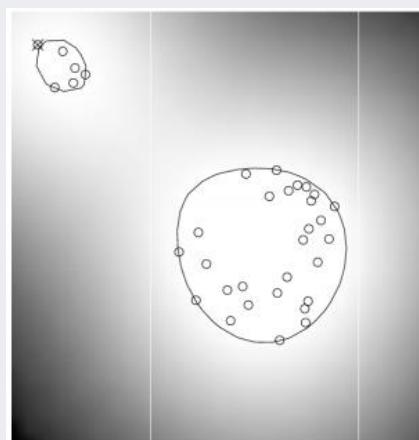
# One-Class Support Vector Machine

- One-class support vector machine (1-SVM)

- ✓ The role of  $\nu$

- The maximum possible value of  $\alpha_i = \frac{1}{\nu l}$
    - At least  $\nu l$  support vectors exist
    - At most  $\nu l$  support vectors can be located outside the hyperplane
    - Thus,  $\nu$  is the **lower bound for the fraction of support vectors** and the **upper bound for the fraction of errors**

- ✓ The higher the  $\nu$ , the more complex decision boundary is generated

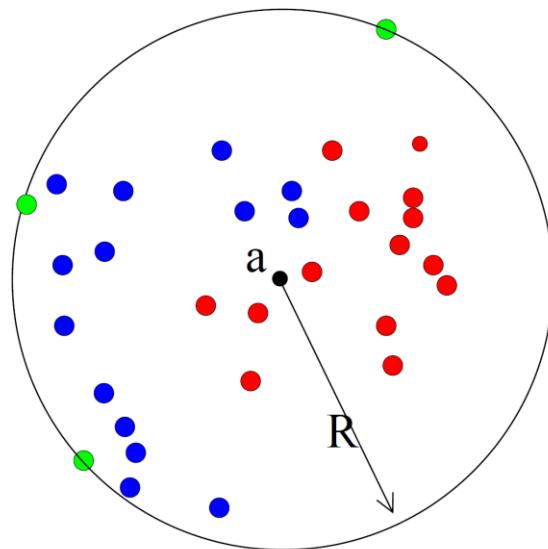


# Support Vector Data Description

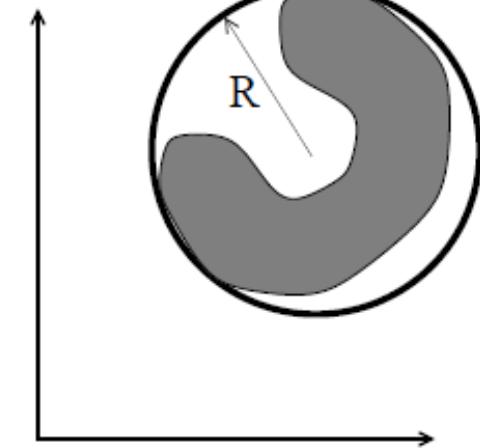
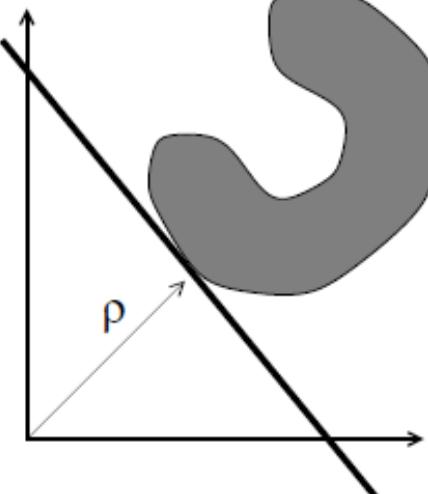
Tax and Duin (2004)

- Support Vector Data Description (SVDD)
  - ✓ Find a hypersphere enclosing all the normal instances in a feature space

SVDD



1-SVM vs. SVDD



# Support Vector Data Description

- Support Vector Data Description (SVDD)
  - ✓ Find a hypersphere enclosing all the normal instances in a feature space
    - Optimization function

$$\min_{R, \mathbf{a}, \xi_i} R^2 + C \sum_{i=1}^l \xi_i$$

$$s.t. \quad ||\Phi(\mathbf{x}_i) - \mathbf{a}||^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \quad \forall i.$$

- Decision function

$$f(\mathbf{x}) = sign(R^2 - ||\Phi(\mathbf{x}_i) - \mathbf{a}||^2)$$

# Support Vector Data Description

- Support Vector Data Description (SVDD)

- ✓ Find a hypersphere enclosing all the normal instances in a feature space

- Primal Lagrangian problem (Minimization)

$$L = R^2 + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i \left( R^2 + \xi_i - (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) - 2 \cdot \mathbf{a} \cdot \Phi(\mathbf{x}_i) + \mathbf{a} \cdot \mathbf{a}) \right) - \sum_{i=1}^l \beta_i \xi_i$$
$$\alpha_i \geq 0, \quad \beta_i \geq 0$$

- KKT condition

$$\frac{\partial L}{\partial R} = 2R - 2R \sum_{i=1}^l \alpha_i = 0 \quad \Rightarrow \quad \sum_{i=1}^l \alpha_i = 1$$

$$\frac{\partial L}{\partial \mathbf{a}} = 2 \sum_{i=1}^l \alpha_i \cdot \Phi(\mathbf{x}_i) - 2\mathbf{a} \sum_{i=1}^l \alpha_i = 0 \quad \Rightarrow \quad \mathbf{a} = \sum_{i=1}^l \alpha_i \cdot \Phi(\mathbf{x}_i)$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \quad \forall i$$

# Support Vector Data Description

- Support Vector Data Description (SVDD)

- ✓ Find a hypersphere enclosing all the normal instances in a feature space
  - Dual Lagrangian problem (Maximization)

$$L = R^2 + C \sum_{i=1}^l \xi_i - \sum_{i=1}^l \alpha_i \left( R^2 + \xi_i - (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) - 2 \cdot \mathbf{a} \cdot \Phi(\mathbf{x}_i) + \mathbf{a} \cdot \mathbf{a}) \right) - \sum_{i=1}^l \beta_i \xi_i$$



$$L = R^2 - R^2 \sum_{i=1}^l \alpha_i + \sum_{i=1}^l \xi_i (C - \alpha_i - \beta_i)$$

$$+ \sum_{i=1}^l \alpha_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) - 2 \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j) + \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j)$$



$$L = \sum_{i=1}^l \alpha_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) - \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j) \quad (0 \leq \alpha_i \leq C)$$

# Support Vector Data Description

- Support Vector Data Description (SVDD)

- ✓ Find a hypersphere enclosing all the normal instances in a feature space

- Dual Lagrangian problem (Maximization)

$$L = \sum_{i=1}^l \alpha_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) - \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j) \quad (0 \leq \alpha_i \leq C)$$

- Dual Lagrangian problem (Minimization)

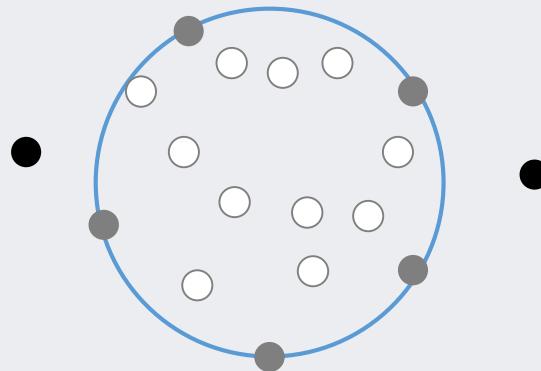
$$L = \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j) - \sum_{i=1}^l \alpha_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_i) \quad (0 \leq \alpha_i \leq C)$$

# Support Vector Data Description

- Support Vector Data Description (SVDD)

- ✓ Location of a point w.r.t.  $\alpha_i$

- ▪ Case 1:  $\alpha_i = 0 \Rightarrow$  a non-support vector
    - ▪ Case 2:  $\alpha_i = C \Rightarrow \beta_i = 0 \Rightarrow \xi_i > 0 \Rightarrow$  Support vector (outsider the hypersphere)
    - ▪ Case 3:  $0 < \alpha_i < C \Rightarrow \beta_i > 0 \Rightarrow \xi_i = 0 \Rightarrow$  Support vector (on the hypersphere)

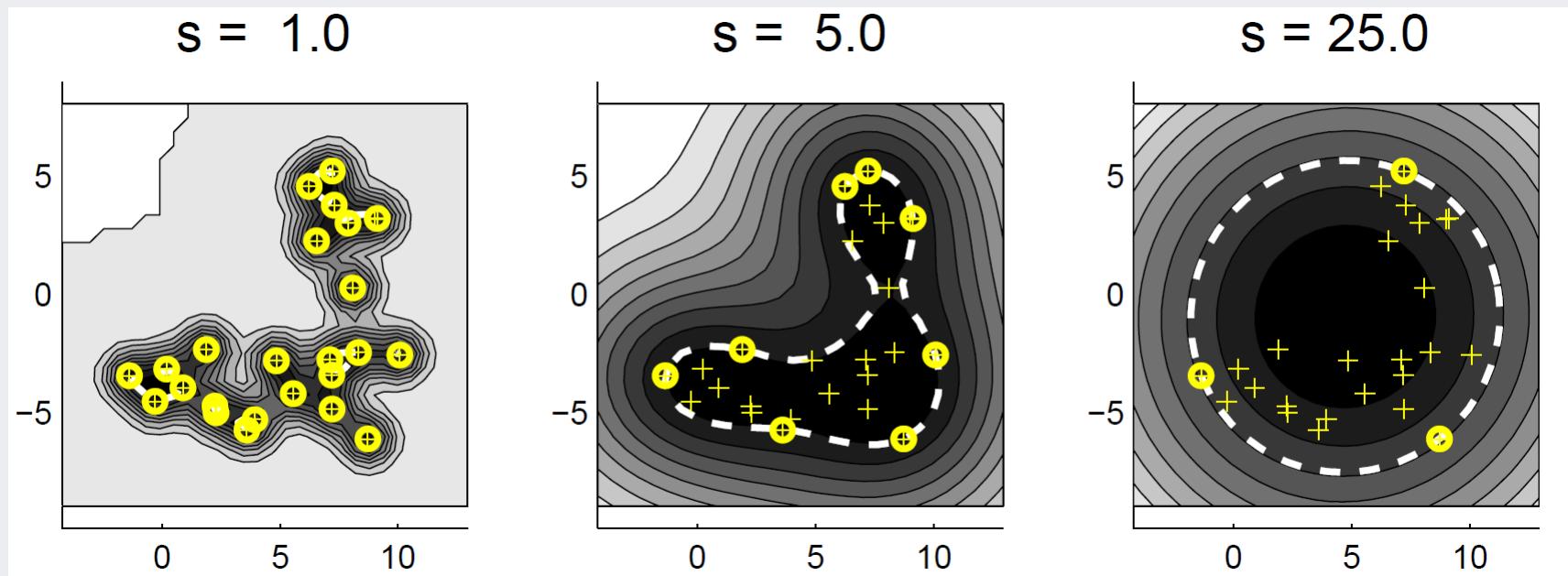


# Support Vector Data Description

- Support Vector Data Description (SVDD)

- ✓ SVDD with Gaussian (RBF) kernels

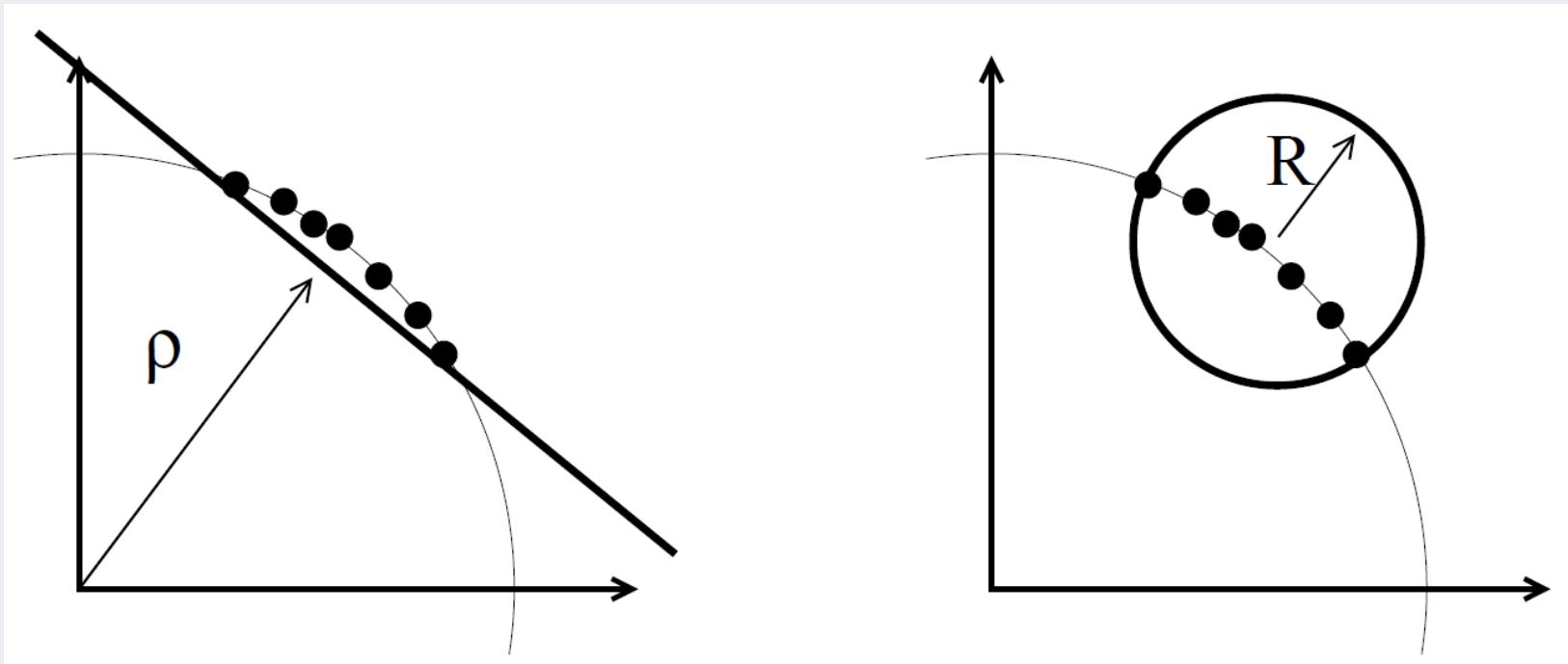
$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{s^2}\right)$$



# Support Vector Data Description

- Support Vector Data Description (SVDD)

- ✓ When all data is normalized to unit norm vector, SVDD and 1-SVM are equivalent

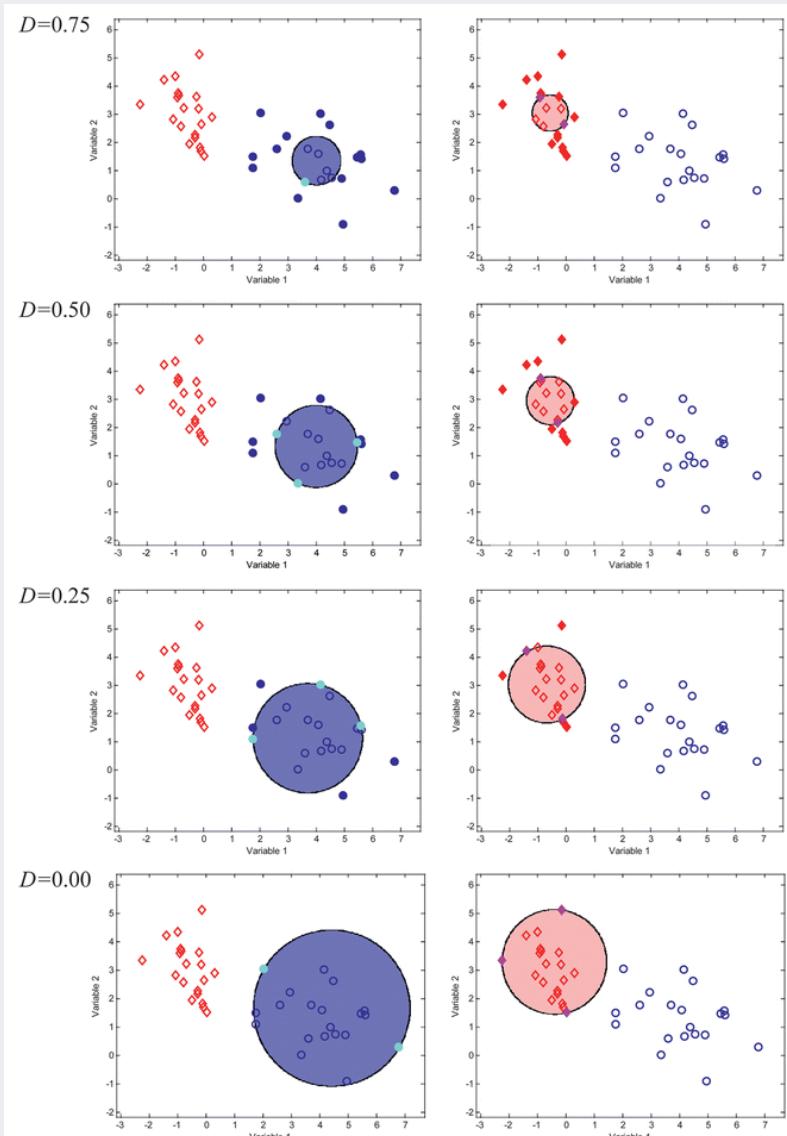


- For the detailed proof, please refer to Tax (2001) pp. 39-41.

# Support Vector Data Description

- Support Vector Data Description (SVDD)

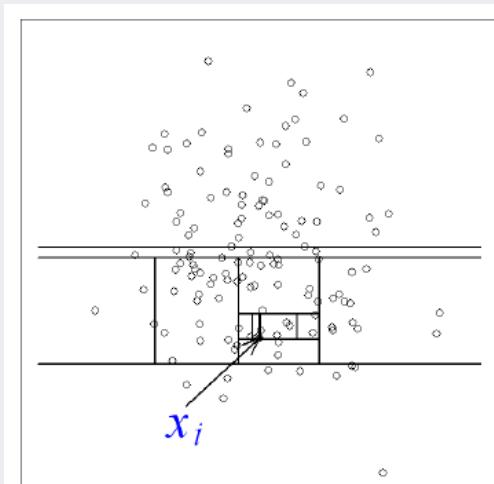
- ✓ As in 1-SVM,  $\nu$ -SVDD can also be formulated
- ✓ D in the right figure is  $\nu$



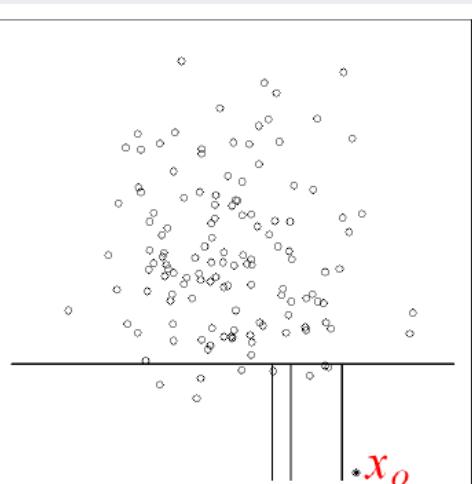
# Isolation Forest

Liu et al. (2008, 2012)

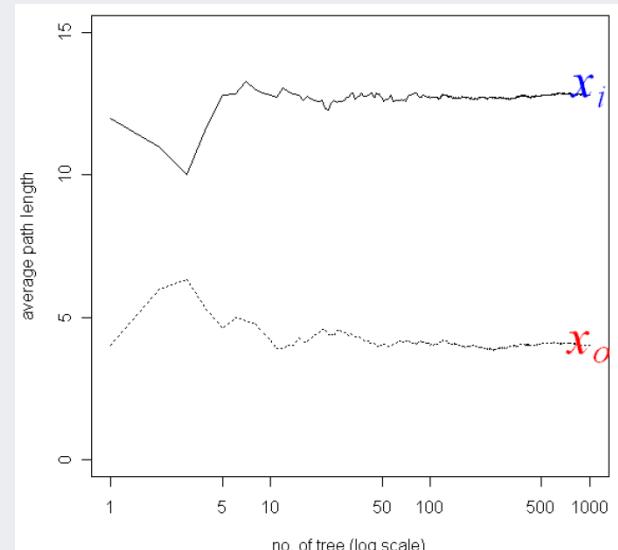
- Motivation: Few and Different
  - ✓ The minority consists of fewer instances
  - ✓ They have attribute-values, which are very different from those of normal instances
- A tree structure can be constructed effectively to **isolate** every single instances
  - ✓ Novel instances are isolated closer to the root of the tree
  - ✓ Normal instances are isolated at the deeper end of the tree



(a) Isolating  $x_i$

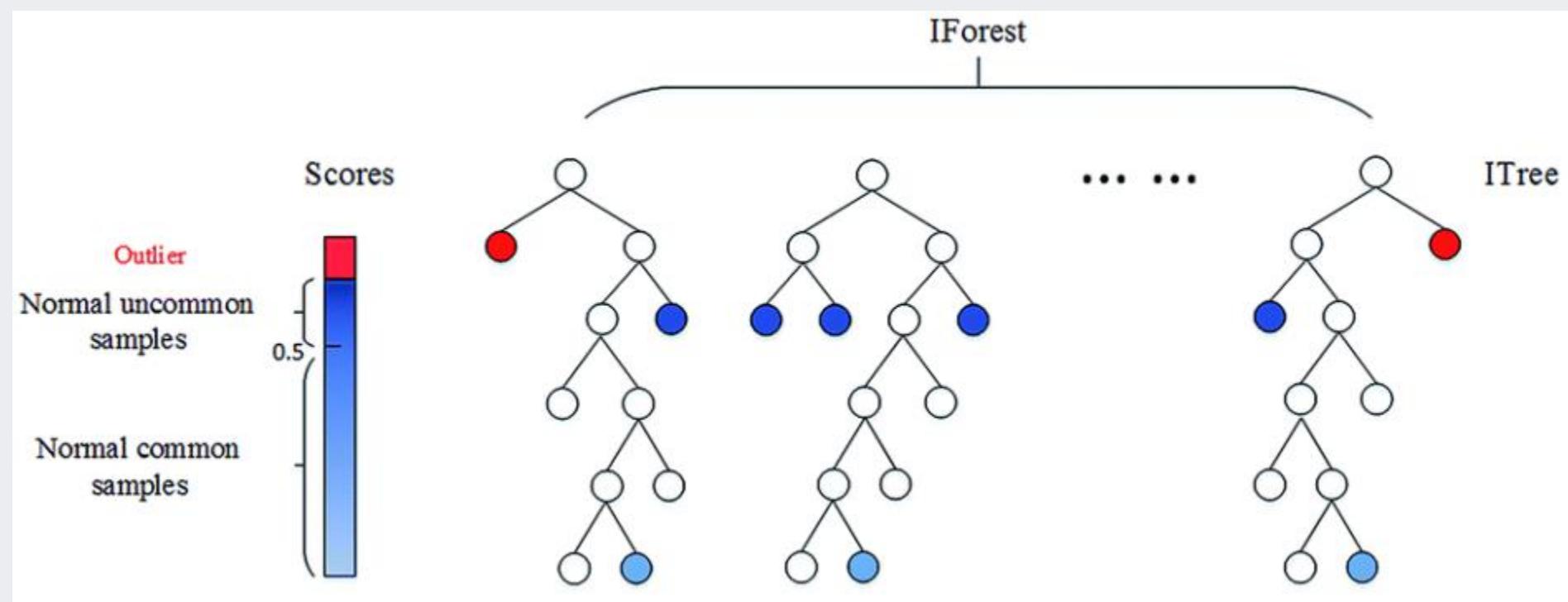


(b) Isolating  $x_o$



# Isolation Forest

- The isolation characteristics of tree forms the basis of the method to detect novel instances
  - ✓ The average path to the terminal node can be used as a novelty score of an instance



# Isolation Forest

- Definition: Isolation Tree (iTree)
  - ✓ Given a sample of data  $X$  of  $n$  instances, the dataset  $X$  is recursively divided by randomly selected attribute  $q$  with a split value  $p$ , until either
    - The tree reaches a height limit
    - $|X| = 1$
    - All instances in  $X$  have the same value
- Definition: Path Length
  - ✓ The path length  $h(x)$  of an instance  $x$  is measured by the number of edges  $x$  traverses an iTree from the root node to the terminal node in which the instance  $x$  is located
  - ✓  $h(x)$  is normalized by the average path length of  $h(x)$  given  $n$ 
    - $c(n) = 2H(n-1)-(2(n-1)/n)$  ( $H(i) = \ln(i) + 0.5772156649$  (Euler's constant))

# Isolation Forest

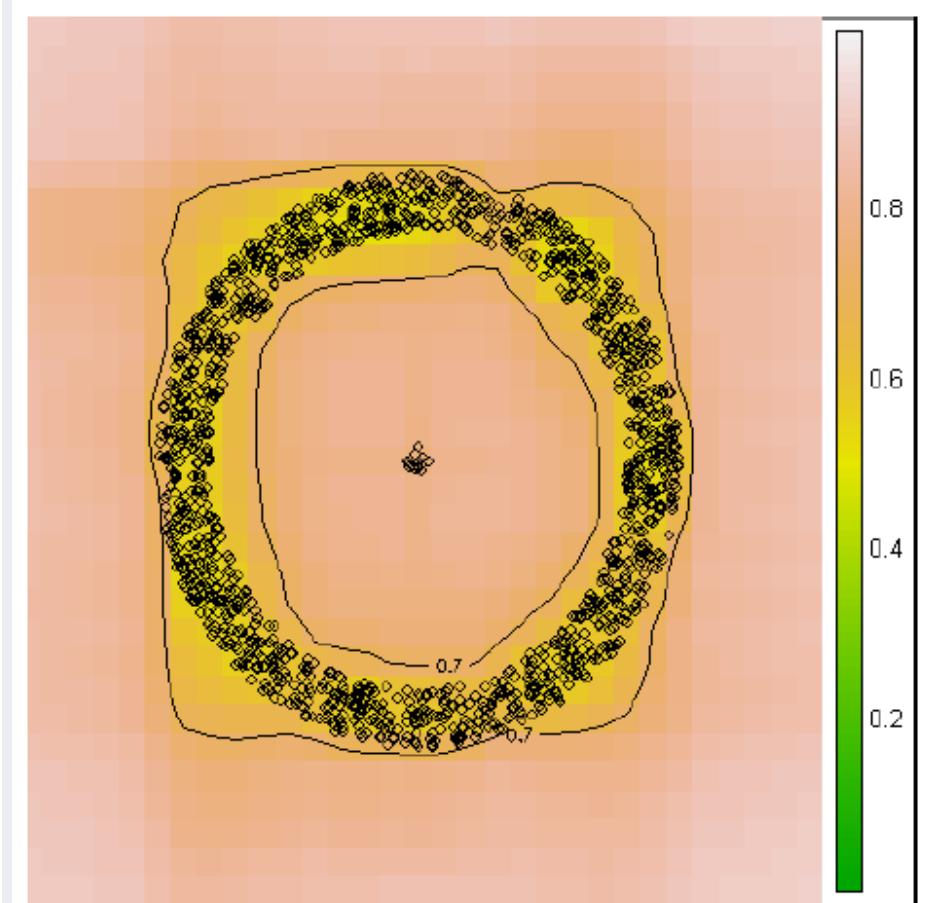
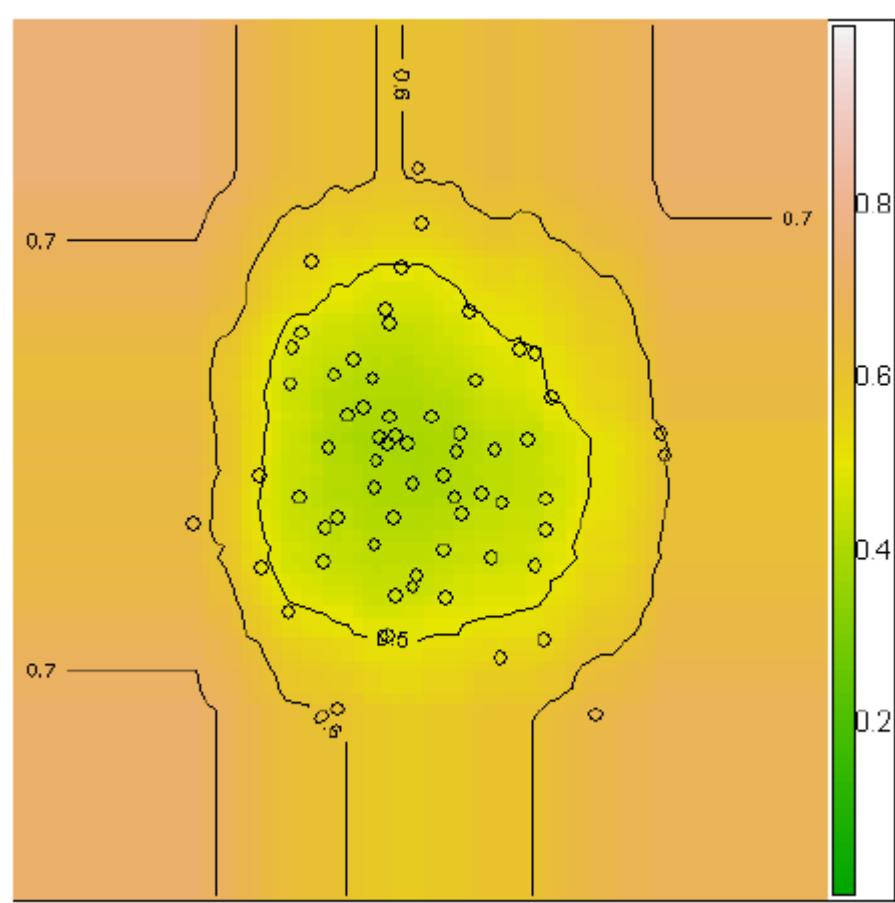
- Definition: Novelty score
  - ✓ The path length  $h(x)$  of an instance  $x$  is measured by the number of edges  $x$  traverses an iTree from the root node to the terminal node in which the instance  $x$  is located
  - ✓  $h(x)$  is normalized by the average path length of  $h(x)$  given  $n$ 
    - $c(n) = 2H(n-1) - (2(n-1)/n)$  ( $H(i) = \ln(i) + 0.5772156649$  (Euler's constant))
  - ✓ The novelty score  $s$  of an instance  $x$  is defined by

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

- When  $E(h(x)) \rightarrow c(n)$ ,  $s \rightarrow 0.5$
- When  $E(h(x)) \rightarrow 0$ ,  $s \rightarrow 1$
- When  $E(h(x)) \rightarrow n - 1$ ,  $s \rightarrow 0$

# Isolation Forest

- Novelty score contour



# Isolation Forest

- Training Isolation Forest

- ✓ Randomly sample datasets
- ✓ Construct iTree
- ✓ Compute the path length

---

**Algorithm 1 :**  $iForest(X, t, \psi)$

---

**Inputs:**  $X$  - input data,  $t$  - number of trees,  $\psi$  - subsampling size

**Output:** a set of  $t$   $iTrees$

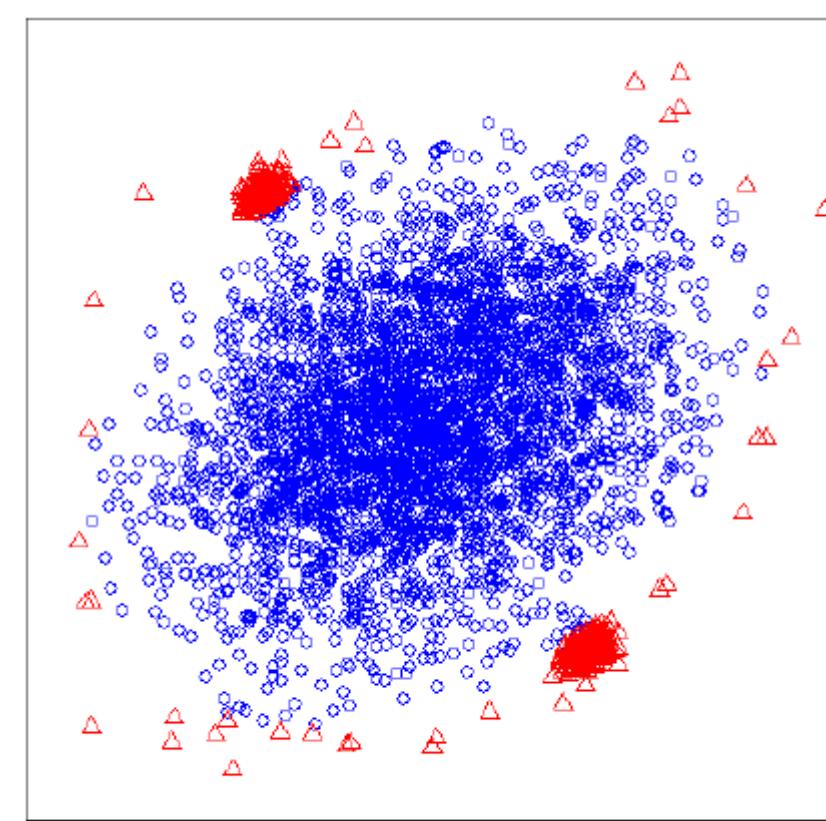
```
1: Initialize Forest
2: for  $i = 1$  to  $t$  do
3:    $X' \leftarrow sample(X, \psi)$ 
4:    $Forest \leftarrow Forest \cup iTree(X')$ 
5: end for
6: return  $Forest$ 
```

---

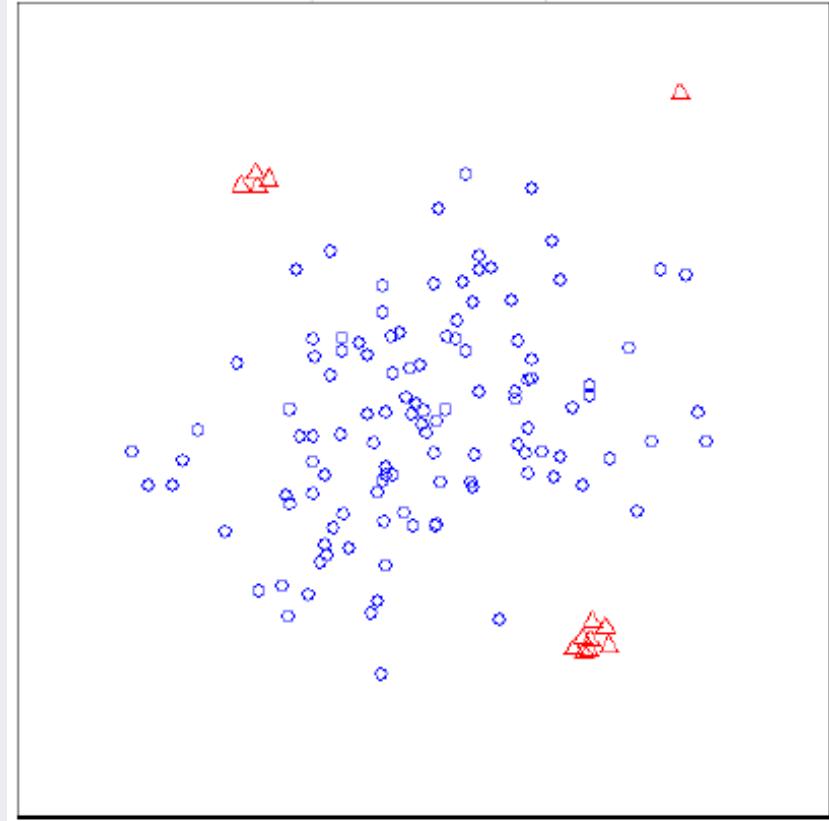
# Isolation Forest

- Training Isolation Forest

- ✓ Randomly sample datasets: 256 is generally enough



(a) Original sample  
(4096 instances)



(b) Sub-sample  
(128 instances)

# Isolation Forest

- Training Isolation Forest

- ✓ Construct iTree

---

**Algorithm 2 :  $iTree(X')$** 

---

**Inputs:**  $X'$  - input data

**Output:** an  $iTree$

```
1: if  $X'$  cannot be divided then
2:   return  $exNode\{Size \leftarrow |X'|\}$ 
3: else
4:   let  $Q$  be a list of attributes in  $X'$ 
5:   randomly select an attribute  $q \in Q$ 
6:   randomly select a split point  $p$  between the  $max$  and  $min$  values of attribute
     $q$  in  $X'$ 
7:    $X_l \leftarrow filter(X', q < p)$ 
8:    $X_r \leftarrow filter(X', q \geq p)$ 
9:   return  $inNode\{Left \leftarrow iTree(X_l),$ 
10:       $Right \leftarrow iTree(X_r),$ 
11:       $SplitAtt \leftarrow q,$ 
12:       $SplitValue \leftarrow p\}$ 
13: end if
```

# Isolation Forest

- Training Isolation Forest

✓ Compute the path length

---

**Algorithm 3** :  $\text{PathLength}(x, T, hlim, e)$ 

---

**Inputs** :  $x$  - an instance,  $T$  - an  $i$ Tree,  $hlim$  - height limit,  $e$  - current path length;  
to be initialized to zero when first called

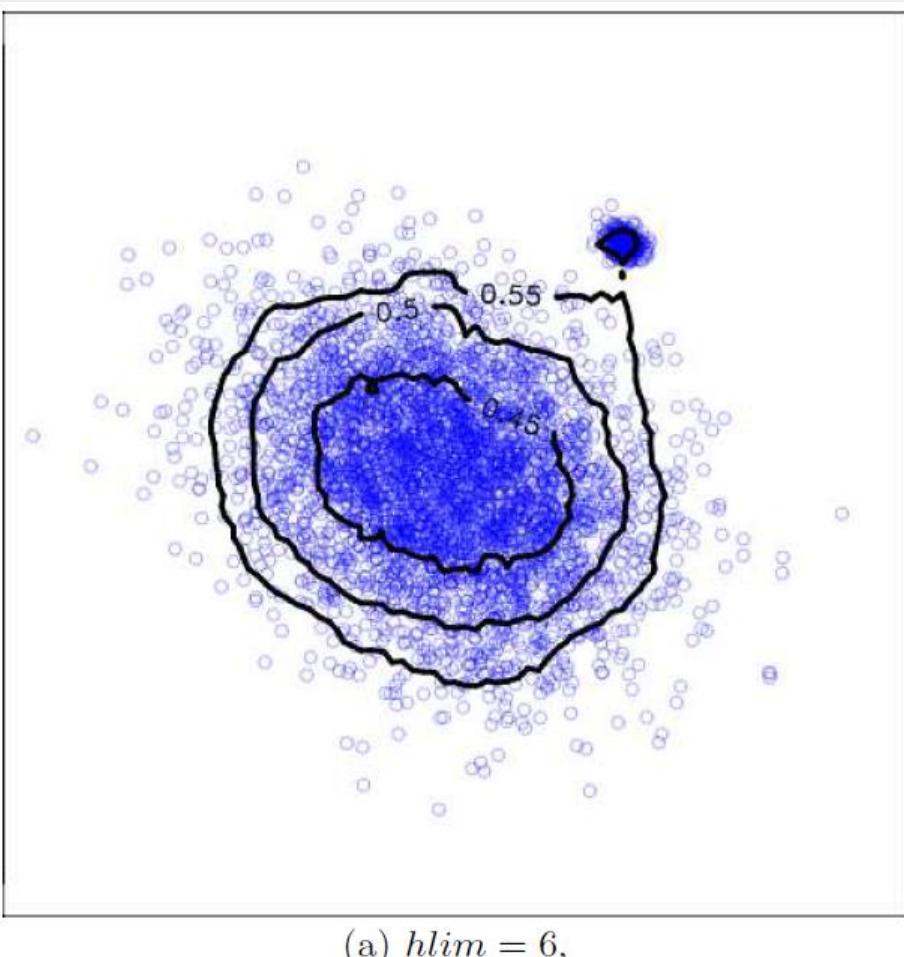
**Output**: path length of  $x$

```
1: if  $T$  is an external node or  $e \geq hlim$  then
2:   return  $e + c(T.\text{size})$   $\{c(\cdot)\}$  is defined in Equation 1}
3: end if
4:  $a \leftarrow T.\text{splitAtt}$ 
5: if  $x_a < T.\text{splitValue}$  then
6:   return  $\text{PathLength}(x, T.\text{left}, hlim, e + 1)$ 
7: else  $\{x_a \geq T.\text{splitValue}\}$ 
8:   return  $\text{PathLength}(x, T.\text{right}, hlim, e + 1)$ 
9: end if
```

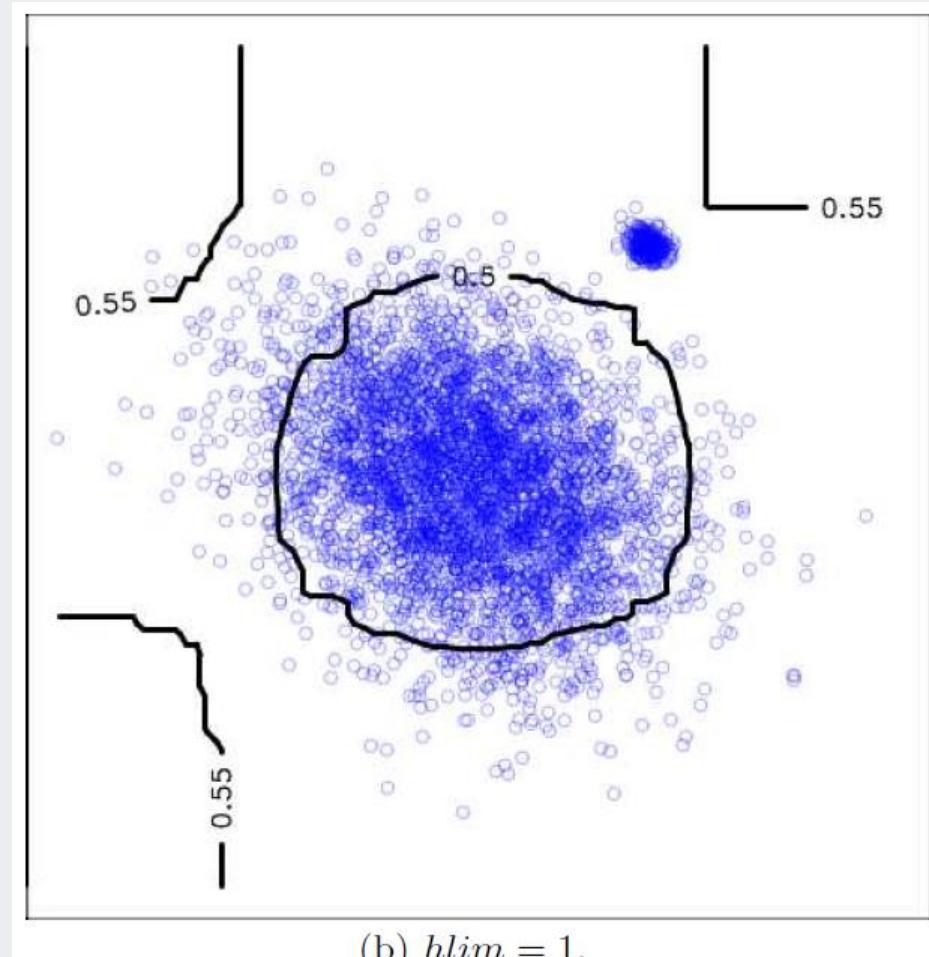
---

# Isolation Forest

- Effect of the height limit



(a)  $hlim = 6,$



(b)  $hlim = 1,$

# Isolation Forest

- Empirical evaluation

- ✓ Datasets

	<i>n</i>	<i>d</i>	anomaly class
Http (KDDCUP99)	567497	3	attack (0.4%)
			class 4 (0.9%)
ForestCover	286048	10	vs. class 2
Mulcross	262144	4	2 clusters (10%)
Smtp (KDDCUP99)	95156	3	attack (0.03%)
Shuttle	49097	9	classes 2,3,5,6,7 (7%)
Mammography	11183	6	class 1 (2%)
Annthyroid	6832	6	classes 1, 2 (7%)
Satellite	6435	36	3 smallest classes (32%)
Pima	768	8	pos (35%)
Breastw	683	9	malignant (35%)
Arrhythmia	452	274	classes 03,04,05,07, 08,09,14,15 (15%)
Ionosphere	351	32	bad (36%)
hbk	75	4	14 points (19%)
wood	20	6	6 instances (30%)

# Isolation Forest

- Empirical evaluation

- ✓ Performance (in terms of AUROC)

	AUC				
	<i>i</i> Forest	ORCA	SVM	LOF	RF
Http (KDDCUP99)	<b>1.00</b>	0.36	0.90	*	**
ForestCover	0.87	0.83	<b>0.90</b>	0.57	**
Mulcross	<b>0.96</b>	0.33	0.59	0.59	**
Smtp (KDDCUP99)	<b>0.89</b>	0.80	0.78	0.32	**
Shuttle	<b>1.00</b>	0.60	0.79	0.55	**
Mammography	<b>0.84</b>	0.77	0.65	0.67	**
Annthyroid	<b>0.84</b>	0.68	0.63	0.72	**
Satellite	<b>0.73</b>	0.65	0.61	0.52	**
Pima	0.67	<b>0.71</b>	0.55	0.49	0.65
Breastw	<b>0.98</b>	<b>0.98</b>	0.66	0.37	0.97
Arrhythmia	<b>0.81</b>	0.78	0.71	0.73	0.60
Ionosphere	0.83	<b>0.92</b>	0.71	0.89	0.85

(a) AUC performance

# Isolation Forest

- Empirical evaluation

- ✓ Performance (in terms of computational complexity)

	Time (seconds)							
	iForest			ORCA	SVM	LOF	RF	
	Train	Eval.	Total					
Http	0.25	15.33	<b>15.58</b>	9487.47	35872.09	*	**	
ForestCover	0.76	15.57	<b>16.33</b>	6995.17	9737.81	224380.19	**	
Mulcross	0.26	12.26	<b>12.52</b>	2512.20	7342.54	156044.13	**	
Smtp	0.14	2.58	<b>2.72</b>	267.45	986.84	24280.65	**	
Shuttle	0.30	2.83	<b>3.13</b>	156.66	332.09	7489.74	**	
Mammography	0.16	0.50	<b>0.66</b>	4.49	10.8	14647.00	**	
Annthyroid	0.15	0.36	<b>0.51</b>	2.32	4.18	72.02	**	
Satellite	0.46	1.17	<b>1.63</b>	8.51	8.97	217.39	**	
Pima	0.17	0.11	0.28	<b>0.06</b>	<b>0.06</b>	1.14	4.98	
Breastw	0.17	0.11	0.28	<b>0.04</b>	0.07	1.77	3.10	
Arrhythmia	2.12	0.86	2.98	<b>0.49</b>	0.15	6.35	2.32	
Ionosphere	0.33	0.15	0.48	<b>0.04</b>	0.04	0.64	0.83	

(b) Actual processing time



# References

## Research Papers

- Breunig, M.M., Kriegel, H.-P., Ng, R.T., and Sander, J. (2000). LOF: Identifying density-based local outliers. Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data: 93-104.
- Chandola,V., Banerjee,A., and Kumar,V. (2009).Anomaly detection:A survey.ACM computing surveys 41(3): 15.
- Harmeling, S. Dornhege, G., Tax, D. Meinecke, F., and Muller, K-R. (2006). From outliers to prototype: Ordering data. Neurocomputing 69(13-15): 1608-1618.
- Kang, P. and Cho, S. (2009).A hybrid novelty score and its use in keystroke dynamics-based user authentication. Pattern Recognition 42(11):3115-3127.
- Liu, F.T, Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on (pp. 413-422). IEEE.
- Liu, F.T, Ting, K. M., & Zhou, Z. H. (2012). Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data (TKDD), 6(1), 3.
- Pimentel, M.A.F, Clifton, D.A., Clifton, L., and Tarassenko, L. (2014).A review of novelty detection, Signal Processing 99: 215-249.
- Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., and Williamson, R. C.(2001). Estimating the support of a high-dimensional distribution. Neural computation 13(7): 1443-1471.
- Tax, D.M. (2001). One-class classification, Ph.D.Thesis, Delft University of Technology, Netherlands.
- Tax, D.M. and Duin, R.P. (2004). Support vector data description. Machine learning 54(1): 45-66.

# References

## Other materials

- Pages 28-33 & 36: [http://research.cs.tamu.edu/prism/lectures/pr/pr\\_I7.pdf](http://research.cs.tamu.edu/prism/lectures/pr/pr_I7.pdf)
- Figures in Auto-encoder section: [https://dl.dropboxusercontent.com/u/19557502/6\\_01\\_definition.pdf](https://dl.dropboxusercontent.com/u/19557502/6_01_definition.pdf)
- Gramfort, A. (2016). Anomaly/Novelty detection with scikit-learn: <https://www.slideshare.net/agramfort/anomaly-novelty-detection-with-scikitlearn>