Hes·so
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences Western Switzerland

MSE | MASTER OF SCIENCE IN ENGINEERING

Multimodal Processing, Recognition and Interaction

Anomaly Detection & Support Vector Machines

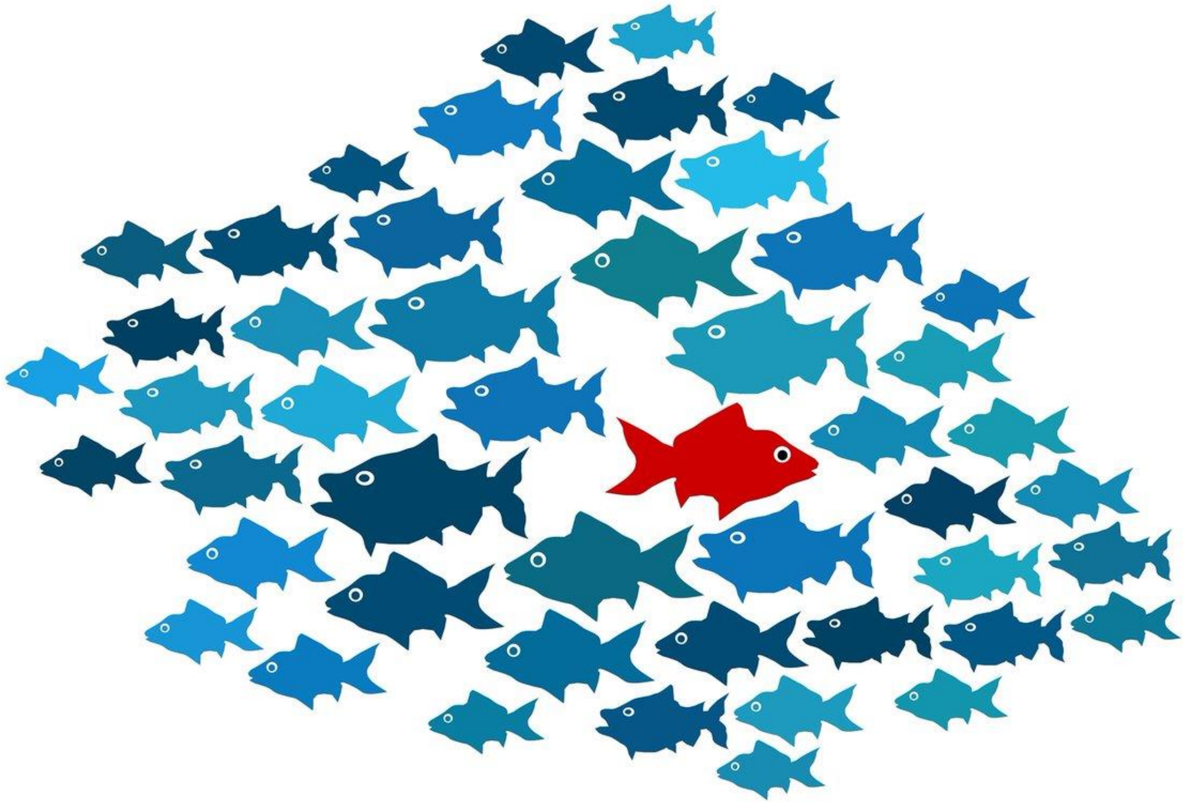
Stefano Carrino

Elena Mugellini, Stefano Carrino, Omar Abou Khaled

Summary

- Introduction
- Anomalies
- Problem Outline & Challenges
- Methodologies
  - Kinds of Anomalies (by data structure)
  - Related algorithms

- Focus on: Support Vector Machines (SVM)
- SVM & Anomalies
- Conclusion
- What you should know



Definitions, challenges and methodologies

ANOMALY DETECTION

Introduction

- Big Data!

Anomalies – What are anomalies?

- Definition: "***Anomalies*** *are patterns in data that do not conform to a* ***well defined*** *notion of* ***normal*** *behavior*"

- *Anomalies might be induced in the data for a* ***variety of reasons****, such as malicious activity, for example, credit card fraud, cyber-intrusion, terrorist activity or breakdown of a system, but all of the reasons have the* ***common characteristic*** *that* ***they are interesting to the analyst****.*

(Chandola et al., 2009)

What is anomaly detection?

- ***Anomaly detection*** *refers to the problem of finding* ***patterns*** *in data that* ***do not conform to expected behavior***

Other names

- Anomalies are frequently called **Outliers**
  - In a scatter plot of the data, they lie far away form other data points
- Anomaly detection is also called
  - **Deviation detection**
    - Anomalous objects attributes *deviate* from the expected behavior of typical values
  - **Exception mining**
    - Anomalies are exceptional

Domains of application

- Intrusion/Attack detection
- Fraud detection
- Industrial Damage Detection
    - Predictive maintenance
- Medical and Public Health Anomaly Detection
- Image processing
- Anomaly detection in text data
- Sensor networks
- …
- Training set cleaning!!

Challenges

- Conceptual challenges
- Technical challenges

Conceptual Challenges

- Defining a normal region that encompasses every possible normal behavior is very difficult
- *Normal* behavior can evolve with time
- Typically (pun intended), in anomaly detection problems, the data representing the "normal" behavior of an entity (user, system, device, ...) is known, but there is no or only few samples of anomalous data.
- The algorithm should be able to detect anomalies **never encountered before**
- If anomalies are the result of malicious actions…

Technical Challenges

- Number of Attributes Used to Define an Anomaly
  - 1 attribute is enough?
  - E.g. man 155 cm && 105 kg
- Global versus Local Perspective
  - E.g., height 200 cm in normal population Vs 200 cm basket team
- Degree to Which a Point Is an Anomaly
  - Beyond a YES/NO approach
- Evaluation
  - Unbalanced dataset => accuracy is useless
  - No label at all => precision, recall are useless too

Useless data

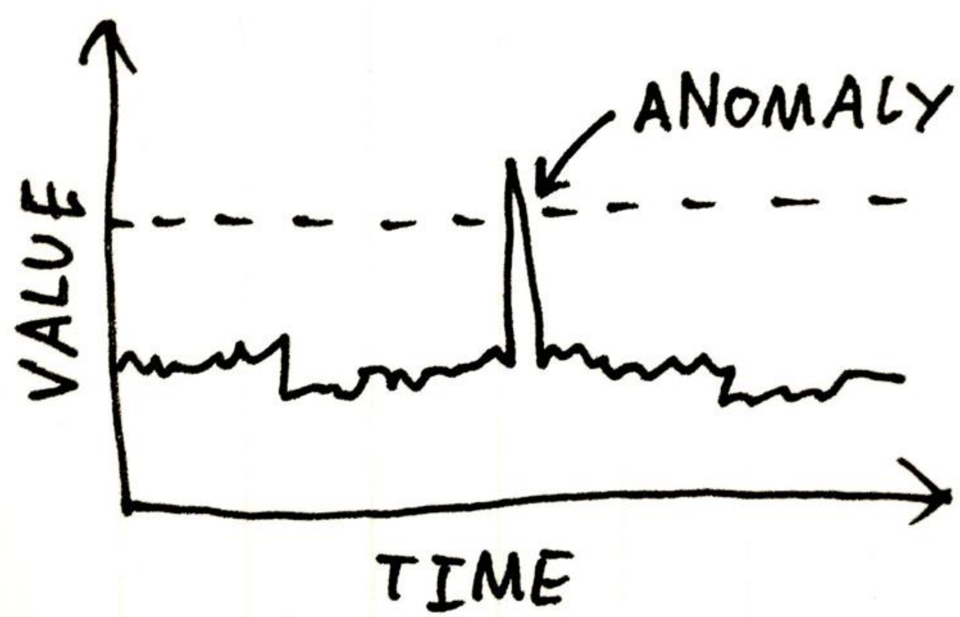
&

Critical information

Anomalies can indicate both!

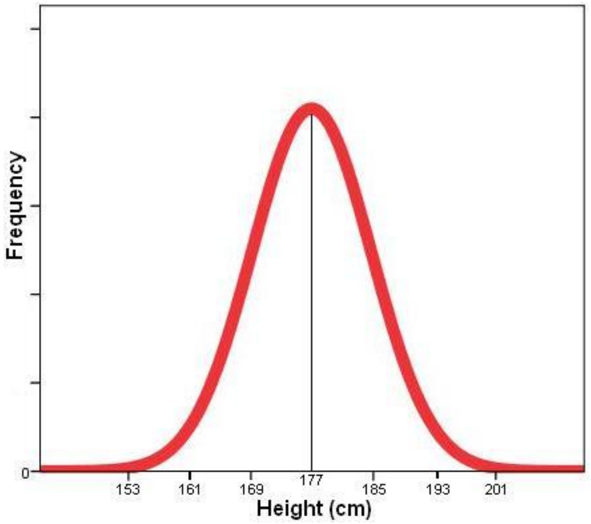
Characteristics of an anomaly detection problem

- The most important characteristics of an anomaly detection problem are:
  - the **causes** of the anomaly
  - the **nature** of the **input data**
  - the **type** of the anomaly
  - the availability of **labeled data**
  - the **output constraints**



Causes of an anomaly (I)

- Data from different classes
  - An object can be different from another because it is of a different type or class
  - E.g. fraud detection
- Natural variation
  - Dataset modeled by statistical distribution
  - E.g. height
- Data measurement and collection error
  - E.g. sensor data error



Causes of an anomaly (II)

- Data from different classes
  - A different class very poorly represented
  - Typically very interesting!
- Natural variation
  - Often interesting
- Data measurement and collection error
  - Reduce the quality of data => TO REMOVE or reduce their impact !

Exercise: Anomalies & Feature Rescaling

- We may have encountered few approaches of feature scaling:
  - Min-max rescaling & feature standardization

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

$$x' = \frac{x - mean(x)}{std(x)}$$

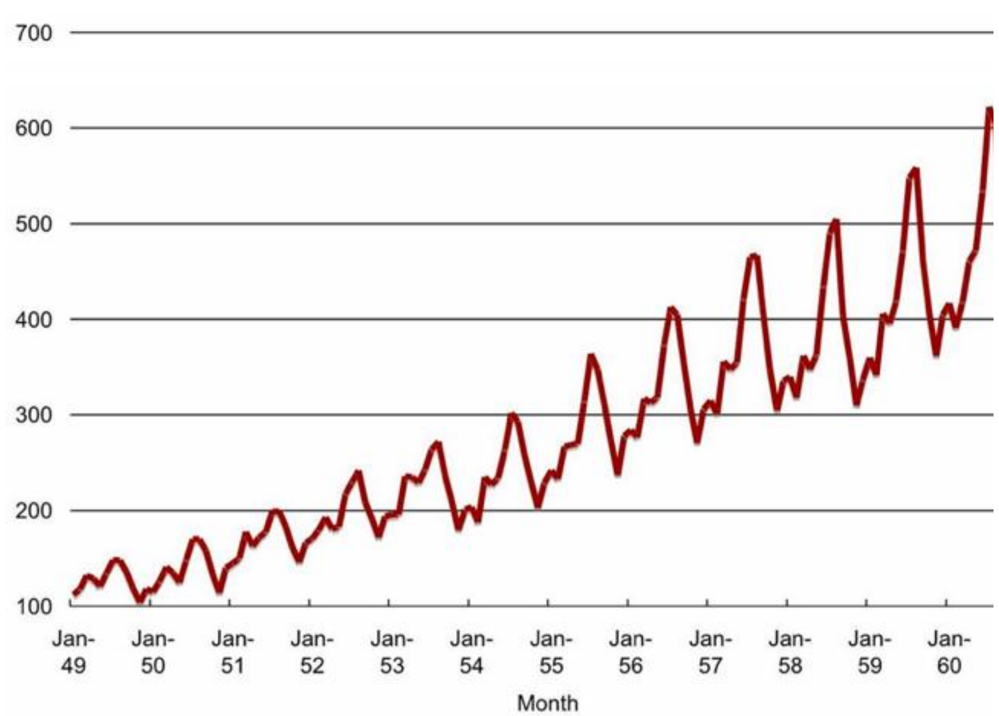
  - Given the some data (X), evaluate the impact of outliers in the normalization approach
  - Download the jupyter notbook from Moodle

Quick Practice

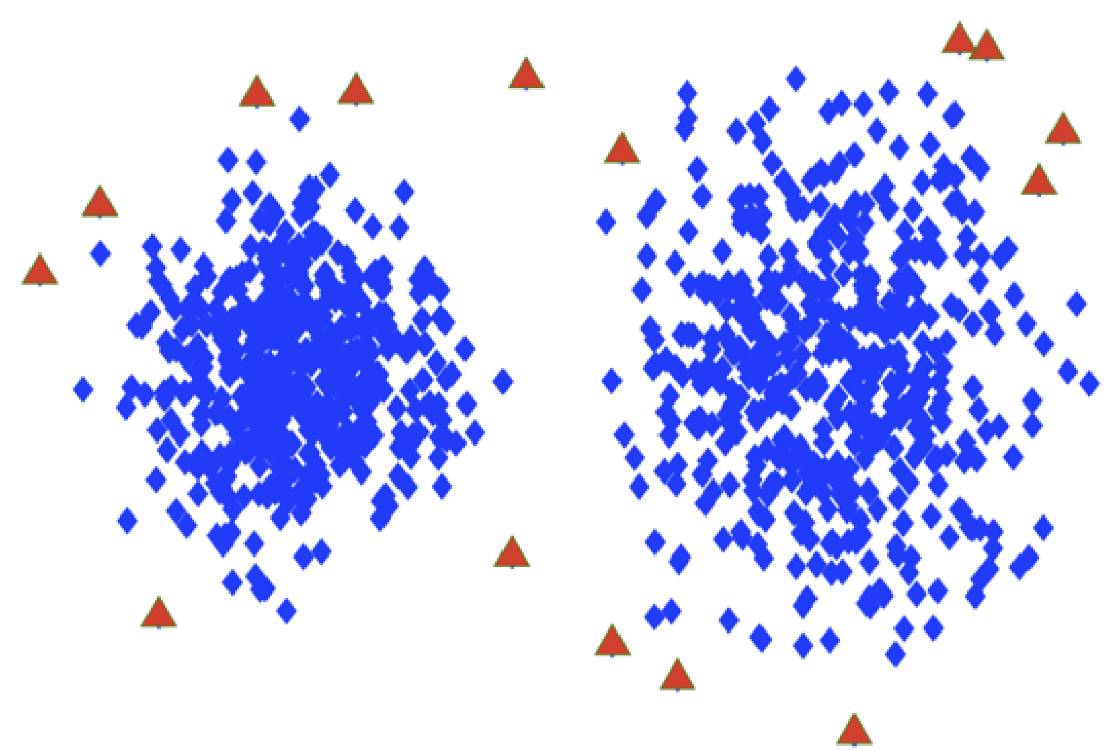
- The impact of outliers on feature scaling
- Retrieve the Jupyter Notebook on Moodle
- Complete the # TODO sections
- Observe the results

Input data

- The nature of the input data has a large influence on the anomaly detection process.
  - In some cases the data instance must be considered individually and in others there is a direct relationship, like in sequence data (e.g., time series)



http://oracledmt.blogspot.ch/



http://networks.ece.mcgill.ca/node/186

Nature of an Anomaly

- The type of anomalies that must be discovered by the system is also an important criteria to select the appropriate algorithms
- Anomalies can be outlined via:
  - the input data structure;
  - the effects of the anomaly on the system (point, recurrent, permanent);
  - …

Anomaly classification by data structure (I)

- Three main types of anomalies
  - Point anomalies
  - Contextual anomalies
  - Collective anomalies

Anomaly classification by data structure (II)

- Point anomalies
  - A single data instance that is anomalous considering the rest of the dataset.
  - It is the simplest type of anomaly to detect

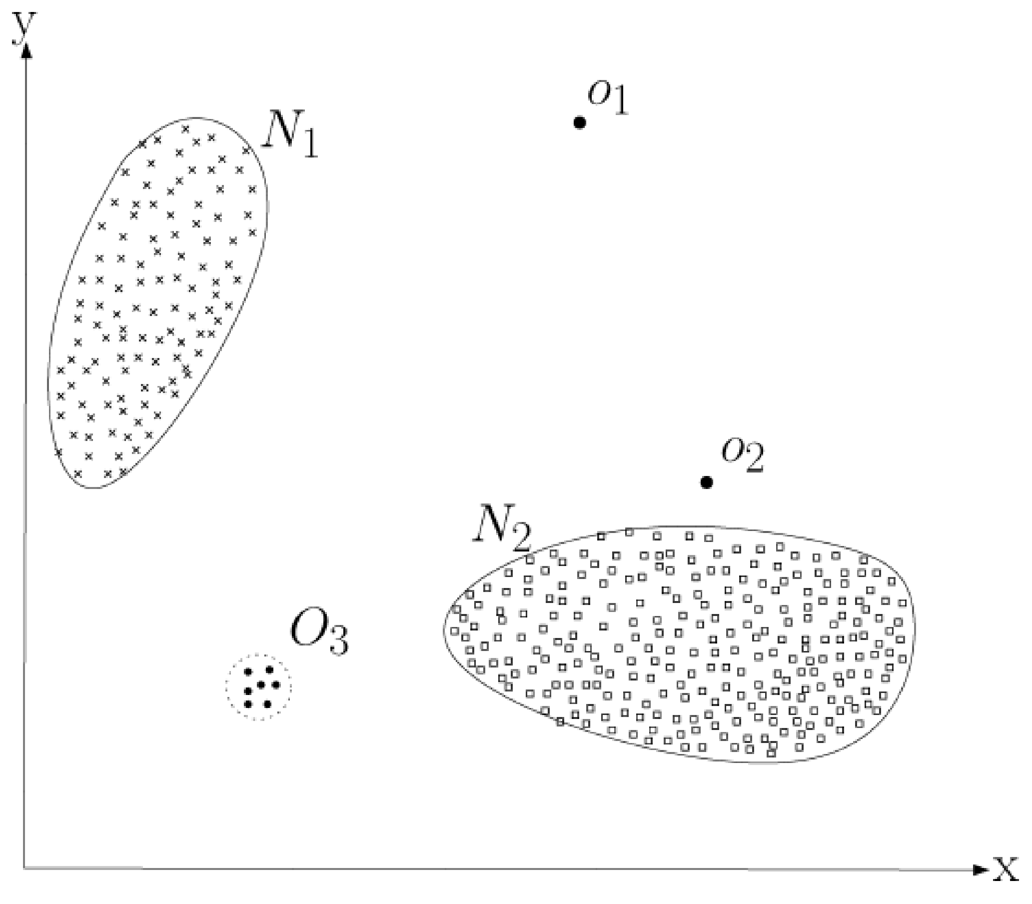


Figure – Point anomalies

In the figure, O_1 and O_2 are point anomalies.

But what about O_3?

Anomaly classification by data structure (III)

- Contextual anomalies
  - If a data instance is anomalous in a specific context, but not otherwise, then it is termed a contextual anomaly (or *conditional anomaly*)

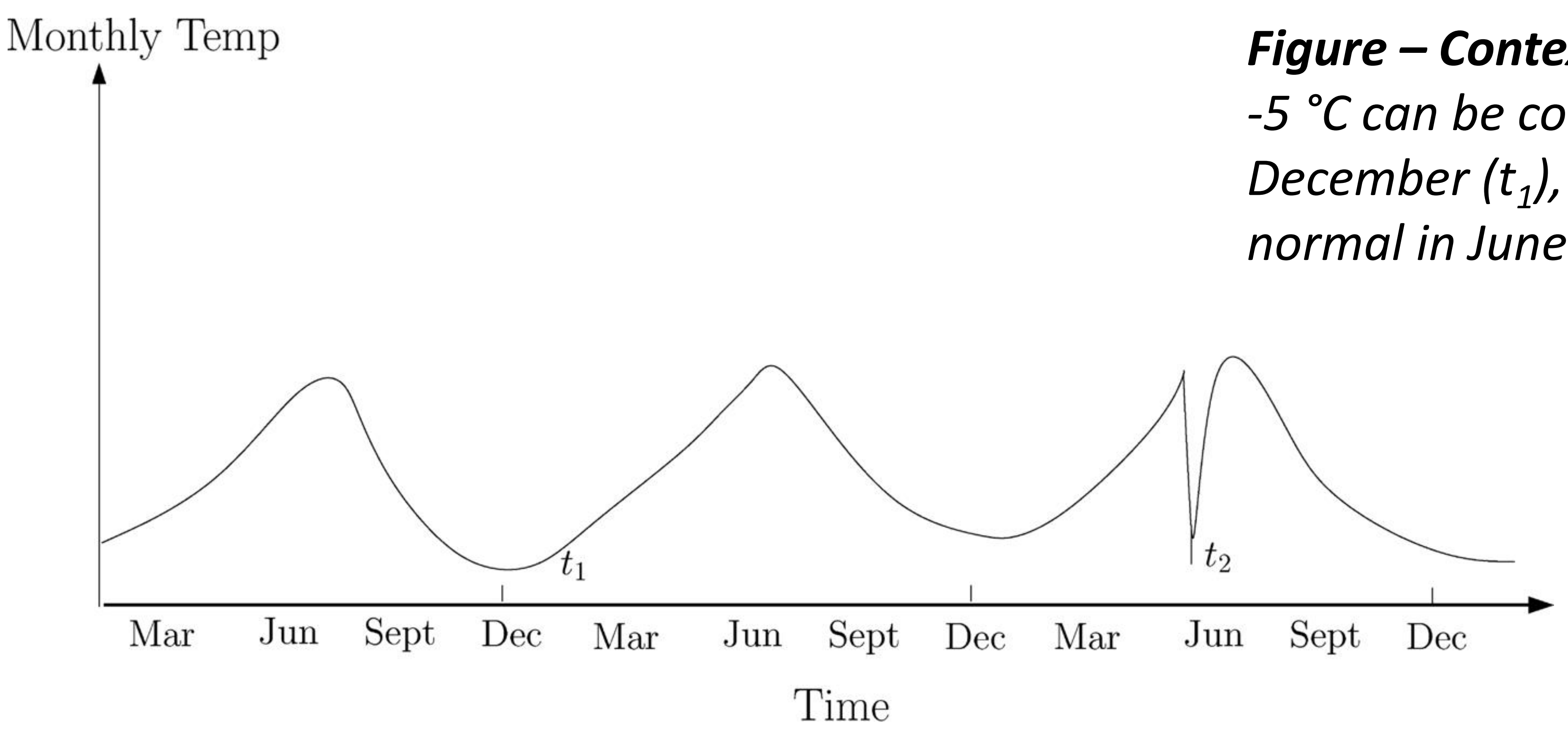


***Figure – Contextual Anomaly**:*
-5 °C can be considered normal in December (t_1), the same value is not normal in June (t_2)

Anomaly classification by data structure (IV)

- Contextual anomalies
  - The anomalous behavior is determined using the values for the **contextual attributes** within a specific context.
  - A data instance might be a contextual anomaly in a given context, but an identical data instance (in terms of **behavioral attributes**) could be considered normal in a different context.

Anomaly classification by data structure (V)

- Collective anomalies
  - If a collection of related data instances is anomalous with respect to the entire data set, it is termed a **collective anomaly**. The individual data instances in a collective anomaly may not be anomalies by themselves, but their occurrence together as a collection is anomalous.

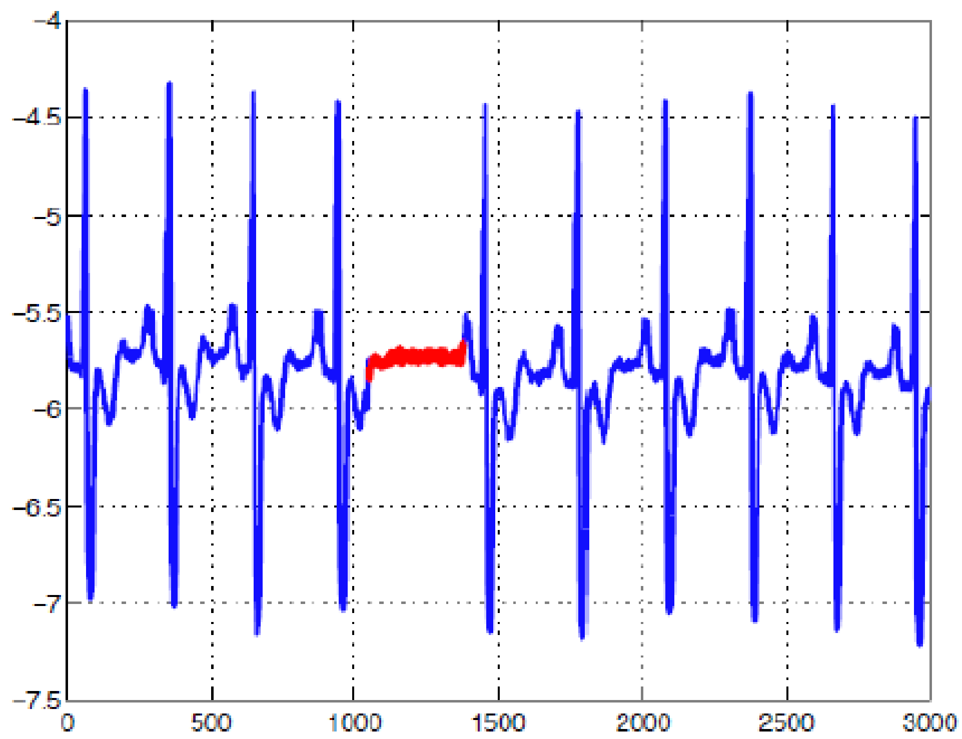
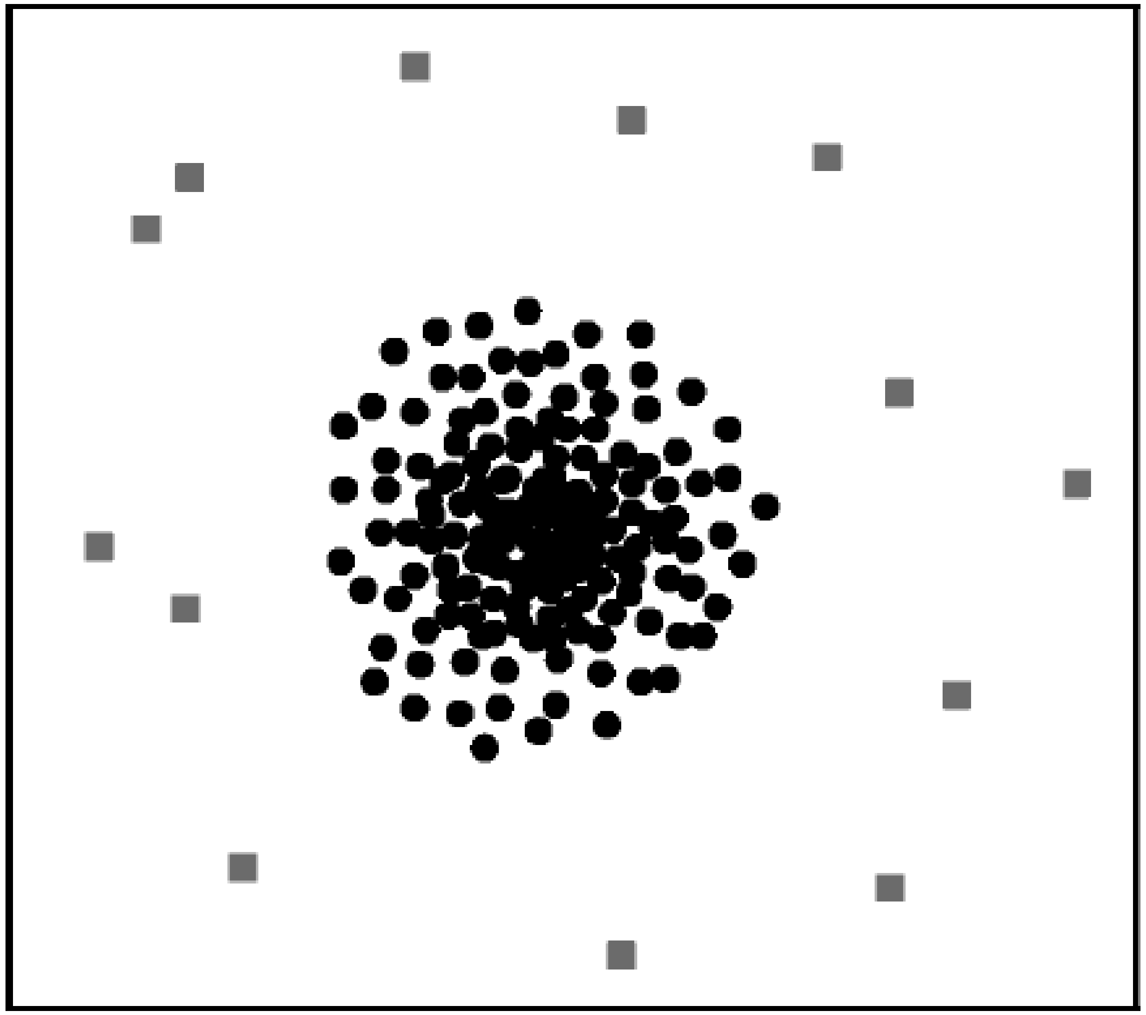


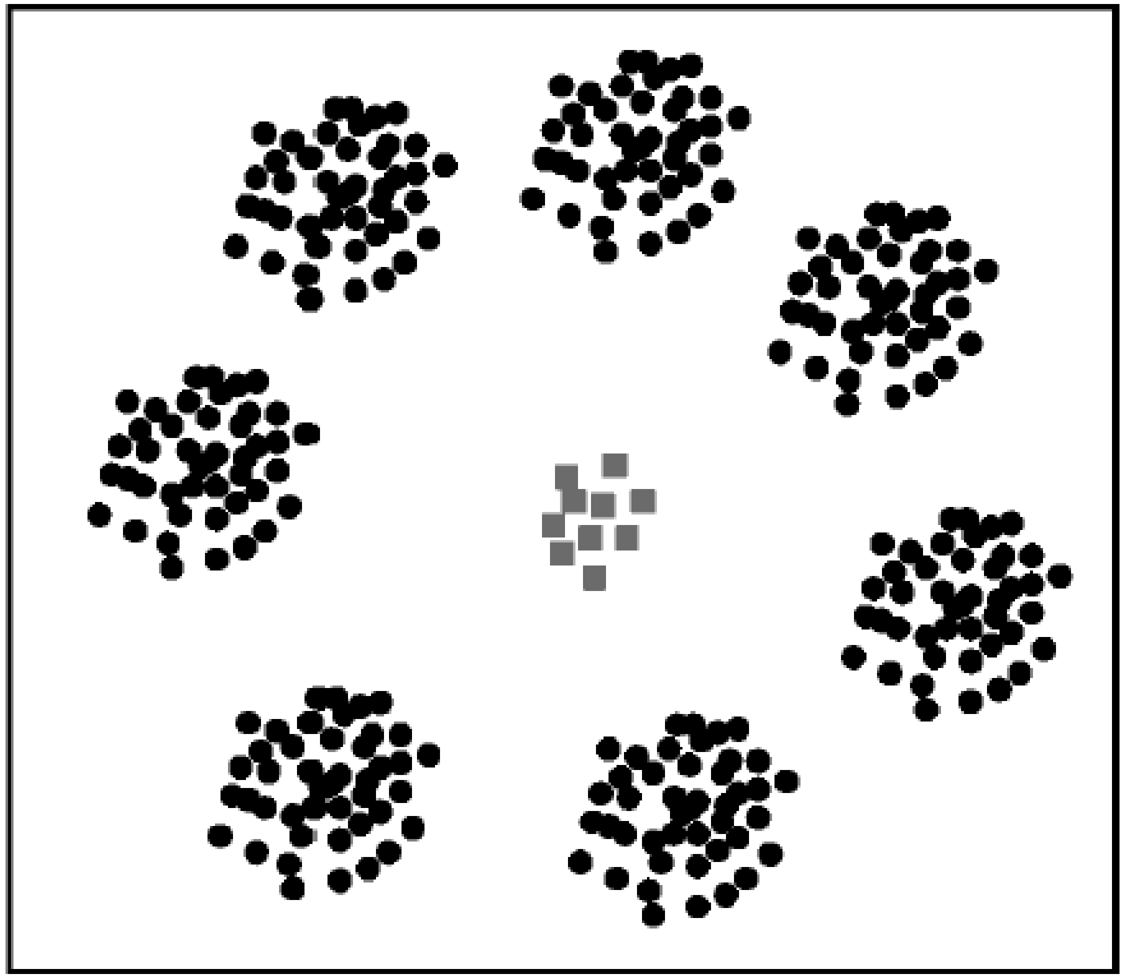
*Figure – **Anomaly in human electrocardiogram output***
*The highlighted region denotes an anomaly because the same low value exists for an **abnormally** long time (corresponding to an* Atrial Premature Contraction*). Note that that low value by itself is not an anomaly.*

Quiz – Point, contextual or collective? (I)



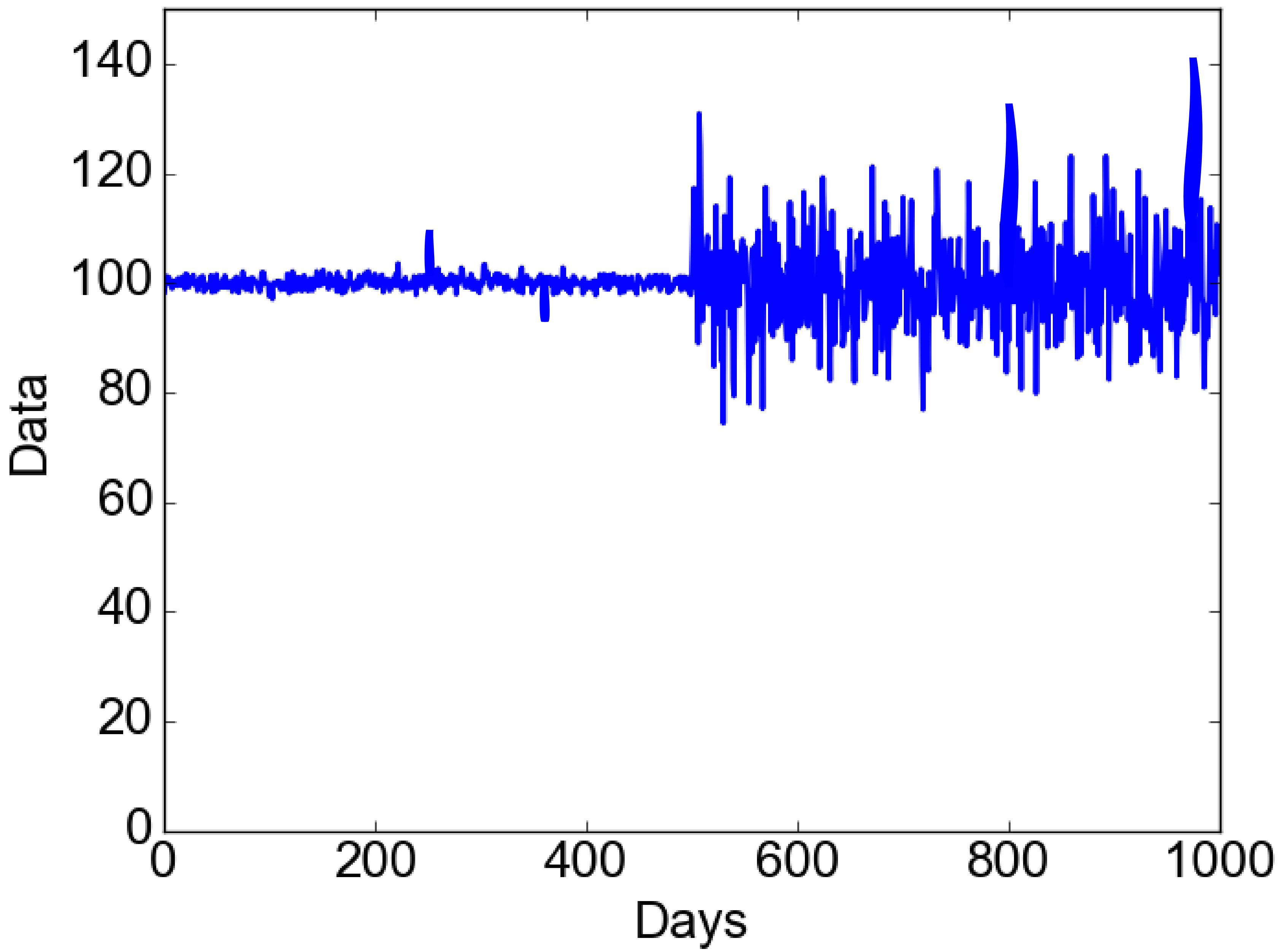
Normal instances are shown as circles and anomalies are shown as square

Quiz – Point, contextual or collective? (II)

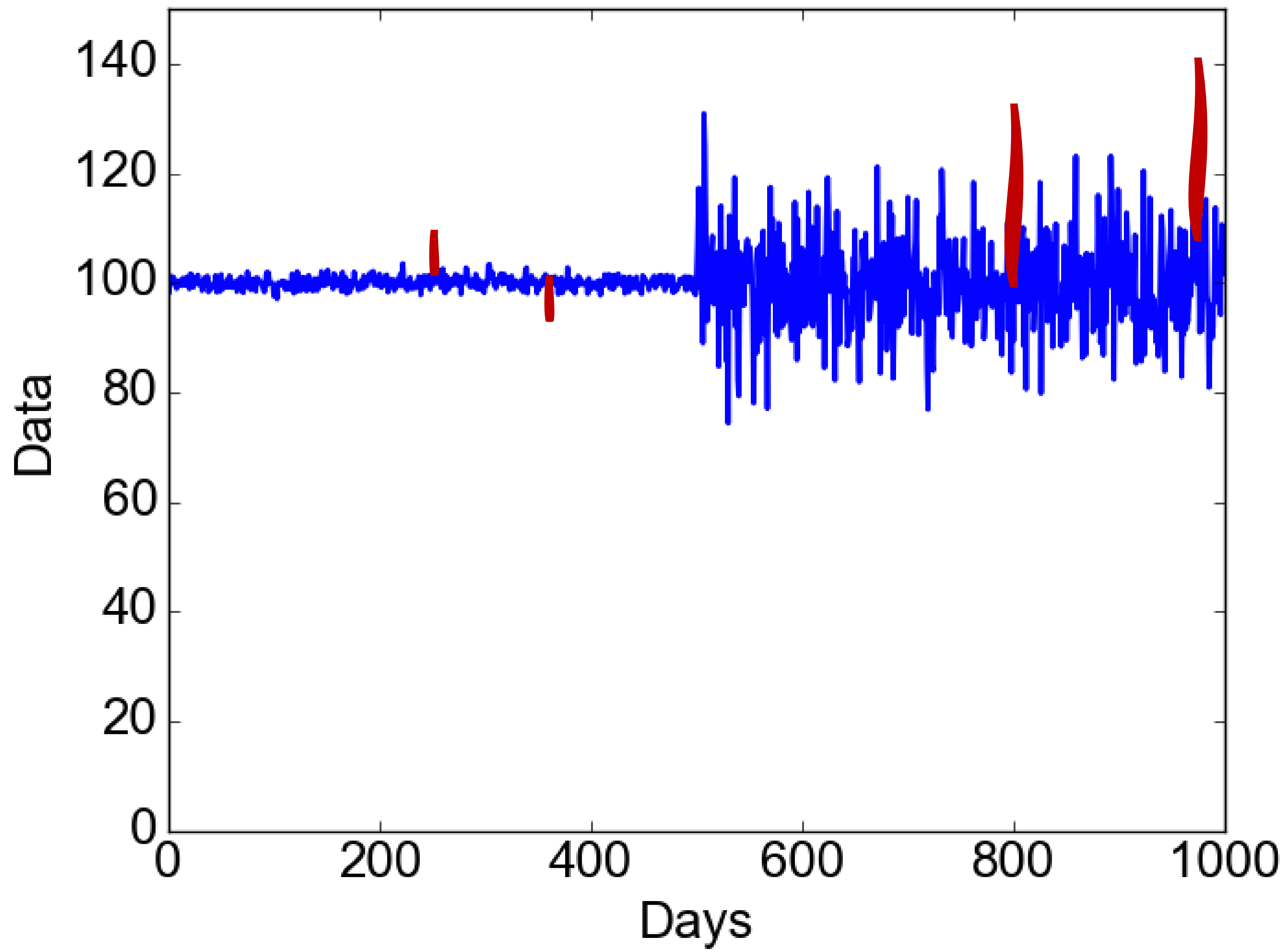


Normal instances are shown as circles and anomalies are shown as square

Quiz – Point, contextual or collective? (III)

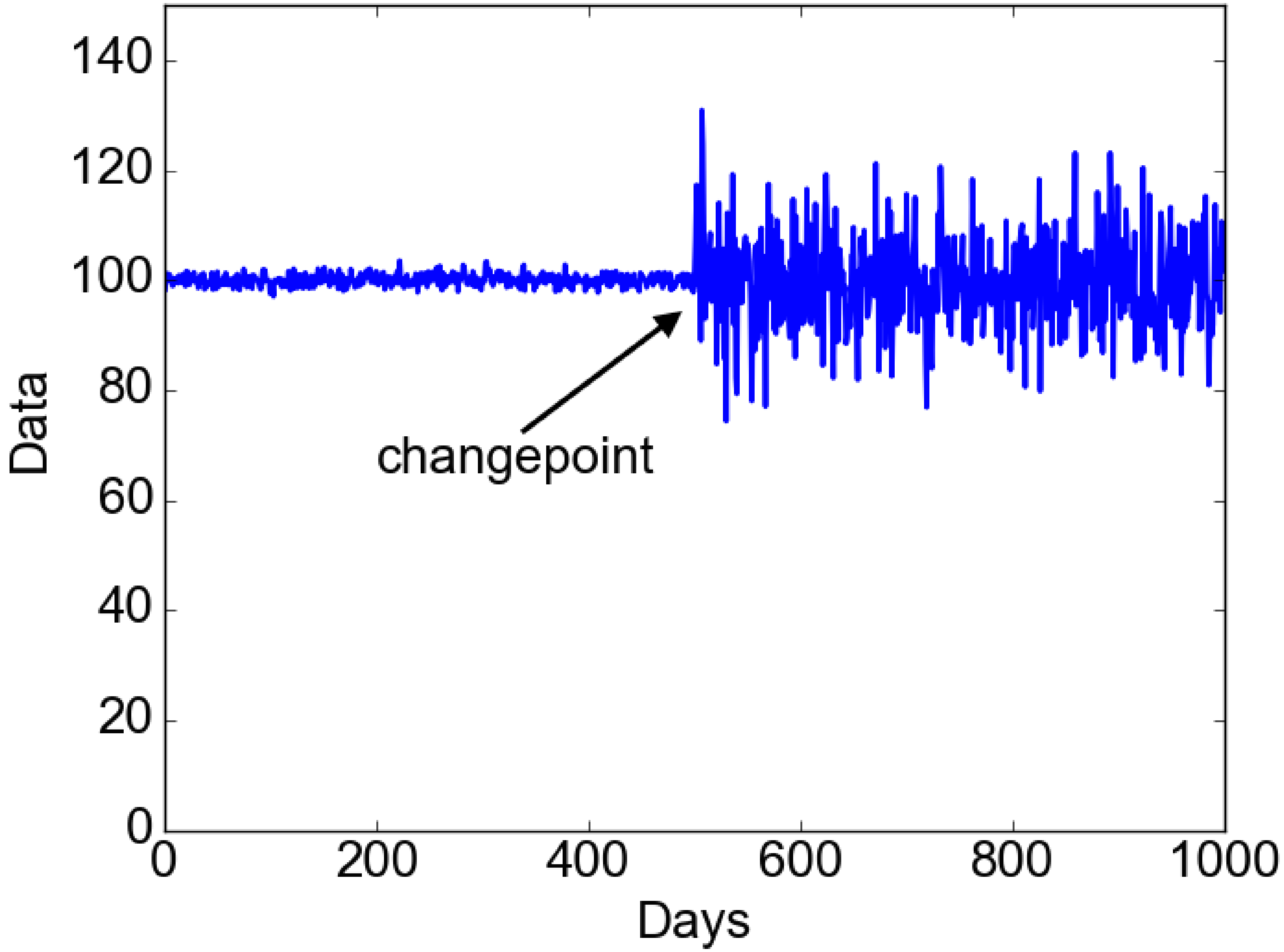


Quiz – Point, contextual or collective? (III)



Quiz – Point, contextual or collective? (III)

Problem: regime shift detection



Labeled Data Availability

- Obtaining labeled data is often prohibitively expensive
- New types of anomalies might arise, for which there is no labeled training data.
- Impact of data availability on the methodology:

Normal Data Labels	Anomalous Data Labels	Methodology
Available	Available	**Supervised** ML
Available	Unavailable	**Semi-supervised** ML
Unavailable	Unavailable	**Unsupervised** ML

Output constraints

- The outputs produced by anomaly detection techniques are one of the following two types:
  - Scores:
    - Scoring techniques assign an anomaly score to each instance in the test data depending on the degree to which that instance is considered an anomaly
  - Labels:
    - Techniques in this category assign a label (*normal* or *anomalous*) to each test instance.

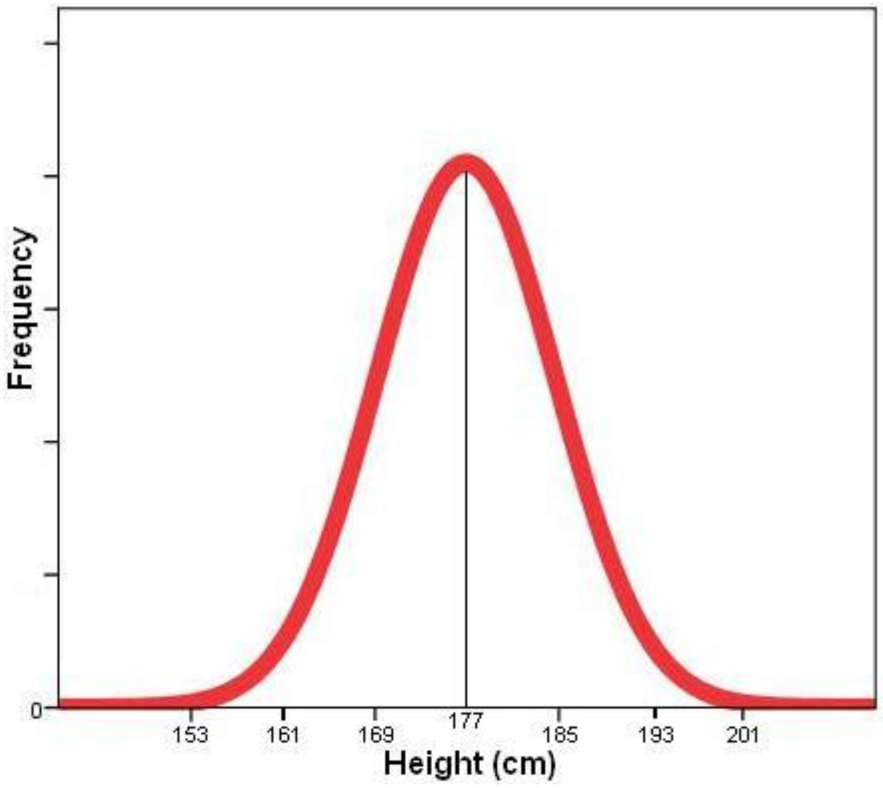
Anomaly Detection Methodologies (I)

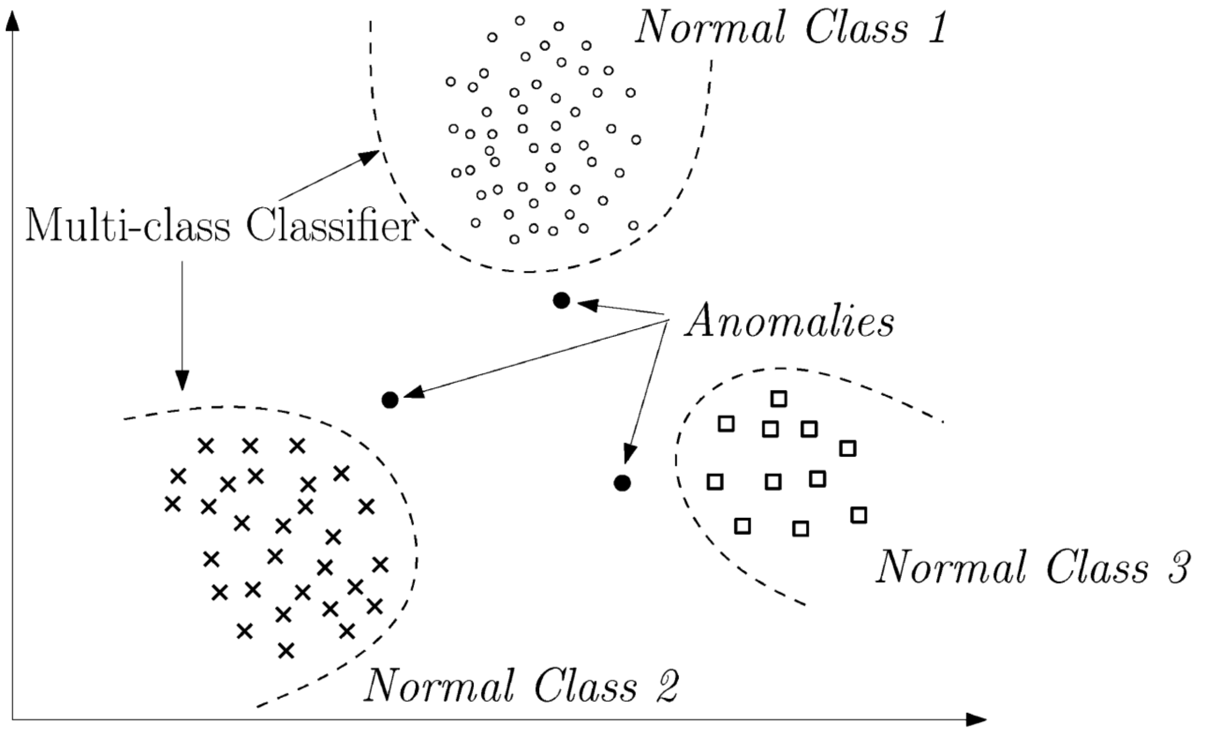
- Generally three methodologies can be identified:
  - **Model-based** techniques
  - **Proximity-based** techniques
  - **Density-based** techniques

- Note: *machine learning solutions* can be classed in one of the three previous methodologies according to the underlying approach

Anomaly Detection Methodologies (II)

- Model-based techniques
  - Many anomaly detection techniques first build a model of the data.
  - Anomalies are objects that do not fit the model very well.
  - E.g. statistical approaches, GMM





Anomaly Detection Methodologies (III)

- In same cases, it is difficult to build a model
  - The distribution of the data is unknown
  - No training data available
- But:
  - It is possible to define a proximity measure between objects

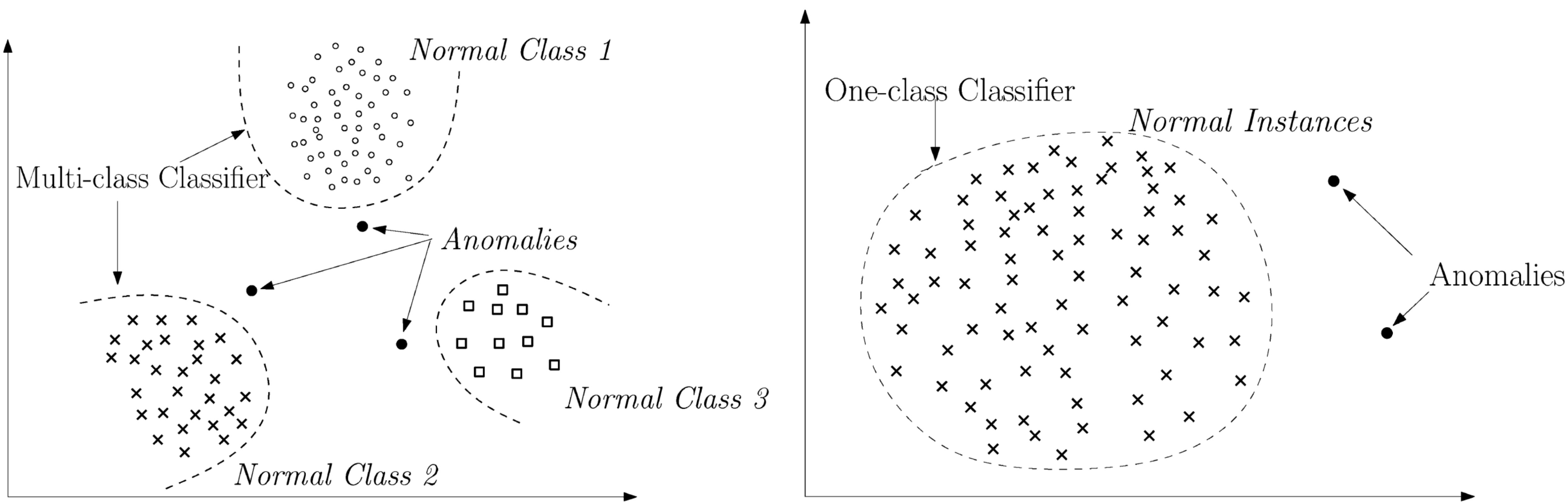
- **Proximity-based** techniques
  - Anomalous objects are those that are distant from most of the other objects
  - E.g. distance to the k-nearest neighbor, SVM

Anomaly Detection Methodologies (IV)

- Difficult to build a model
- But:
  - It is possible to define a proximity measure between objects & estimate of the density of objects

- **Density-based** techniques
  - Objects that are in regions of low density are relatively distant from their neighbors, and can be considered anomalous.
  - Ex. Clustering techniques

Methodologies – Point Anomalies detection (I)



(a) Multi-class Anomaly Detection

(b) One-class Anomaly Detection

Methodologies – Point Anomalies detection (II)

- **Model-based** techniques
  - Statistical techniques
  - Parametric techniques: based on Gaussian models, regression models, or mixtures of parametric distributions
  - Non-Parametric techniques: based on histograms or kernel functions

Methodologies – Point Anomalies detection (III)

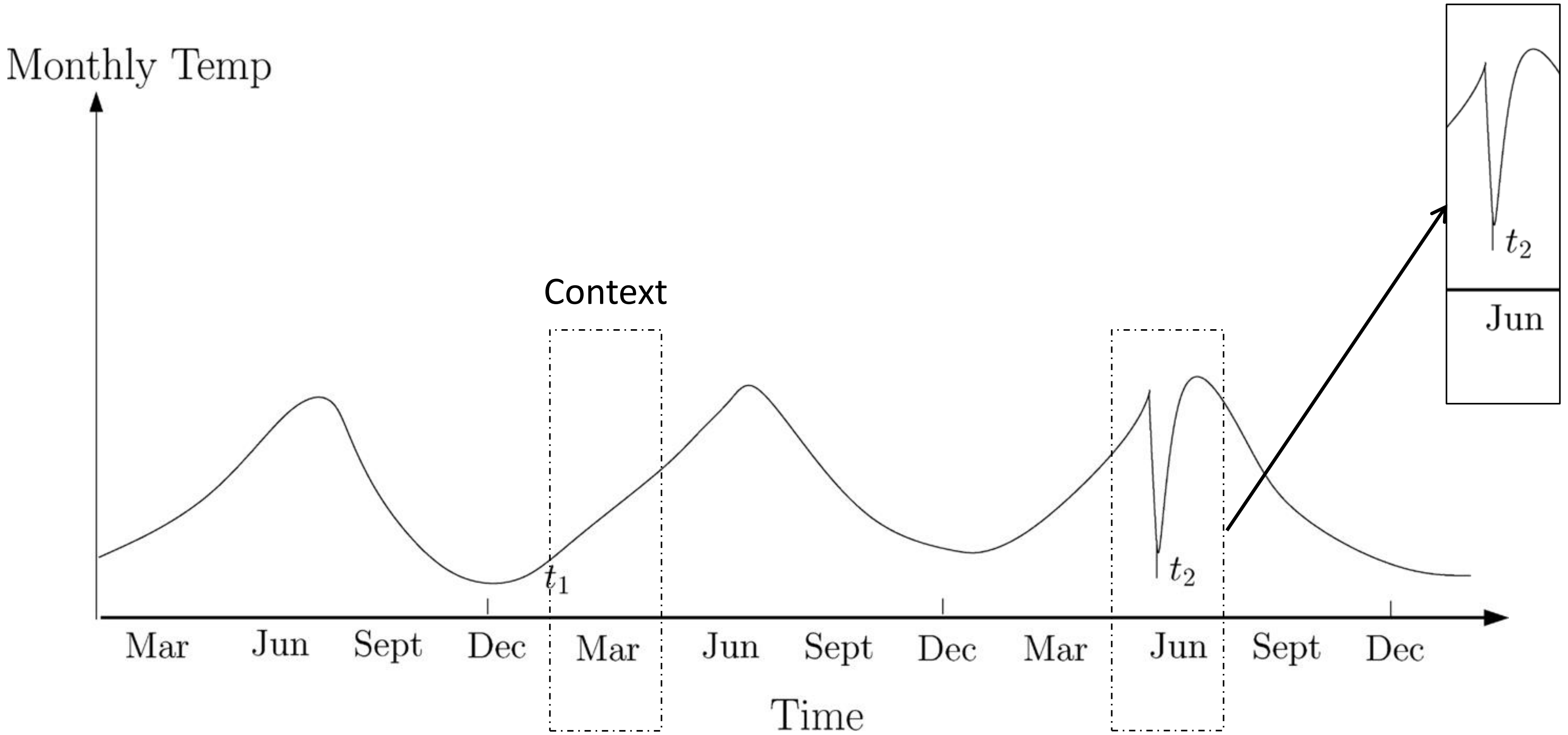
- **Proximity-based** techniques
  - Nearest-neighbors techniques
- **Density-based** techniques
  - Techniques where normal data instances are part of a cluster and anomalies do not belong to any cluster.
  - Techniques where normal data instances are close to a cluster centroid, and anomalies are far away from cluster centroids.
  - Techniques where normal instances belong to dense clusters and anomalies to small or sparse clusters.

Methodology - Contextual anomalies detection (I)

- The methods to handle contextual anomalies (and data) can be split in two categories
  - **Reduction to point anomaly** detection problems
    - The idea of these techniques is to first identify the context of a data instance from its contextual attributes, and then use a point anomaly detection technique trained using the normal data instances of the same context
  - Utilizing the **structure of the data**
    - These techniques are useful when separating the data into different predefined contexts is not doable, such as often with sequence and event data. In these cases, modeling techniques specific to **time-series** and discrete sequence data are needed.
    - E.g. change-point detection methods

Methodology - Contextual anomalies detection (II)

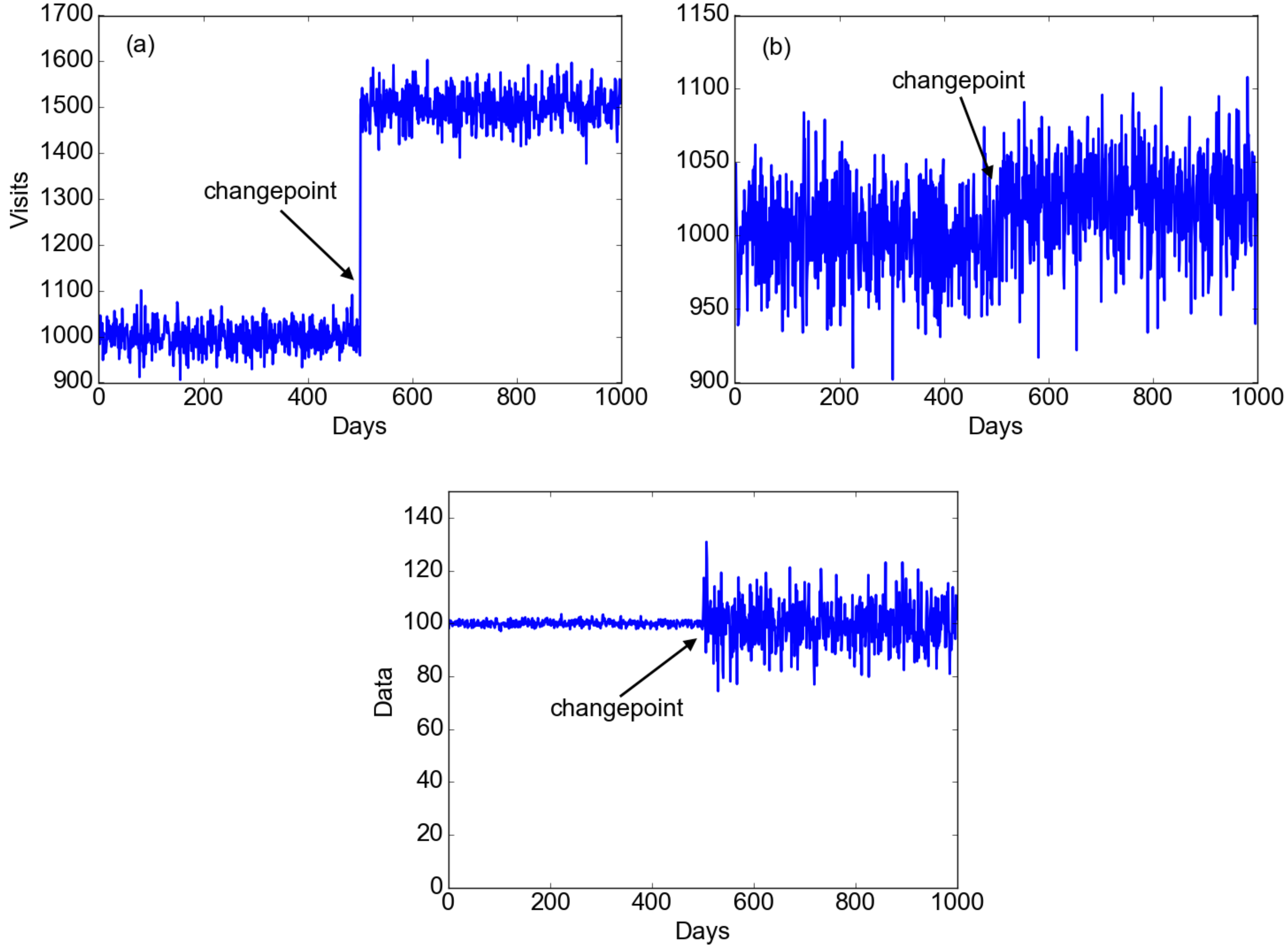
- **Reduction to point anomaly** detection problems



*Now you can apply **point anomalies** detection techniques*

Methodology - Contextual anomalies detection (III)

- Using the **structure of the data: ex. change point detection**



Source: www.claudiobellei.com/2016/11/15/changepoint-frequentist/

Methodology – Collective anomalies detection

- **Specialized techniques are needed** to detect collective anomalies and this problem is **more challenging** because the algorithms need to represent the structure of the data and relations between the data instances.
- Again, the **data structure** is used for the detection
  - The kind of data (sequential, spatial and graph) determines the approach to be used

Strengths and weakenesses

- Each of the large number of anomaly detection techniques discussed in the previous sections have their unique strengths and weaknesses. It is important to know which anomaly detection technique is best suited for a given problem.
    - For this lection: not feasible to provide such an understanding for **every** anomaly detection problem
- Strengths and weaknesses for a few simple problem settings

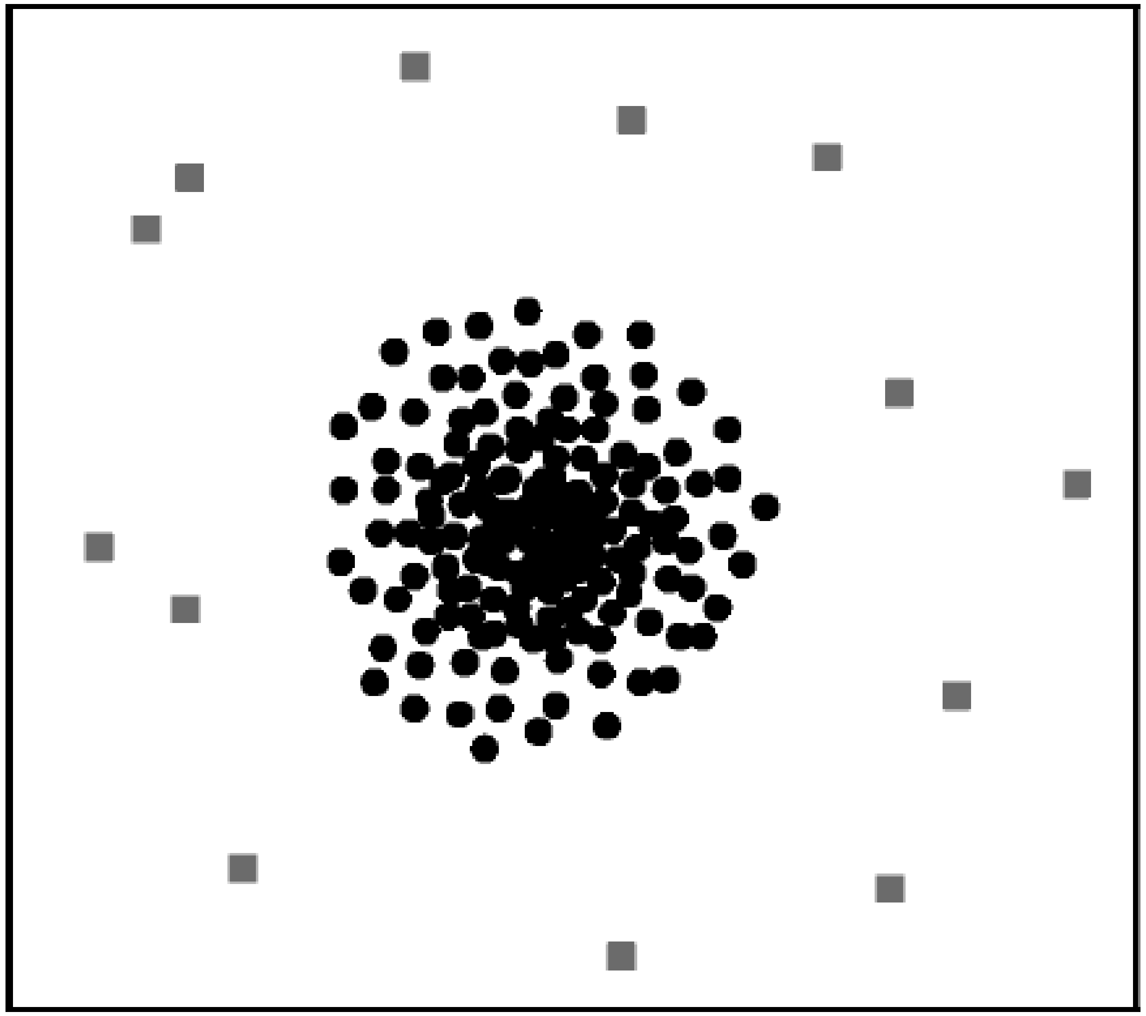
Exercise – Which technique to adopt?

- For the 3 data sets, fill the table in the next slide answering these questions:
  - Kind of anomaly (point, contextual, collective)
  - Techniques that can work

- Techniques to consider:
  - Model, proximity, or density-based

- For each technique, consider 2 cases:
  - Samples with anomalies <u>are presents</u> in the training set
  - Samples with anomalies <u>are NOT presents</u> in the training set (or they have not a label)

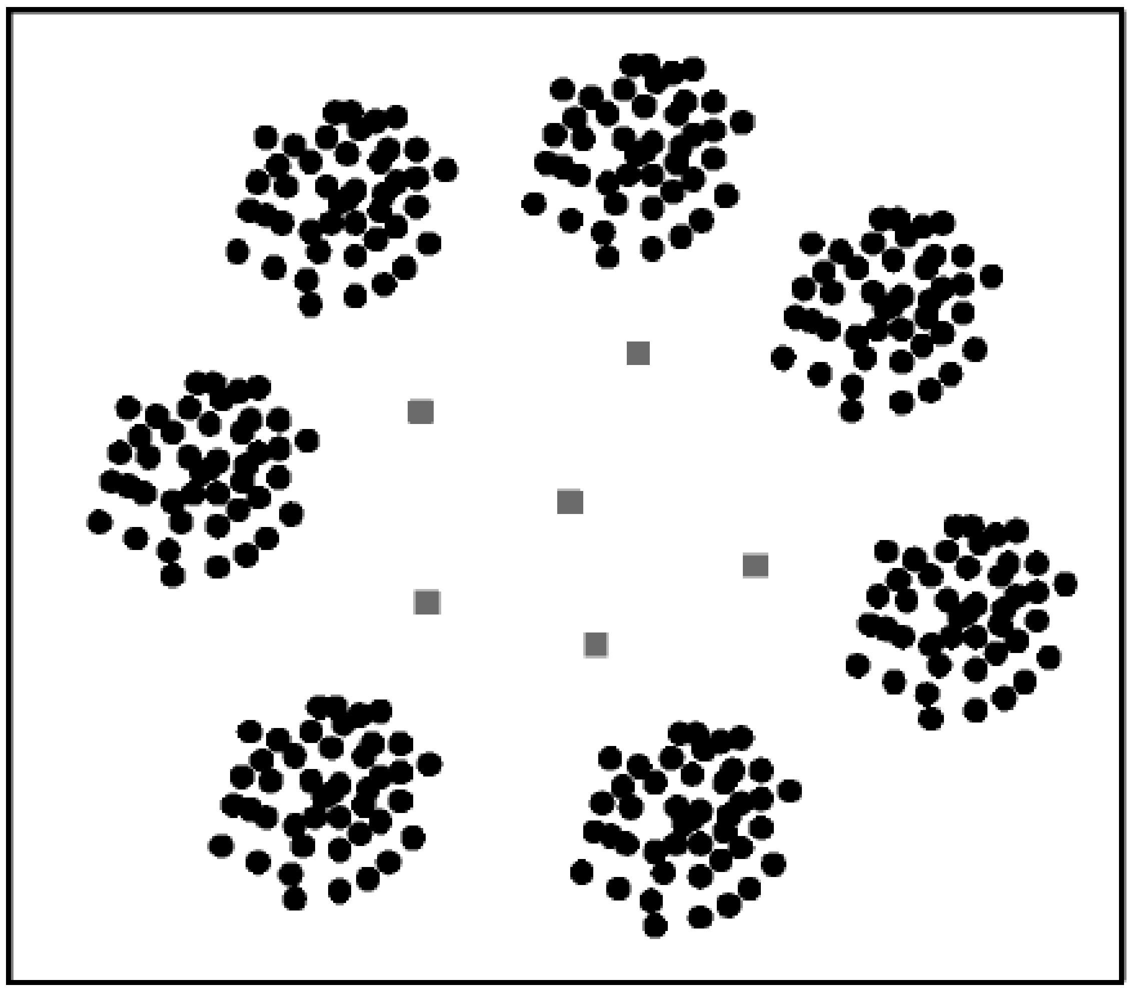
Exercise – Which technique to adopt?



(a) Data Set 1

Normal instances are shown as circles and anomalies are shown as square

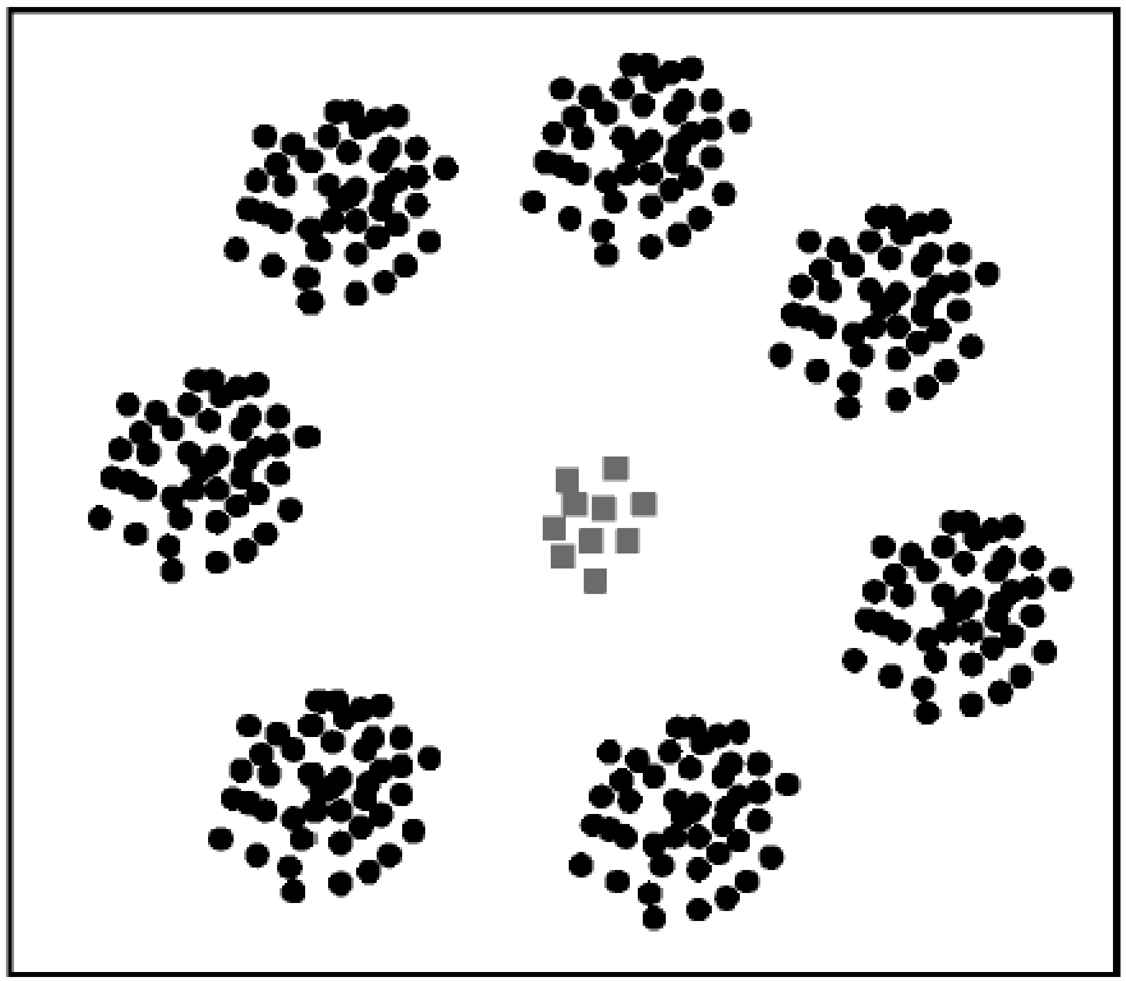
Exercise – Which technique to adopt?



(b) Data Set 2

Normal instances are shown as circles and anomalies are shown as square

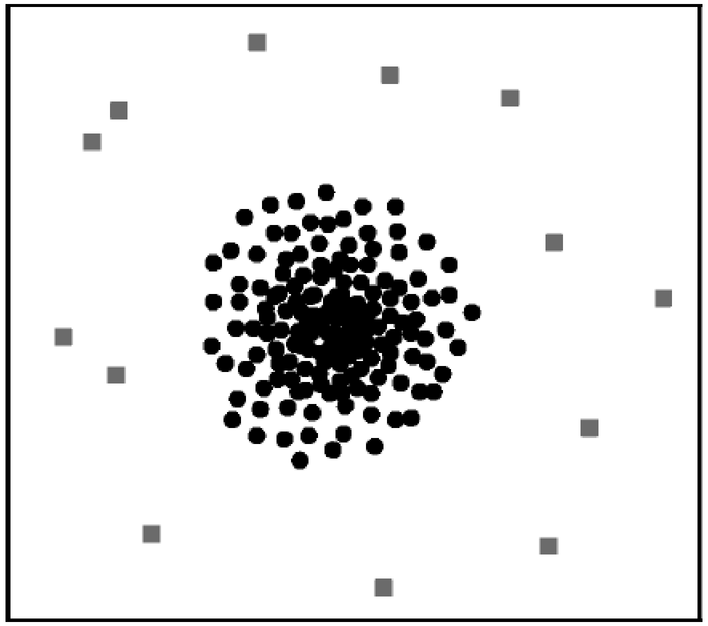
Exercise – Which technique to adopt?



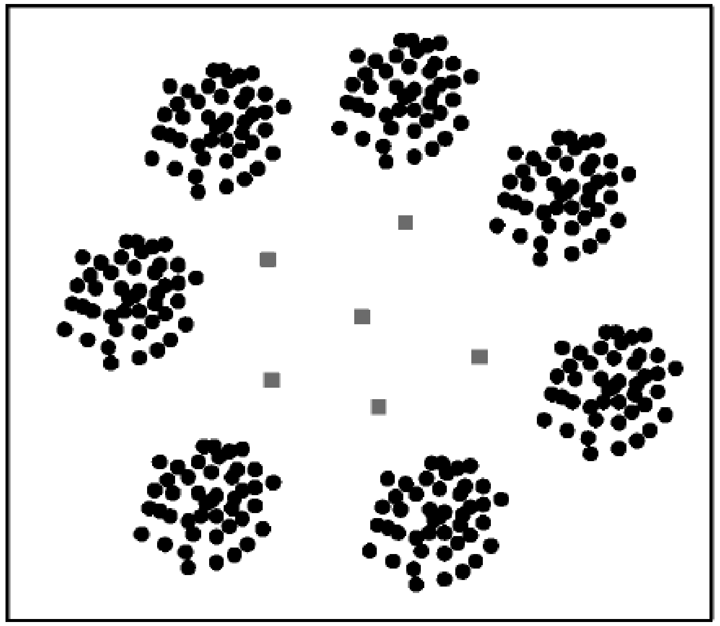
(c) Data Set 3

Normal instances are shown as circles and anomalies are shown as square

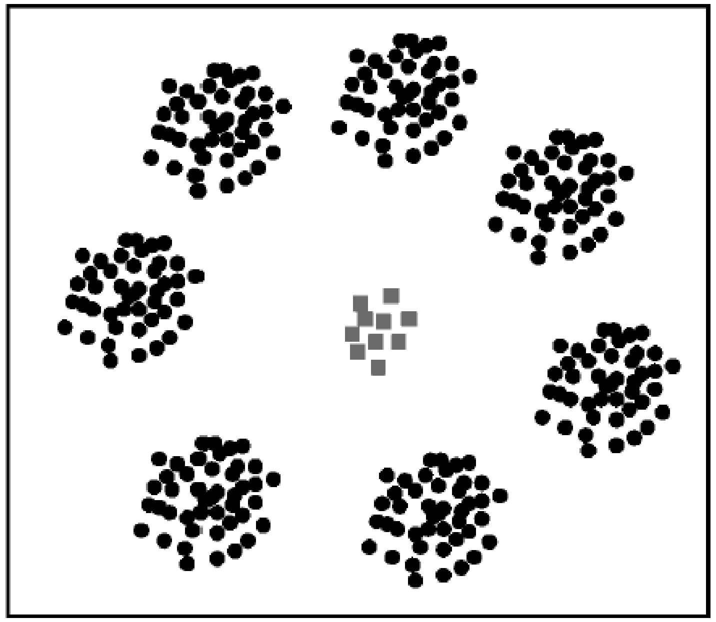
Exercise – Which technique to adopt?



(a) Data Set 1



(b) Data Set 2



(c) Data Set 3

Exercise – Which technique to adopt?

Data set	Model-based		Proximity-based		Density-based	
	With labels	Without labels	With labels	Without labels	With labels	Without labels
a) - Kind of anomaly:						
b) - Kind of anomaly:						
c) - Kind of anomaly:						

Strengths and weaknesses: scenario C

- **Model-based statistical techniques**, though unsupervised, are effective only **when the dimensionality of data is low** and statistical assumptions hold.
- **Proximity** and **density-based** techniques suffer when the number of dimensions is high, because the distance measures in a high number of dimensions are not able to differentiate between normal and anomalous instances
- **Spectral techniques** (including SVM) explicitly address the high dimensionality problem by mapping data to a lower dimensional projection. But their **performance is highly dependent on the assumption** that the normal instances and anomalies are distinguishable in the projected space.

Strengths and weaknesses: scenario C

Discussion: Reading at Home

- **Classification-based techniques can be a good choice** in such scenario. But to be most effective, classification-based techniques **require labels** for both normal and anomalous instances, which are not often available.
  - Even if the labels for both normal and anomalous instances are available, the imbalance in the distribution of the two labels often makes learning a classifier quite challenging.
- **Semi-supervised nearest neighbor and clustering techniques**, that only use the normal labels, can often be more effective than the classification-based techniques.

Algorithm Computational Complexity [1]

- *The **computational complexity** of an anomaly detection technique is a **key aspect**, especially when the technique is applied to a real domain.*
- **Classification-based, density-based, and model techniques**
  - Have **expensive training times**, **testing is usually fast**.
  - Often this is acceptable, since models can be trained in an offline fashion while testing is required to be in real time.

Algorithm Computational Complexity [2]

- *The **computational complexity** of an anomaly detection technique is a **key aspect**, especially when the technique is applied to a real domain.*
- Techniques such as **nearest neighbor-based**, information theoretic, and spectral techniques
  - **do not have a training phase**, have an **expensive testing phase** which can be a limitation in a real setting.

SciKit - Algorithms

- In SciKit anomalies are generally presented under two different names:
  - Outlier / Novelties
- **Outliers**:
  - The training data contains outliers, and we need to fit the central mode of the training data, ignoring the deviant observations.
- **Novelties**:
  - The training data is not polluted by outliers, and we are interested in detecting anomalies in new observations.

Novelties detection

- *Consider a data set of observations from the same distribution described by features. Consider now that we add one more observation to that data set. Is the new observation so different from the others that we can doubt it is regular? (i.e. does it come from the same distribution?) Or on the contrary, is it so similar to the other that we cannot distinguish it from the original observations?*

- These are the questions addressed by the novelty detection tools and methods.

- Algorithm available (on scikit-learn)
    - One-class SVM

Novelties detection

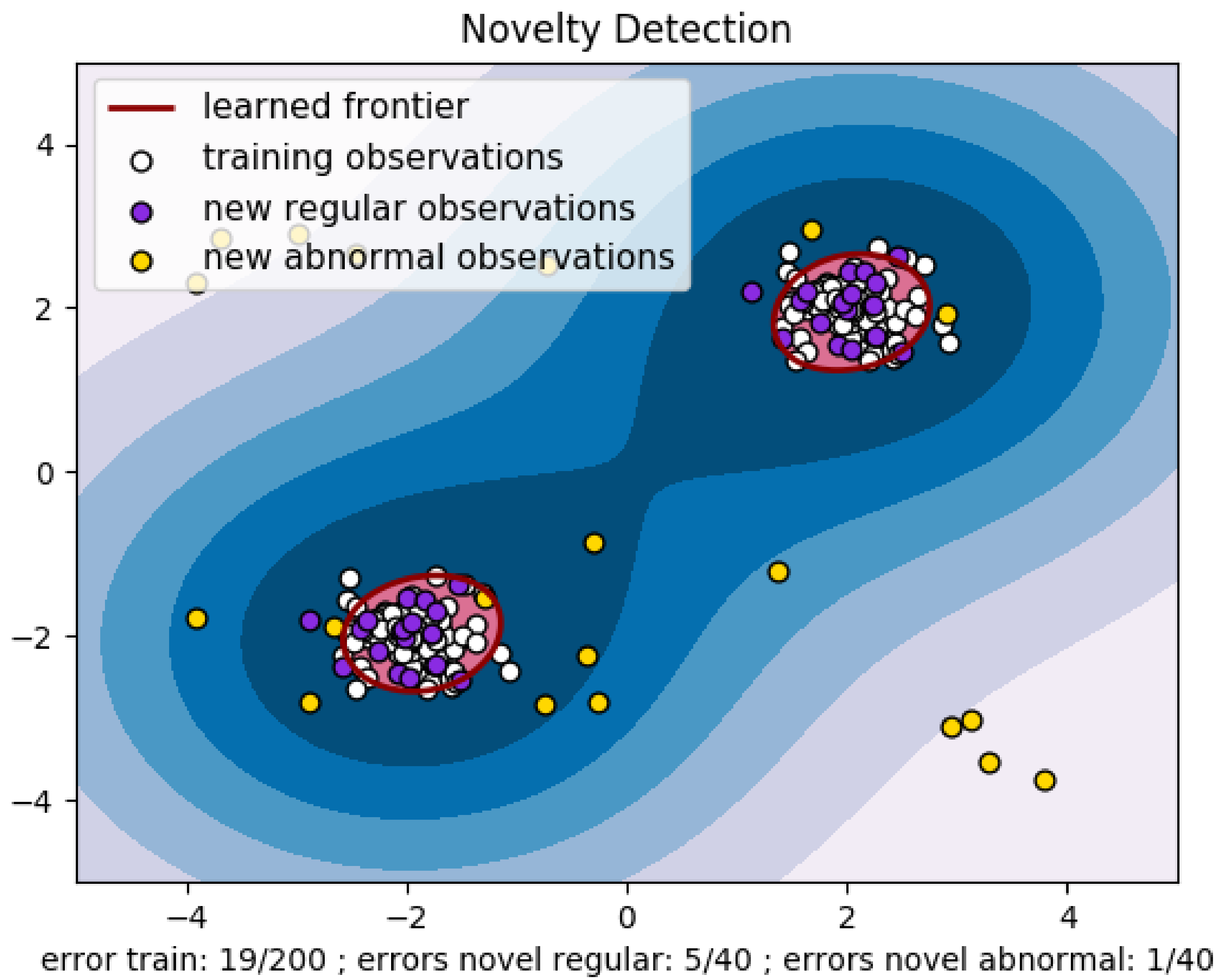


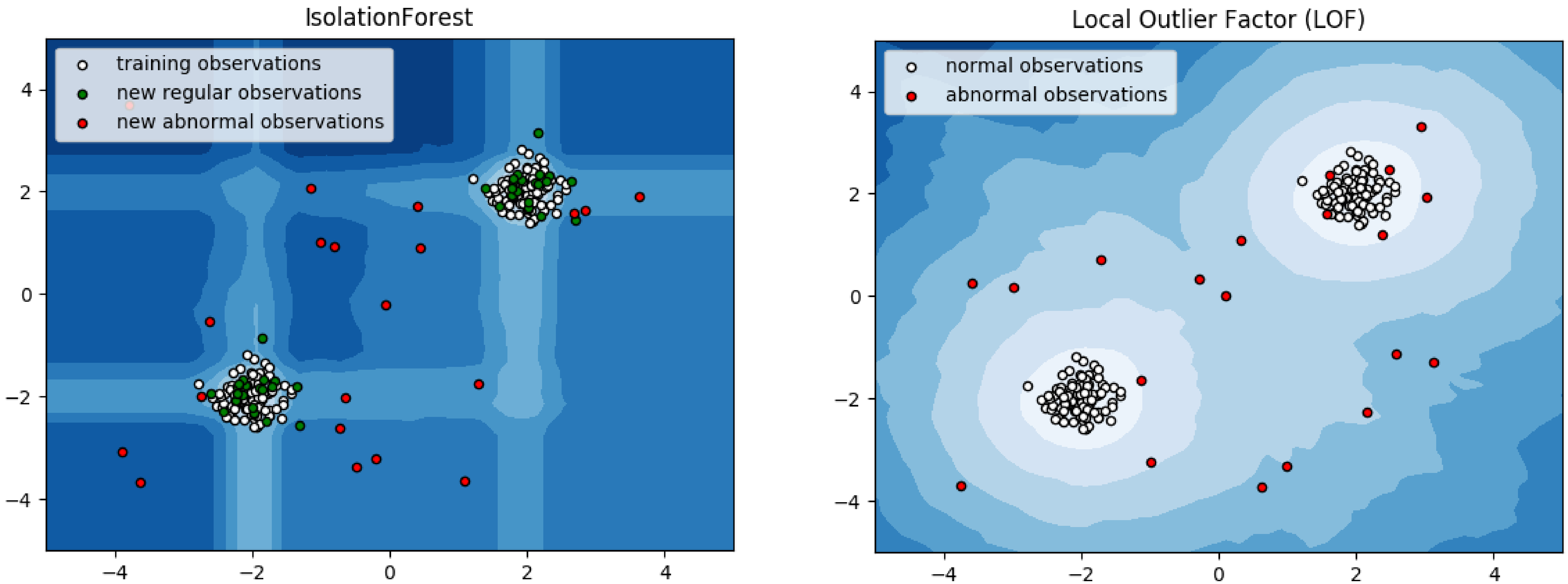
Image source: http://scikit-learn.org/stable/modules/outlier_detection.html

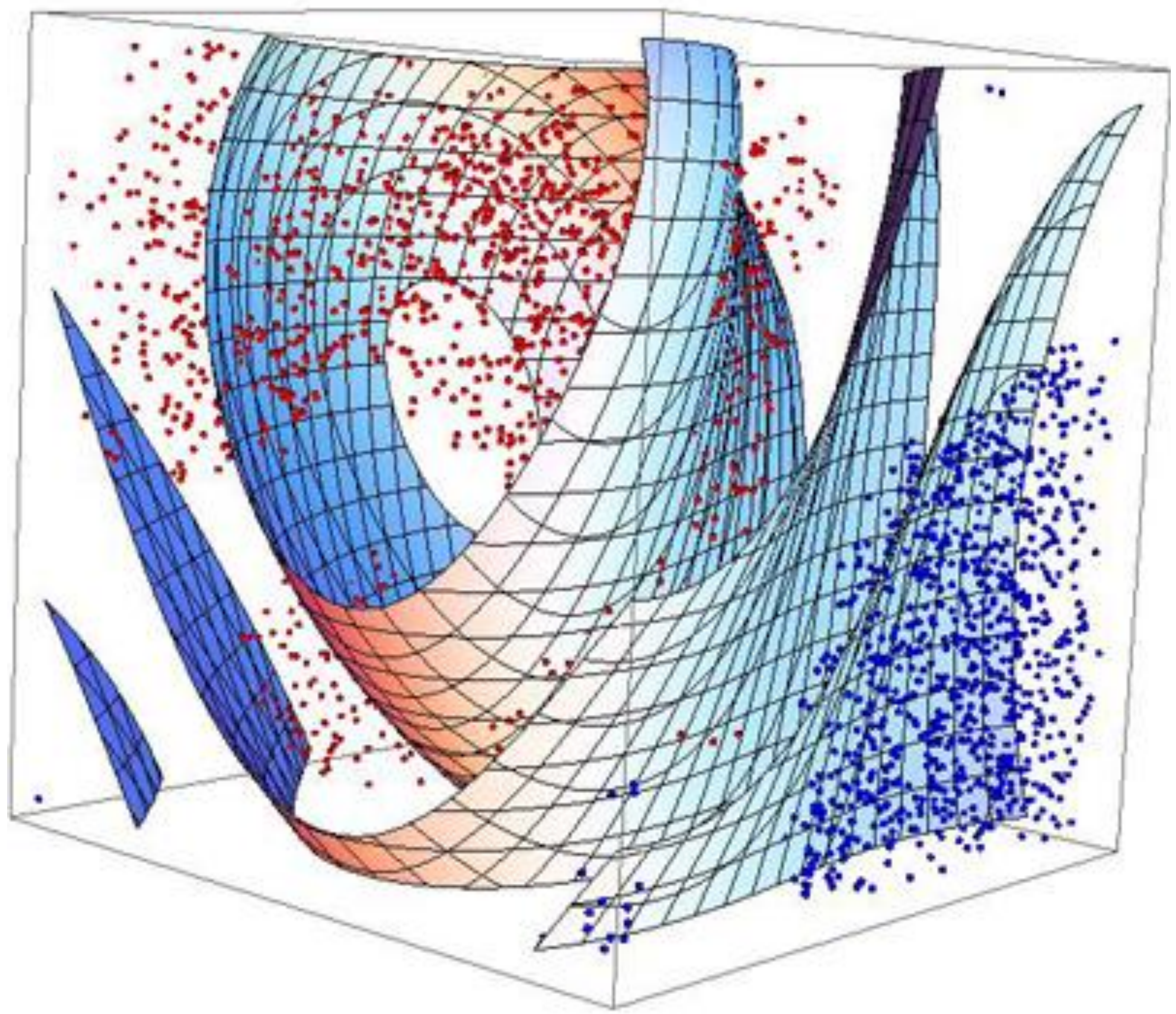
Outlier detection

- Outlier detection is similar to novelty detection in the sense that the goal is to separate a core of regular observations from some polluting ones, called “outliers”. Yet, in the case of outlier detection, we don’t have a clean data set representing the population of regular observations that can be used to train any tool.

- Algorithms available (on scikit-learn)
  - Insolation forest
  - Local Outlier Factor¶

Outlier detection





SUPPORT VECTOR MACHINES

SVM - Recall – Classification Problem

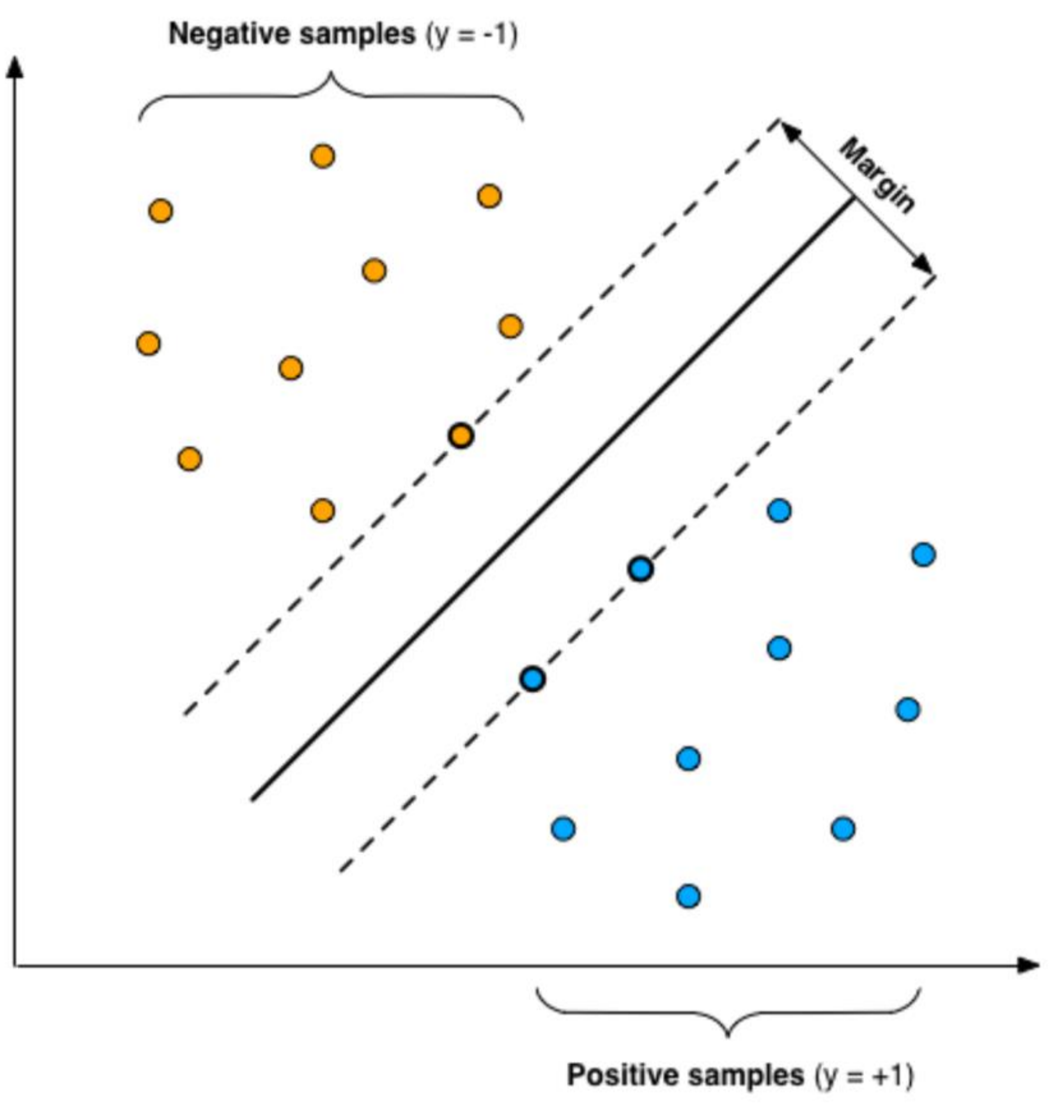
- SVM, GMM, HMM are called **supervised classifiers**
- **Train samples are data points**
  - whose **class labels are known** by a classifier
  - which are used by a classifier **to build class models or separating hyperplanes** during the so-called **train** (or learning) phase
- **Test samples are data points**
  - whose **class labels are unknown** by a classifier
  - whose **class labels must be assigned by a classifier** by comparing them to the previous class models or separating hyperplanes during the so-called test (or classification) phase

SVM

An SVM finds the optimal **hyperplane**, i.e. the one which maximizes the margin between the 2 classes.

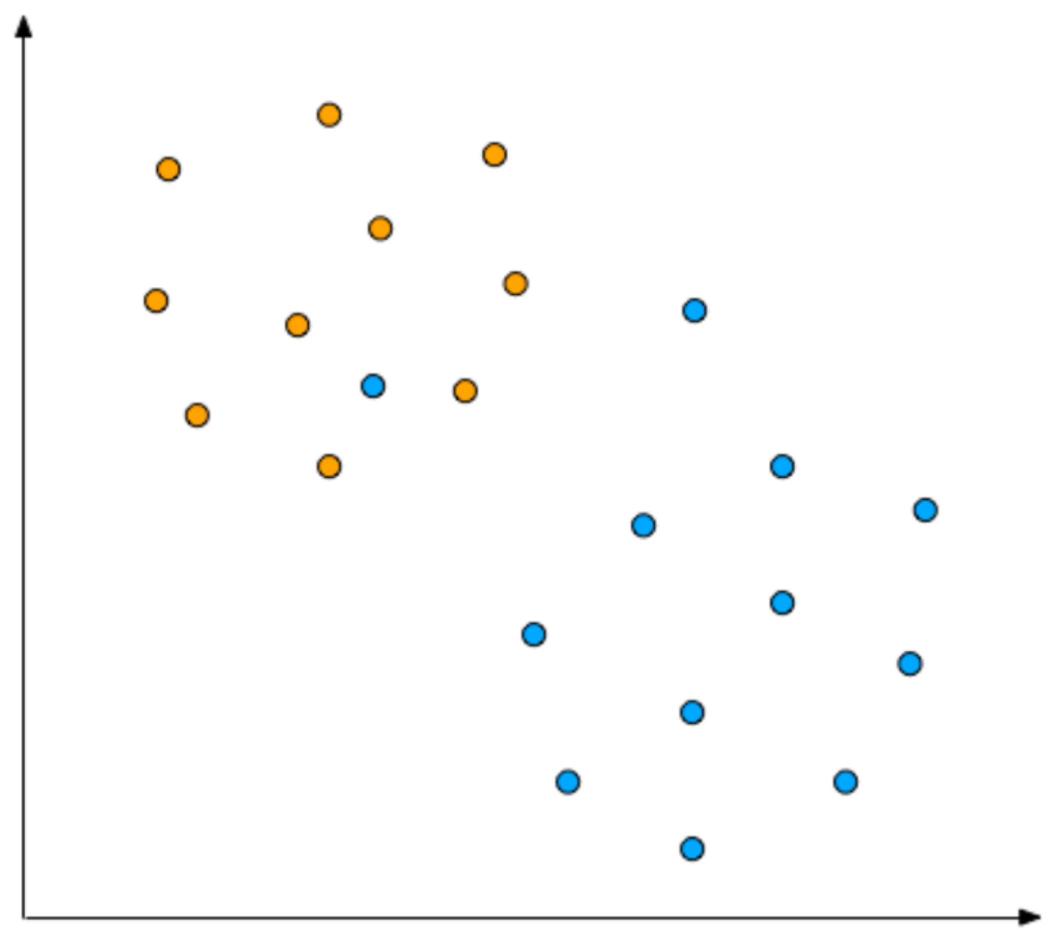
(Samples on the margin boundaries are called **support vectors**).

For mathematical reasons, the SVM algorithm finds the best hyperplane.



SVM – Non linear data [1]

Problem: How to linearly separate these sets?



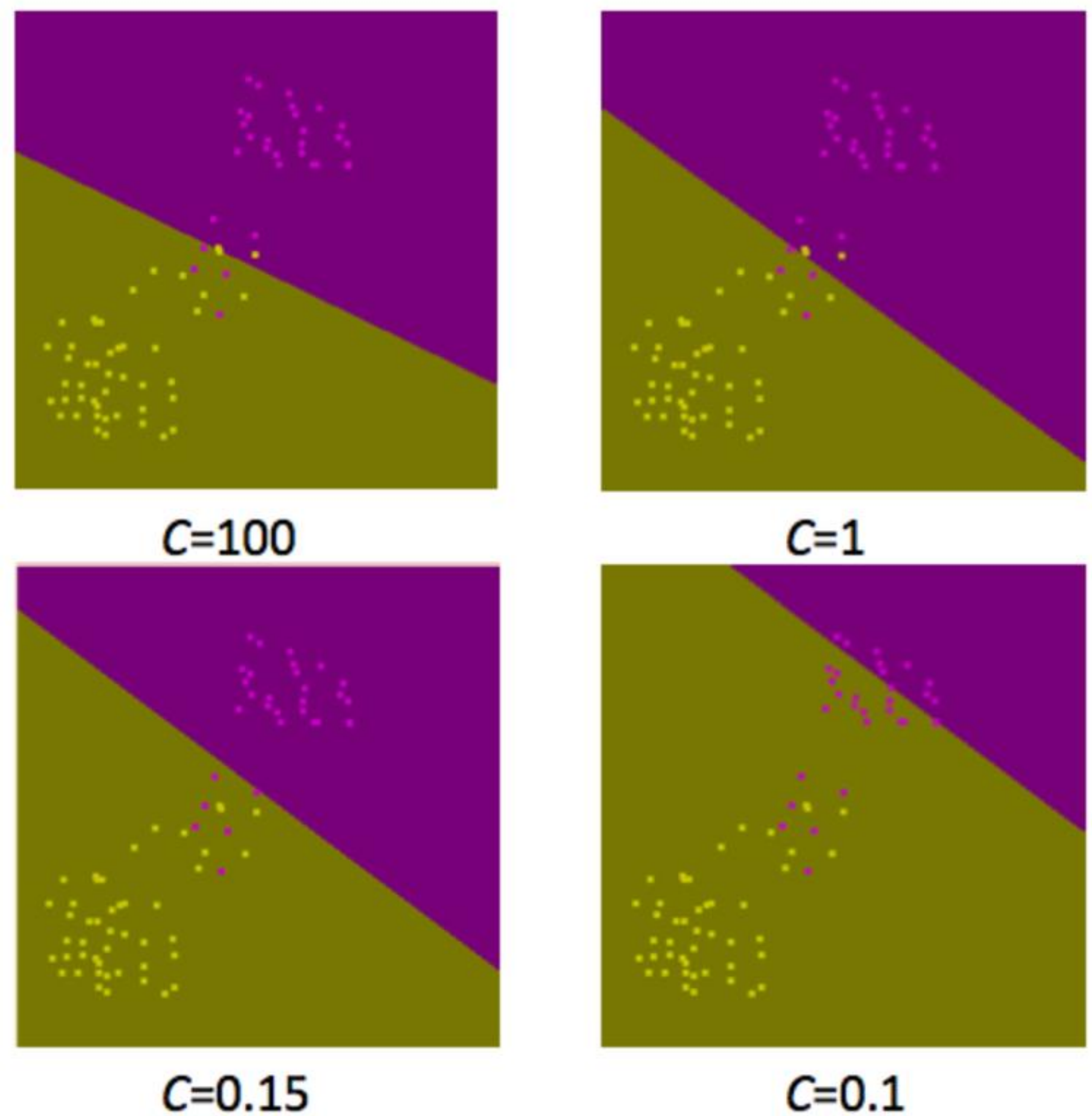
Solution: Soft margins

SVM – Non linear data [2]

SVM - Hyperparameters

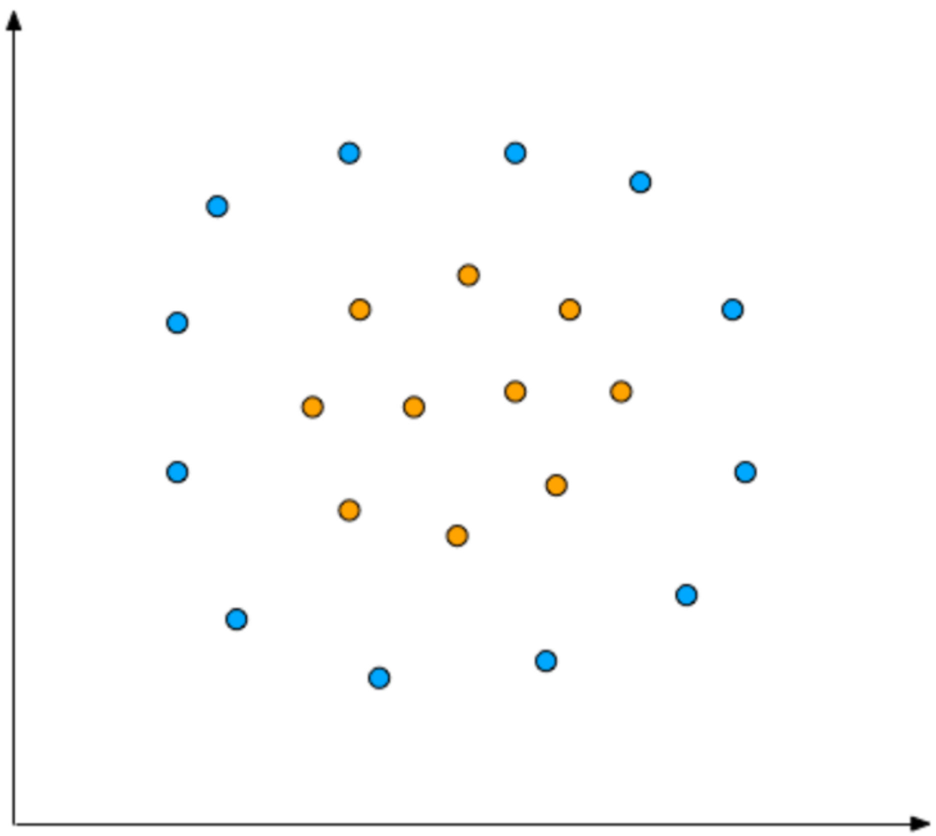
The factor **C** is a **regularization parameter** which **trades off the margin size and the training error**

The smaller C, the greater the number of admitted misclassified train samples



SVM – Non linear data [3]

Problem: how to linearly separate these 2 classes of samples?

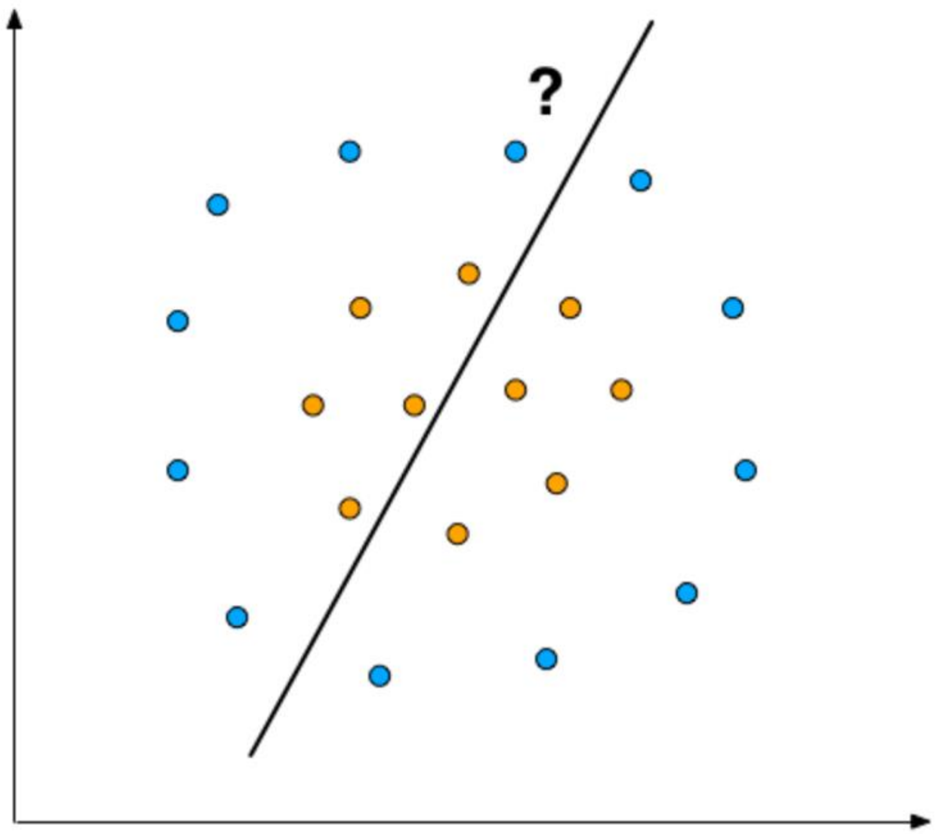


SVM – Non linear data [3]

Problem: how to linearly separate these 2 classes of samples?

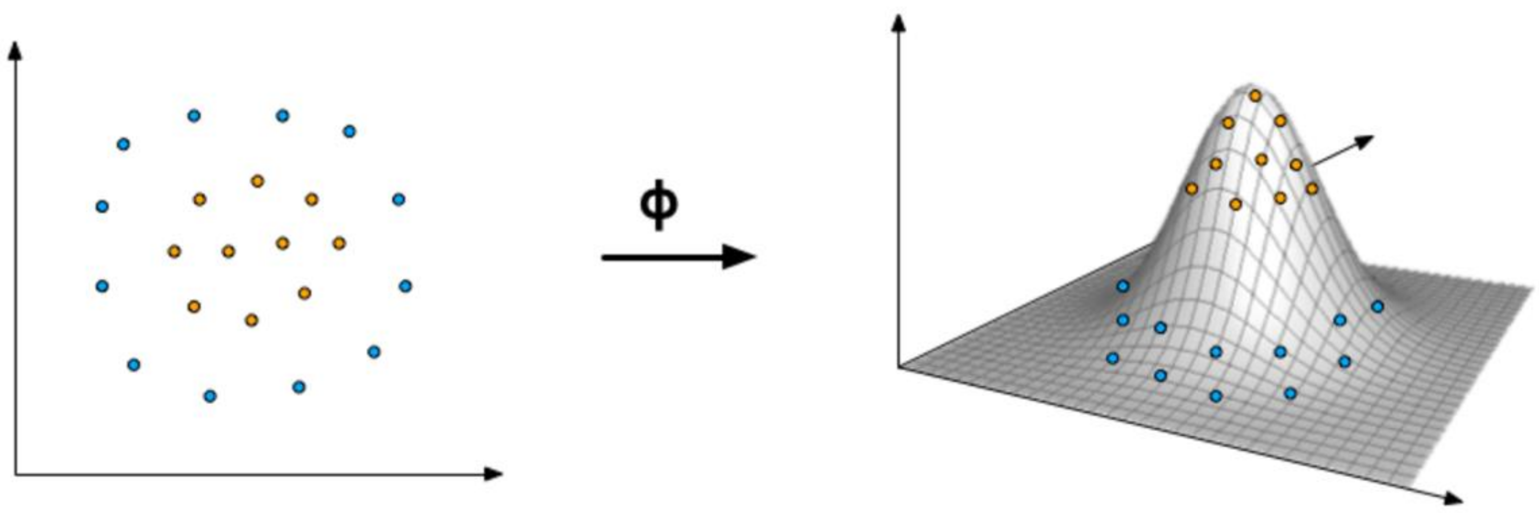
~~Answer 1) Again, samples are not linearly separable!~~

Answer 2) Kernel Trick!



SVM – Non linear data [4]

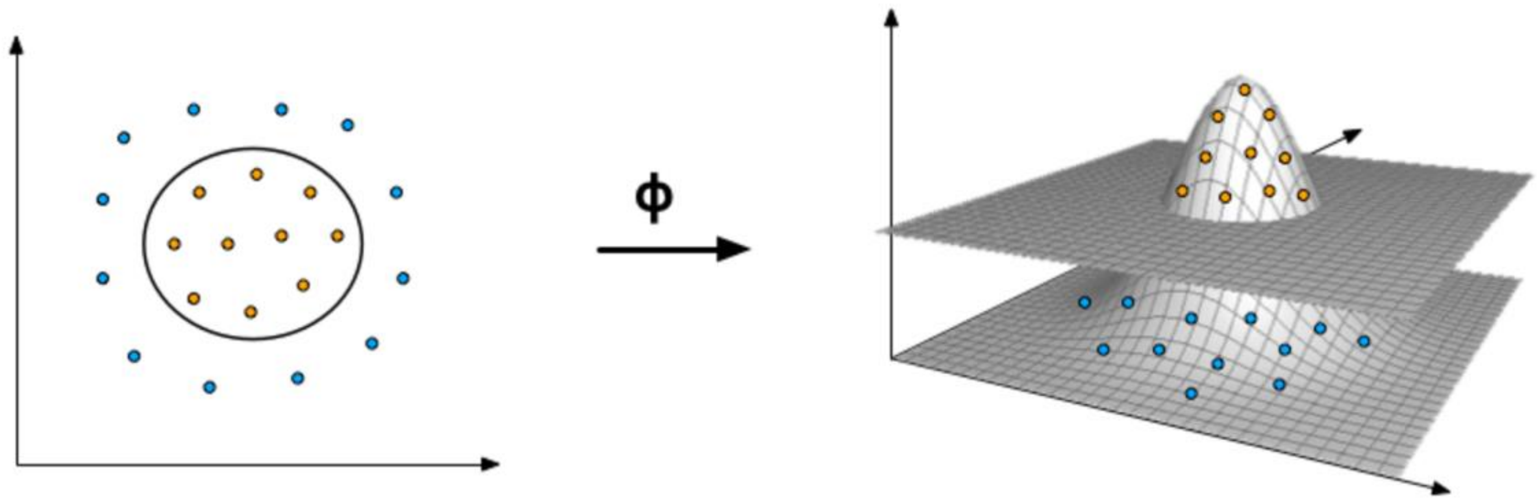
Map the sample **input space to** a higher dimensional space (called **feature space**) with a function ϕ



SVM – Non linear data [5]

Map the sample **input space to** a higher dimensional

space (called **feature space**) with a function **ϕ**



Example: https://youtu.be/9NrALgHFwTo

SVM – Non linear data [6]

- **Common kernels**
  - Linear
  - Polynomial
  - Gaussian or radial basis function (RBF)

  - Other kernels exist
    - Hyperbolic tangent, …

SVM – Hyperparameters (I)

- **Linear Kernel – C**
- C = Cost parameter [0, ∞(
  - The **C parameter trades off misclassification of training examples against simplicity of the decision surface**. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly by giving the model freedom to select more samples as support vectors.

SVM – Hyperparameters (II)

- **RBF Kernel – C & Gamma**
- C = Cost parameter [0, ∞(
  - The **C parameter trades off misclassification of training examples against simplicity of the decision surface**. A low C makes the decision surface smooth, while a high C aims at classifying all training examples correctly by giving the model freedom to select more samples as support vectors.
- Gamma [0, ∞(
  - Intuitively, the gamma parameter defines **how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'**. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

SVM – Hyperparameters (III)

- **Polynomial Kernel – Degree & Gamma**
- Degree
  - Simply the degree of the polynomial used for the kernel trick
- Gamma [0, ∞(
  - Intuitively, the gamma parameter defines how far the influence of a single training example reaches, with low values meaning ‘far’ and high values meaning ‘close’. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

How to select the best parameters?

- **Grid Search** (typical approach)
  - Select several models
  - Train each of the models and evaluate it using cross-validation.
  - In practice for **C** and **γ**:
    - Try exponentially growing sequences of C and γ is a practical method to identify good parameters
    - For example, $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}$, $\gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$
    - NOTE: very easy to implement in scikit-learn
- Alternative: **Random Grid Search**

Suggested reading: "*A Practical Guide to Support Vector Classification*": http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

Suggestions [1]

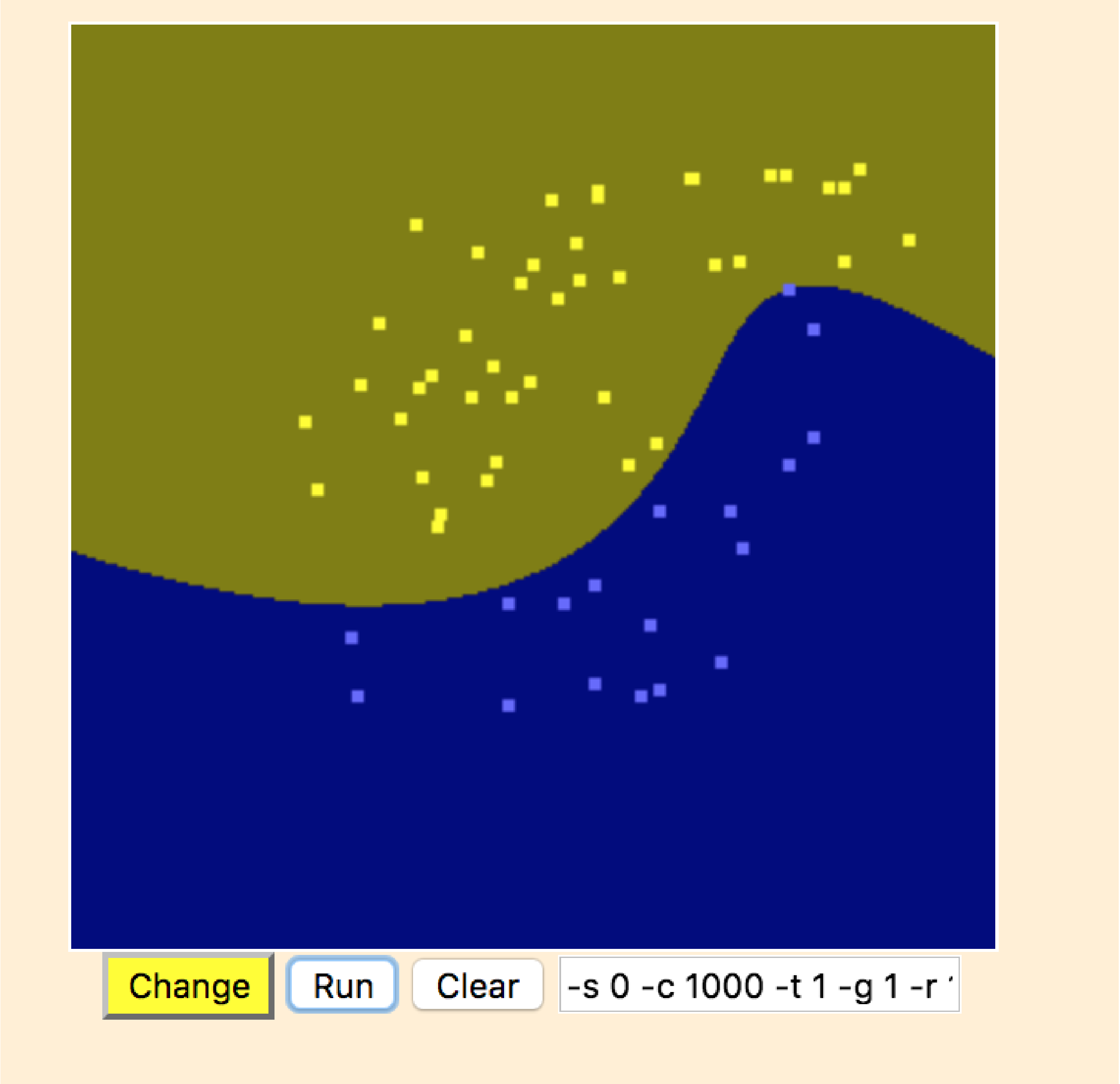
- RBF
  - Typically the first choice
  - Deal well with non-linearity
  - More generic approach (Linear kernel can be seen as a special case of RBF)
- Linear kernel
  - Faster
  - Good for very complex, high-dimensional dataset
- Polynomial
  - In between
  - Good performances
  - Many hyper-parameters

Suggestions [2]

- Use linear kernel when number of features is larger than number of observations.
- Use RBF kernel when number of observations is larger than the number of features.
- If number of observations is larger than 50,000* speed could be an issue when using RBF kernel; hence, one might want to use linear kernel.

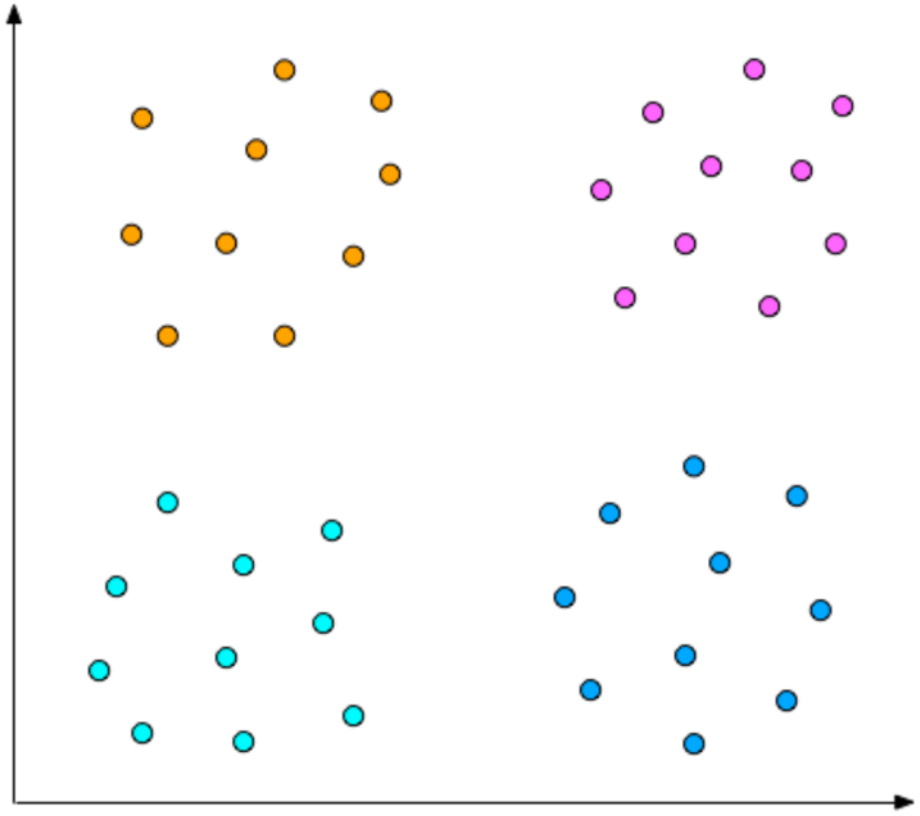
Playing Around

- In practice (thank you libsvm!):
  - https://www.csie.ntu.edu.tw/~cjlin/libsvm/#nuandone



Multiclass

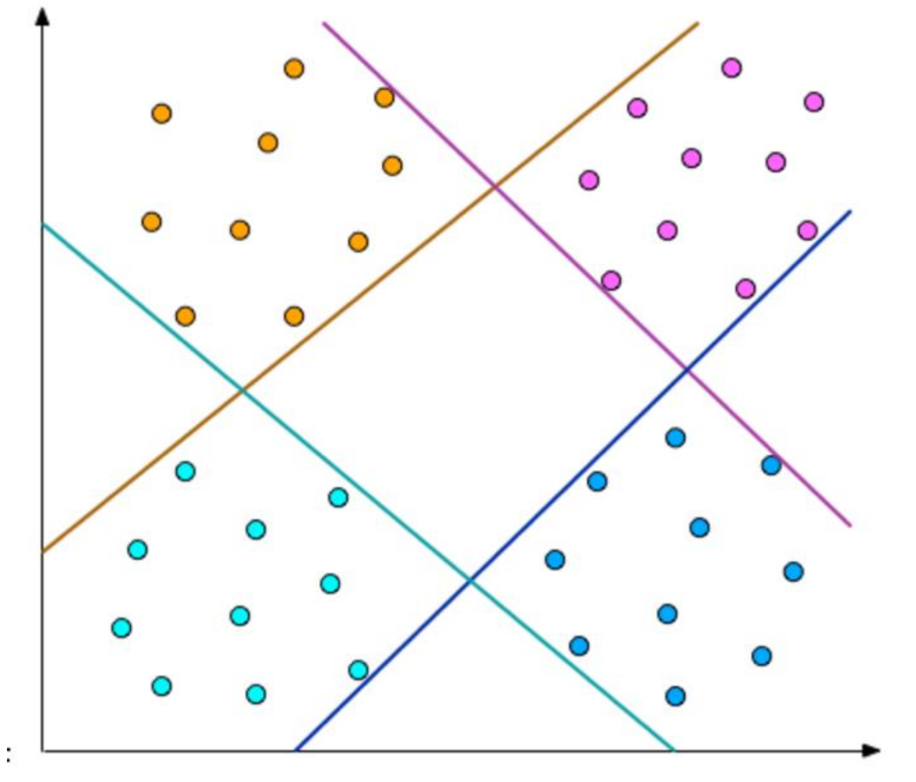
SVM are binary classifiers, then how to deal with multiple classes?



Multiclass (I)

One Vs All – the classification of new samples is done by a winner-takes-all strategy, in which **the SVM with the highest output value assigns the class to a given sample**

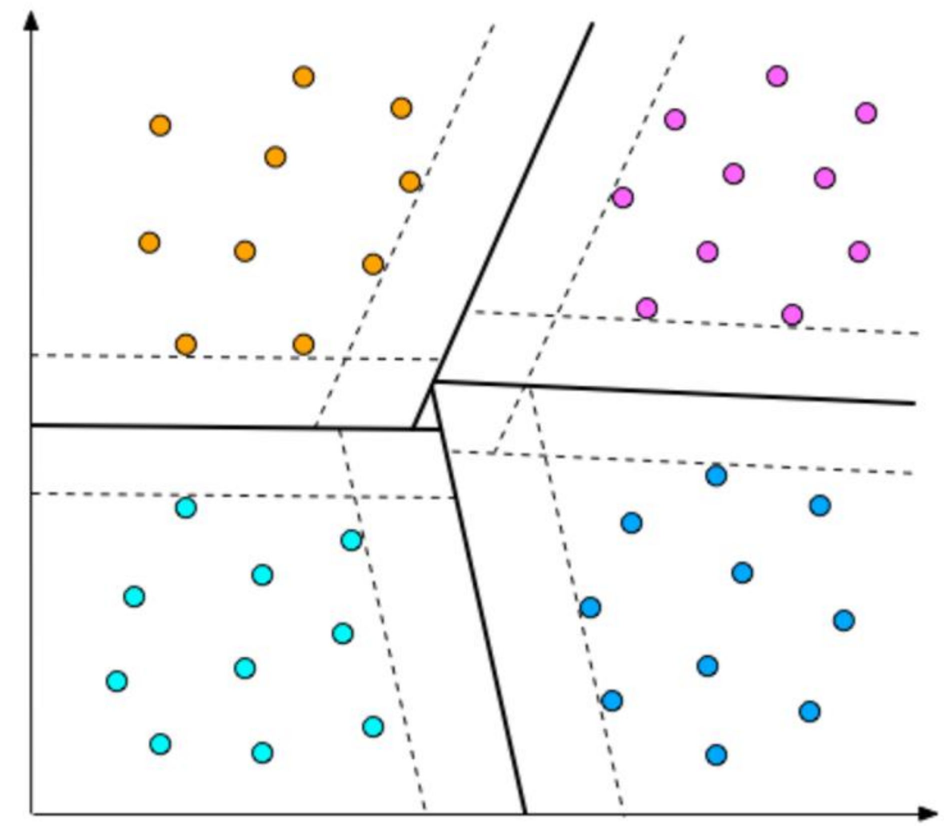
1 classifier per class



Multiclass (II)

One Vs One – the classification of new samples is done by a max-wins voting strategy. In one vs one you have to train a separate classifier for each different pair of labels. This leads to N*(N−1)/2 classifiers.

Every SVM classifier assigns a given sample to one of the two classes, the class with the highest number of votes is assigned to the sample



1 classifier for each pair of labels

Multiclass (III)

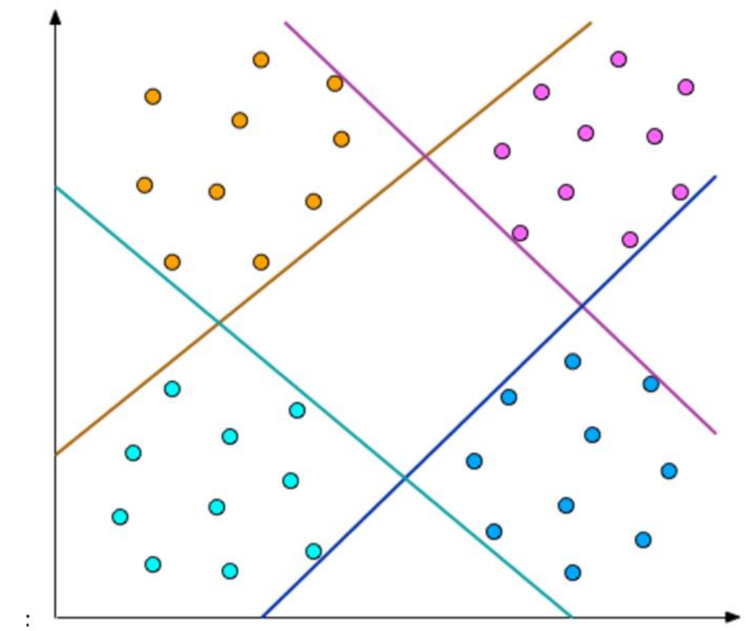
Drawbacks

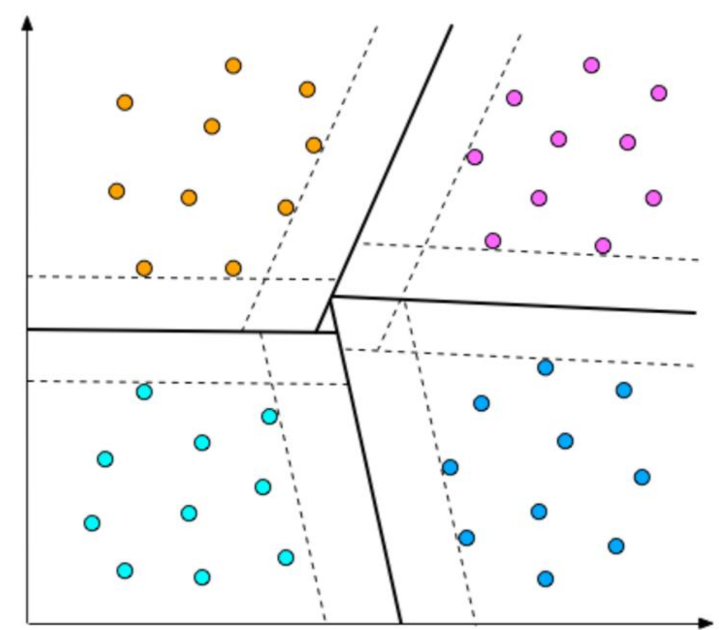
One Vs All:

- Unbalanced datasets!

One Vs One:

- Computationally expensive!





One-Class SVM

SVM FOR NOVELTY AND OUTLIER DETECTION

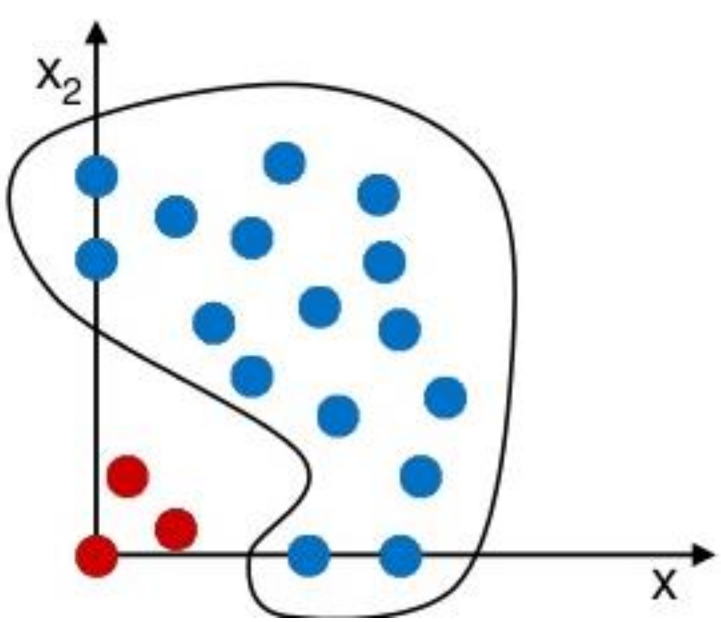
Beyond classification

- SVM can also be used as **unsupervised algorithms**
- A specific use case is the anomaly detection and the **novelty detection** in particular
  - novelty detection: given a set of samples, to detect the soft boundary of that set so as to classify new points as belonging to that set or not.

- The used approach is called: **One-Class SVM**

One-Class SVM

- One-class SVM is an **unsupervised** algorithm that learns a decision function for novelty detection: classifying new data as similar or different to the training set.
- In this case, as it is a type of unsupervised learning, the fit method will only take as input an array X, as there are no class labels.



How it works (in simple words)

- One Class SVM aims at finding the smaller **hyper-sphere** circumscribing the items in high-dimensional space

- In practice (thank you again libsvm!):
  - https://www.csie.ntu.edu.tw/~cjlin/libsvm

One-Class SVM - Use Cases

- This approach is particularly useful in scenarios where you have **a lot of "normal" data** and not many cases of the anomalies you are trying to detect.
  - E.g. detect fraudulent transactions
- The One-Class SVM is **able to capture the real data structure**
- Difficulty is to adjust its kernel parameters..
  - …to obtain a good compromise between the shape of the data scatter matrix and the risk of overfitting the data.

Anomaly detection & SVM

WHAT YOU SHOULD KNOW

Anomaly detection

- **Definitions** and goals
- Anomaly characterization and their meaning for a data analyst
  - the **nature** of the **input data**
  - the **type** of anomalies
    - **Point**, **Contextual** and **Collectives**
  - the availability of **labeled data**
  - the **output constraints**
- SVM and Anomalies
  - **One-Class SVM**

Support Vector Machines

- SVM **functioning principles** (not mathematics)
- How a **linear SVM** tries to deal with **linearly separable data**?
  - hyper-plan with maximal margin,...
- How a **linear SVM** tries to deal with **not linearly separable data**?
  - Soft margin
- How a **nonlinear SVM** works?
  - Map samples to higher dimensional space with function **ϕ**, kernel trick principle, common kernel types, hyperparameters...)
- How **multiclass SVM** works?
  - one-vs-all and one-vs-one methods