Anomaly Detection and Domain Adaptation

Giacomo Boracchi, Francesco Trovò

May 18th, 2022

Politecnico di Milano, DEIB

giacomo.boracchi@polimi.it

Outline

- Anomaly Detection in the random variable world
- Out of the «Random Variable World»
  - Reconstruction based-methods
  - Feature-based methods
- Domain Adaptation
- Bonus slides?

Statistical Approaches to Anomaly Detection

..a different monitoring problem

The Change/Anomaly Detection Problems

Change-detection problem:

*Given the previously estimated model, the arrival of new data invites the question: "**Is yesterday's model capable of explaining today's data?**"*

Detecting process changes is important to understand the monitored phenomenon

Anomaly-detection problem:

*Locate those **samples that do not conform the normal ones** or a **model explaining normal ones***

Anomalies in data translate to significant information

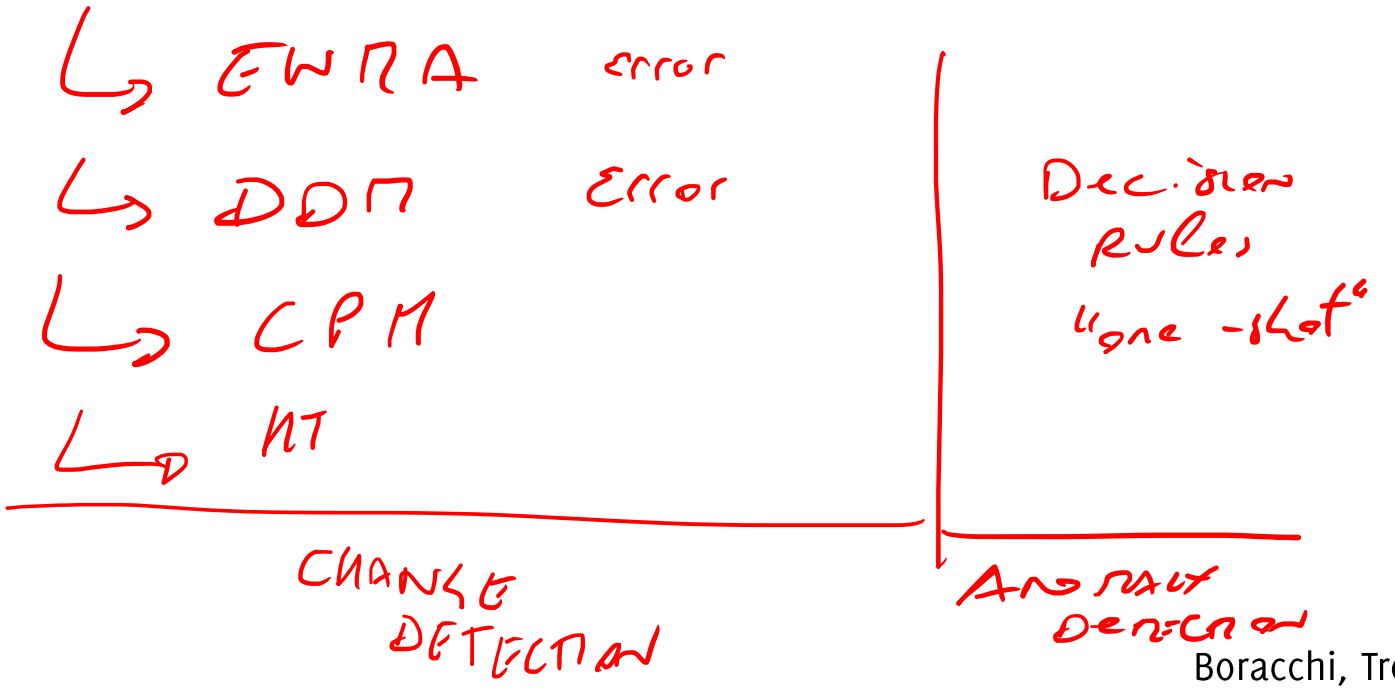
V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

C. J. Chu, M. Stinchcombe, H. White "*Monitoring Structural Change*" Econometrica Vol. 64, No. 5 (Sep., 1996), pp. 1045-1065.

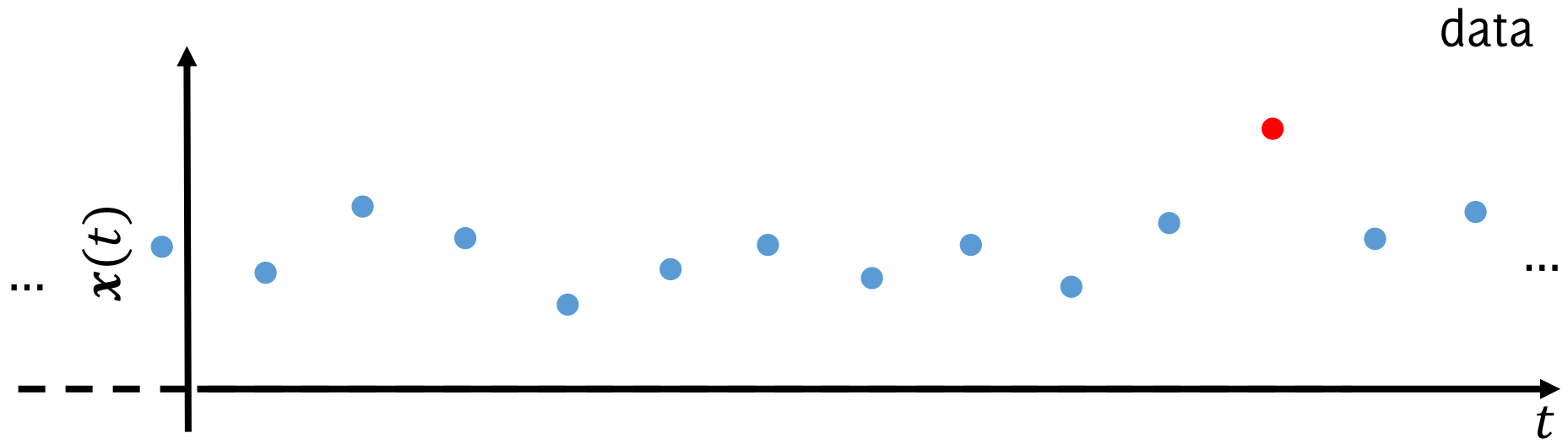
The Typical Anomaly Detection Solutions

Most algorithms are composed of:

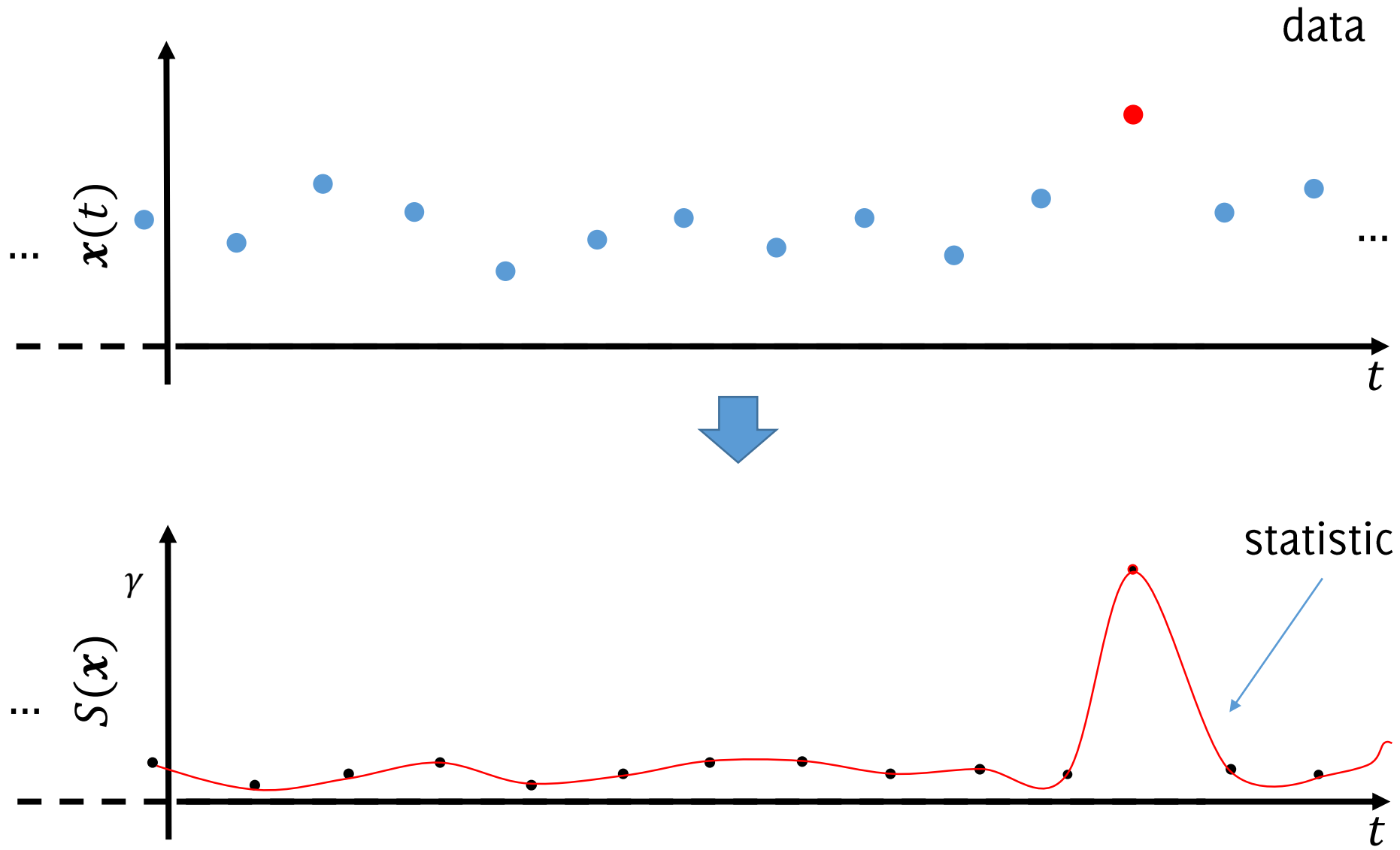
- A **statistic** that has a known response to normal data (e.g., the average, the sample variance, the log-likelihood, the confidence of a classifier, an "anomaly score"…)
- A **decision rule** to analyze the statistic (e.g., an adaptive threshold, a confidence region)



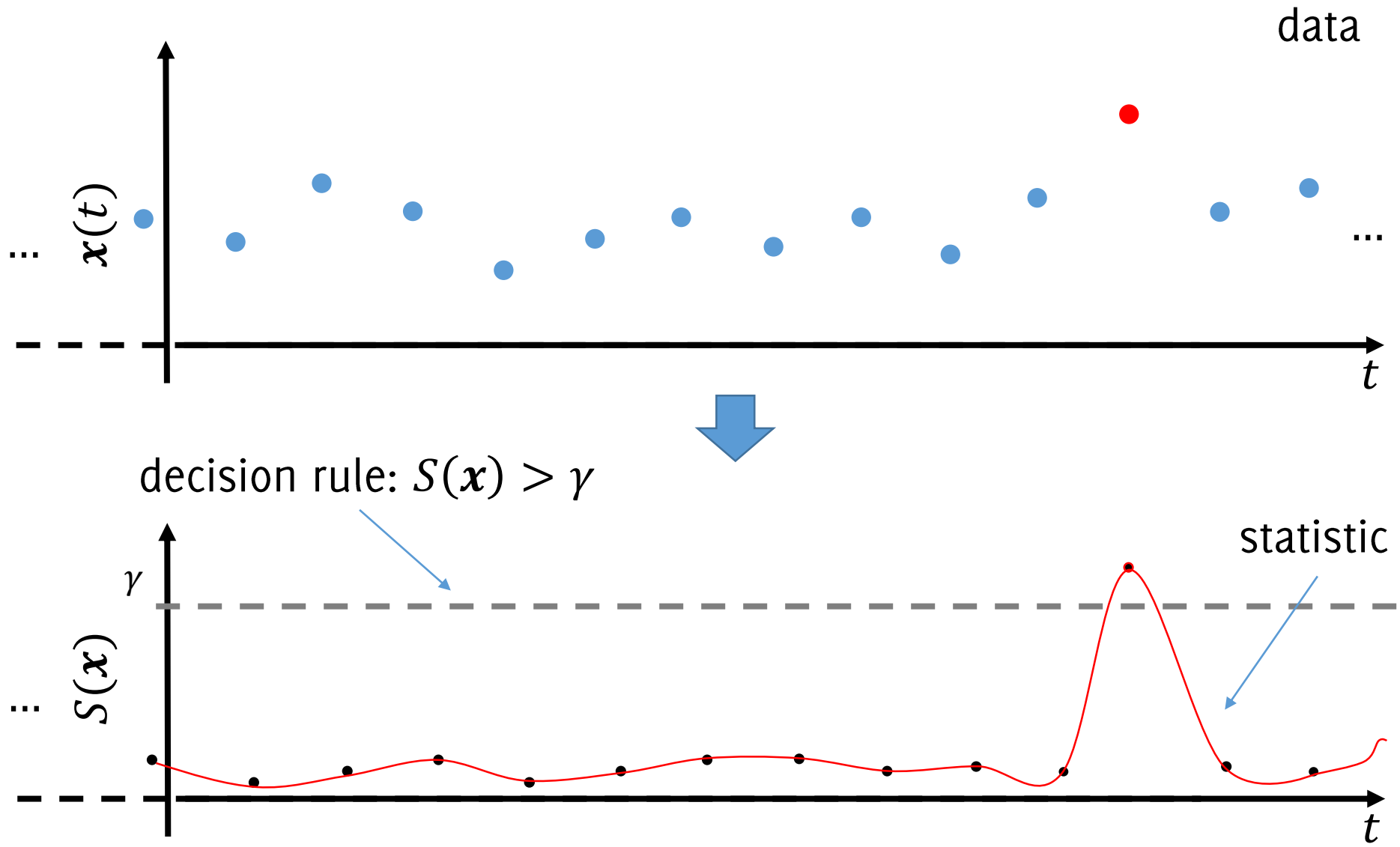
The Typical Anomaly Detection Solutions



The Typical Anomaly Detection Solutions



The Typical Anomaly Detection Solutions



Anomaly detection approaches

...when ϕ_0 and ϕ_1 are unknown

Anomaly detection when ϕ_0 and ϕ_1 are unknown

Most often, **only a training set** TR is **provided:**

There are three scenarios:

- **Supervised:** Both normal and anomalous training data are provided in TR.
- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.

$x \sim \phi_0$ $TR = \{x \sim \phi_0\}$ $x_i \sim \phi_1 \nexists \phi_0$

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

Anomaly detection when $\boldsymbol{\phi_0}$ and $\boldsymbol{\phi_1}$ are unknown

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Supervised:** Both normal and anomalous training data are provided in TR.
- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.

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

Anomaly detection when $\boldsymbol{\phi_0}$ and $\boldsymbol{\phi_1}$ are unknown

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Supervised:** Both normal and anomalous training data are provided in TR.
- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.

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

Practical Operating conditions

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

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

There are many reasons to opt for a semi-supervised / unsupervised approach

- **Normal data are easy to gather**. A training set of normal signals denoted as TR is provided
- **Anomalous** / changed data are **difficult to collect**
- Training examples in TR might not be **representative of all the possible anomalies** / changes that can occur
- **In some cases** TR is not labeled, but it is reasonable to assume that **normal data are the vast majority**

V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.
Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L. *"A review of novelty detection"* Signal Processing, 99, 215-249 (2014)

Semi-supervised Anomaly-Detection Methods

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

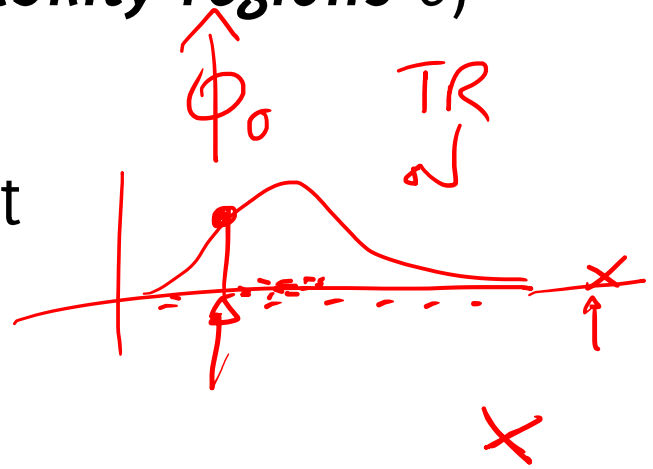
$$TR = \{x(t), t < t_0, x \sim \phi_0\}$$

Moreover… all in all… it is sometimes **safer** to **detect any data departing from** the **normal conditions**

Semi-supervised anomaly-detection methods are also referred to as **novelty-detection methods**

Density-based methods

Density-Based Methods: ***Normal*** *data occur in* ***high probability regions*** *of a stochastic model, while* ***anomalies*** *occur in the* ***low probability regions*** *of the model*



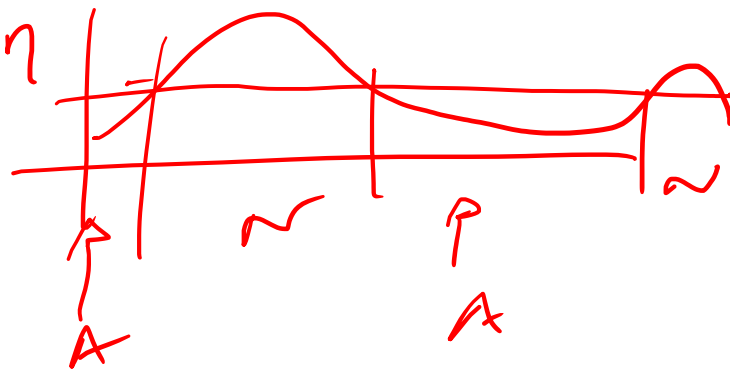
During training: $\hat{\phi}_0$ can be **estimated** from the training set

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

- parametric models (e.g., Gaussian mixture models)
- nonparametric models (e.g. KDE, histograms)

During testing:

- Anomalies are detected as data yielding $\hat{\phi}_0(\boldsymbol{x}) < \eta$



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

Density-based methods

Advantages:

- $\hat{\phi}_0(\boldsymbol{x})$ indicates how safe a detection is (like a p-value)
- If the density estimation process is robust to outliers, it is possible to tolerate few anomalous samples in TR
- **Histograms are simple to compute** in relatively small dimensions

Challenges:

- **It is challenging to fit models for high-dimensional data**
- Histograms traditionally suffer of **curse of dimensionality** when d increases
- Often the **1D histograms** of the marginals are monitored, **ignoring the correlations** among components

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

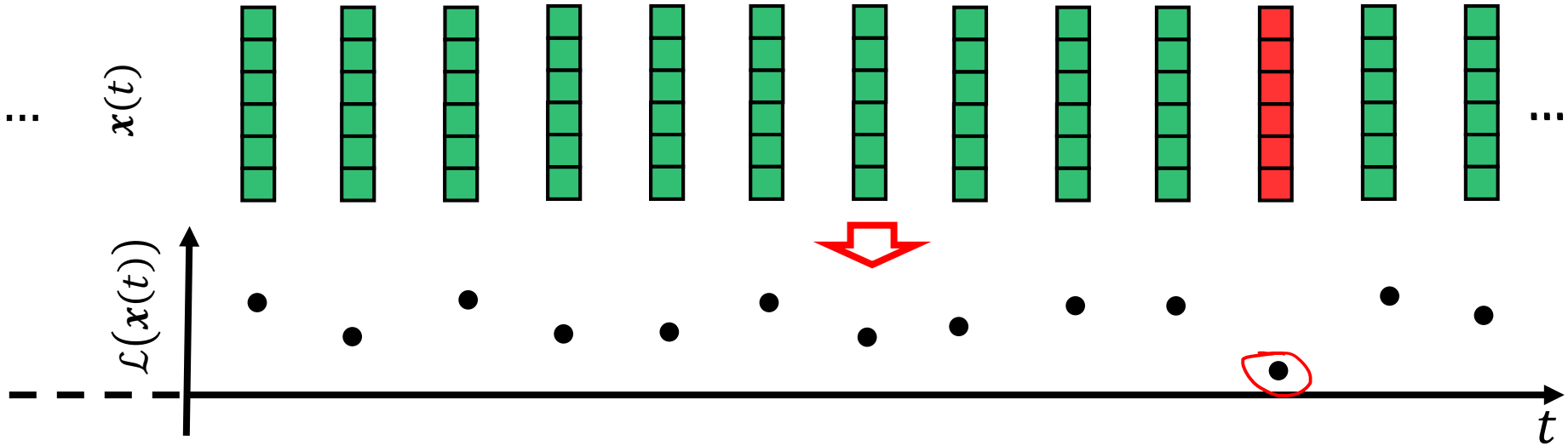
Density-based methods: Monitoring the Log-likelihood

Monitoring the log-likelihood of data w.r.t $\hat{\phi}_0$ allow to address anomaly-detection problem in multivariate data

1. During training, estimate $\hat{\phi}_0$ from TR
2. During testing, compute

$$\mathcal{L}\big(\boldsymbol{x}(t)\big) = \log(\hat{\phi}_0(\boldsymbol{x}(t)))$$

3. Monitor $\{\mathcal{L}\big(\boldsymbol{x}(t)\big),\ t = 1, \ldots\}$



Density-based methods: Monitoring the Log-likelihood

Monitoring the log-likelihood of data w.r.t $\hat{\phi}_0$ allow to address anomaly-detection problem in multivariate data

1. During training, estimate $\hat{\phi}_0$ from TR
2. During testing, compute

$$\mathcal{L}\big(\boldsymbol{x}(t)\big) = \log(\hat{\phi}_0(\boldsymbol{x}(t)))$$

3. Monitor $\big\{\mathcal{L}\big(\boldsymbol{x}(t)\big),\ t = 1, \ldots\big\}$

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

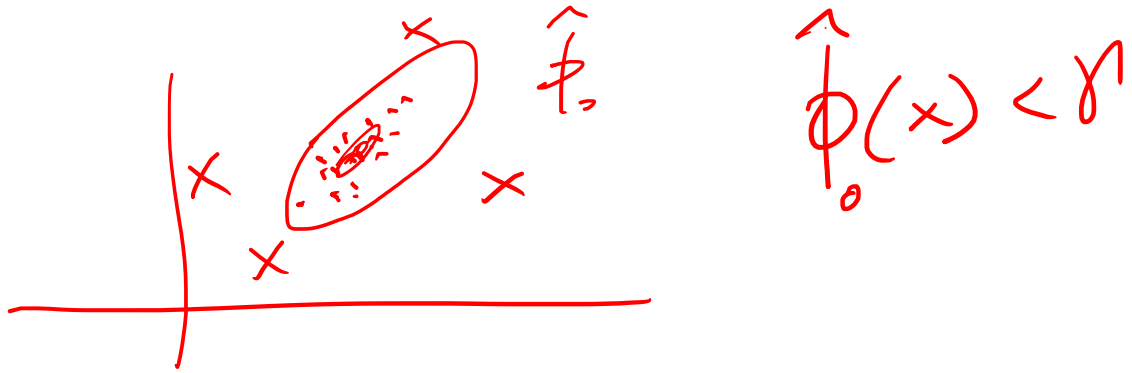
L. I. Kuncheva, “Change detection in streaming multivariate data using likelihood detectors," IEEE TKDE 2013.
X. Song, M. Wu, C. Jermaine, and S. Ranka, “Statistical change detection for multidimensional data” KDD, 2007.
J. H. Sullivan and W. H. Woodall, “Change-point detection of mean vector or covariance matrix shifts using multivariate individual observations," IIE transactions, vol. 32, no. 6, 2000.
C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss" IJCAI 2016,

Domain-based methods

Domain-based methods: ***Estimate a boundary around normal*** *data, rather than the density of normal data.*

A **drawback of density-estimation methods** is that they are meant to be accurate in high-density regions, while anomalies live in low-density ones.

One-Class SVM are domain-based methods defined by **the normal samples at the periphery of the distribution.**



Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., Platt, J. C. "*Support Vector Method for Novelty Detection*". In NIPS 1999 (Vol. 12, pp. 582-588).

Tax, D. M., Duin, R. P. "*Support vector domain description*". Pattern recognition letters, 20(11), 1191-1199 (1999)

Pimentel, M. A., Clifton, D. A., Clifton, L., Tarassenko, L. *"A review of novelty detection"* Signal Processing, 99, 215-249 (2014)

One-class svm (Schölkopf et al. 1999)

Idea: define boundaries by estimating a **binary function** f that **captures regions of the input space where density is higher**.

As in support vector methods, f is **defined in the feature space** F and **decision boundaries are defined by a few support vectors** (i.e., a few normal data).

Let $\psi(\boldsymbol{x})$ the feature associated to $\boldsymbol{x}$, f is defined as

$$f(\boldsymbol{x}) = \text{sign}(< w, \psi(\boldsymbol{x}) > -\rho)$$

Where the hyperplane parameters w, ρ are optimized to yield a **function that is positive on most training samples.** Thus in the feature space, normal points can be separated from the origin.

A linear separation in the feature space corresponds to a **variety of nonlinear boundaries in the space of** $\boldsymbol{x}$.

Schölkopf, B., Williamson, R. C., Smola, A. J., Shawe-Taylor, J., Platt, J. C. "*Support Vector Method for Novelty Detection*". In NIPS 1999 (Vol. 12, pp. 582-588).

One-class svm (Tax and Duin 1999)

Boundaries of normal region can be also defined by an **hypersphere that, in the feature space, encloses most of the normal data.**

Similar detection formulas hold, **measuring the distance in the feature space from the sphere center for each** $\psi(\boldsymbol{x})$ for $\boldsymbol{x} \in TR$.

The function is always defined by a few support vectors.

Remarks: In both one-class approaches, the amount of samples that falls within the margin (outliers) is controlled by regularization parameters.

This parameter regulates the number of outliers in the training set and the detector sensitivity.

Tax, D. M., Duin, R. P. "*Support vector domain description*". Pattern recognition letters, 20(11), 1191-1199 (1999)

Anomaly detection when $\boldsymbol{\phi_0}$ and $\boldsymbol{\phi_1}$ are unknown

Most often, **only a training set** TR is **provided**:

There are three scenarios:

- **Supervised:** Both normal and anomalous training data are provided in TR.
- **Semi-Supervised:** Only normal training data are provided, i.e. no anomalies in TR.
- **Unsupervised:** TR is provided without label.

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

Unsupervised anomaly-detection

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

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

Underlying assumption: ***Anomalies are rare*** *w.r.t. normal data* TR

One in principle could use:

- Density/Domain based methods that are **robust to outliers** can be applied in an unsupervised scenario
- Unsupervised methods can be improved whenever labels are available

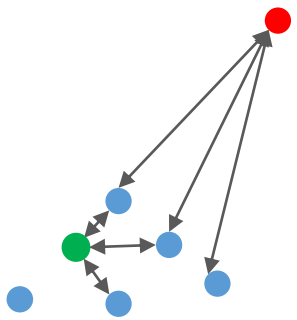
Distance-based methods

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors****.*

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its k **–nearest neighbor**



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

Zhao, M., Saligrama, V. *"Anomaly detection with score functions based on nearest neighbor graphs"*. NIPS 2009

A. Zimek, E. Schubert, H. Kriegel. *"A survey on unsupervised outlier detection in high-dimensional numerical data"* SADM 2012.

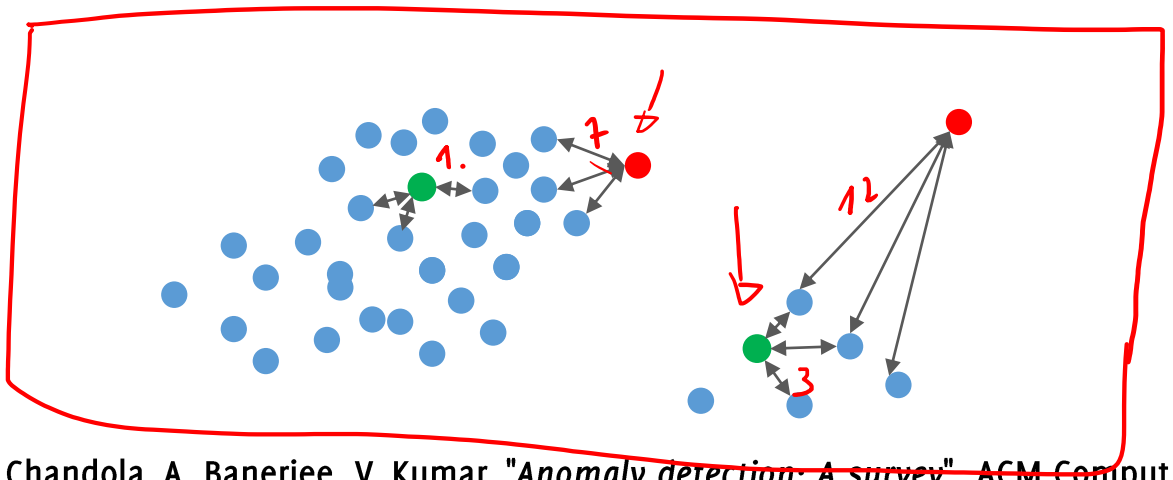
Distance-based methods

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors****.*

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its k **–nearest neighbor**
- the **above distance** considered **relatively to neighbors**



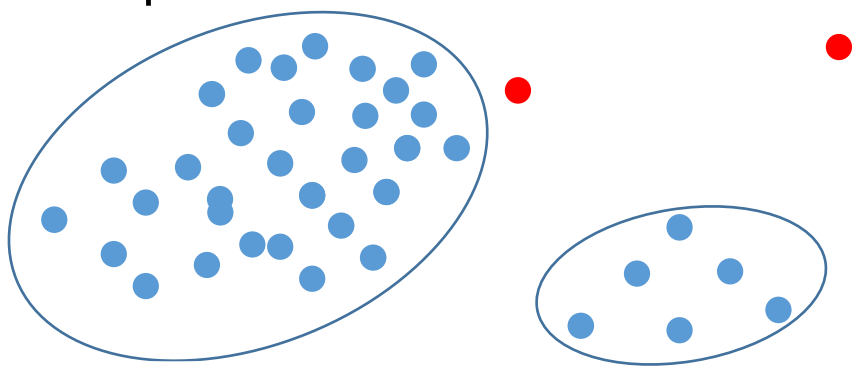
Distance-based methods

Distance-based methods: ***normal*** *data fall in* ***dense neighborhoods****, while* ***anomalies*** *are* ***far from their closest neighbors****.*

A critical aspect is the **choice of the similarity measure** to use.

Anomalies are detected by **monitoring**:

- **distance** between each data and its k **–nearest neighbor**
- the **above distance** considered **relatively to neighbors** LOF
- whether they do not belong to **clusters**, or are at the cluster periphery, or belong to small and sparse clusters

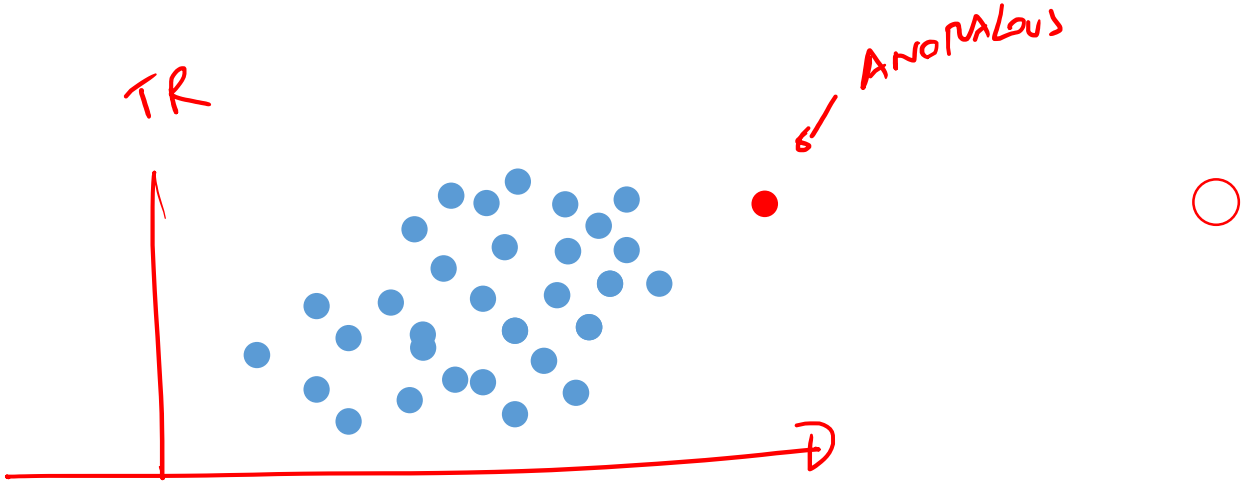


V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

IFOR: Isolation Forest

Builds upon the rationale that **«anomalies are easier to separate from the rest of normal data»**

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure

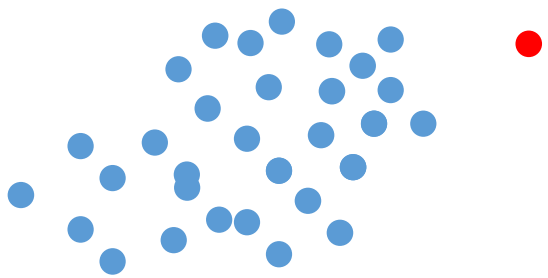


Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

Builds upon the rationale that **«anomalies are easier to separate from the rest of normal data»**

This idea is implemented very efficiently through **a forest of binary trees** that are constructed via an iterative procedure



Randomly choose

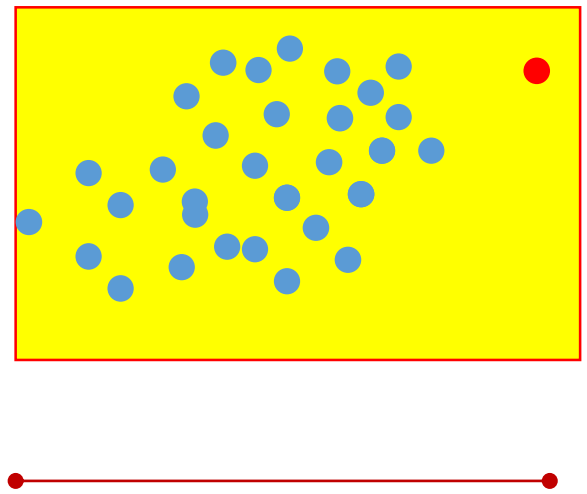
1. a component x_i

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure



Randomly choose

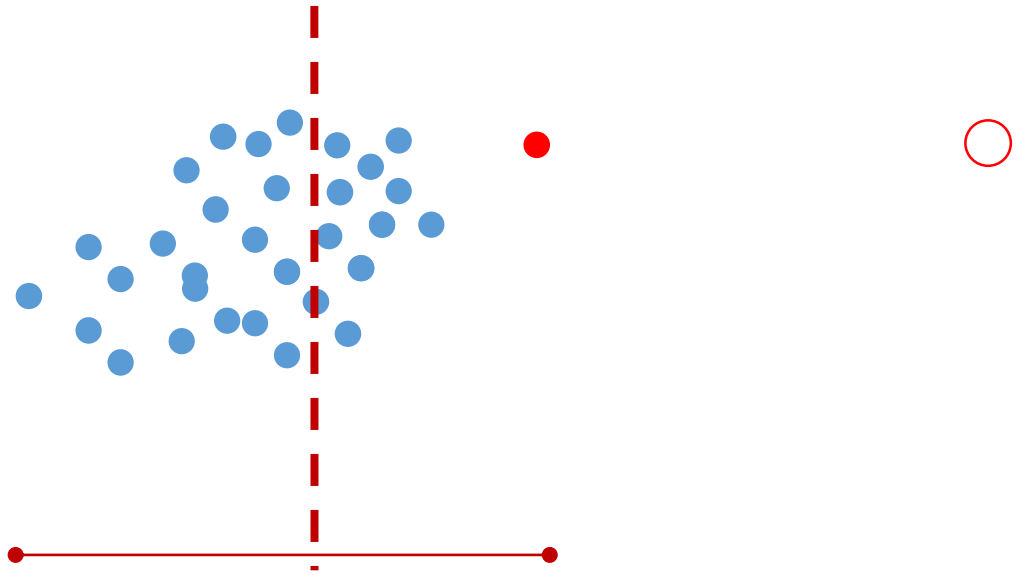
1. a component x_i

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure



Randomly choose

1. a component x_i
2. a value in the range of projections of TR over the i-th component

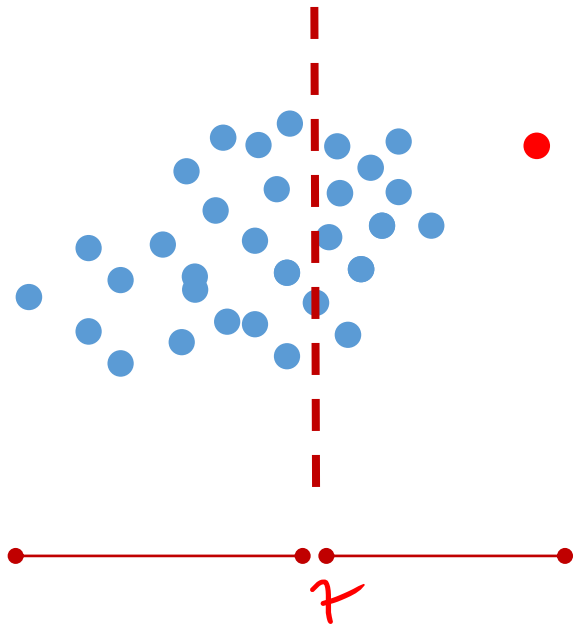
This yields a splitting

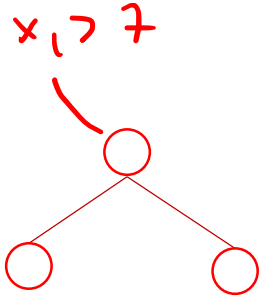
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





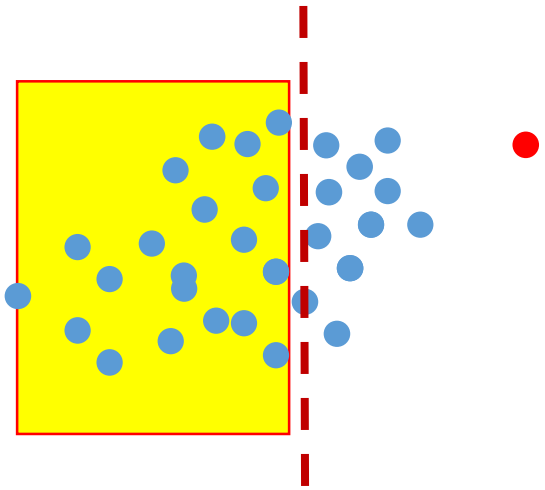
This yields a splitting criteria

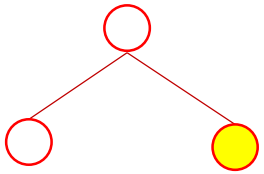
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





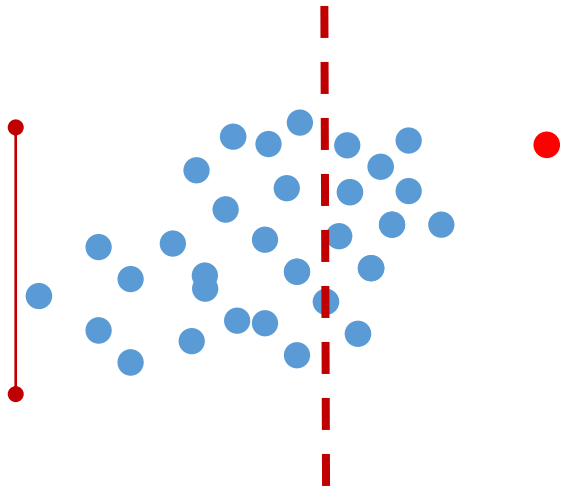
Repeat the procedure on each node:

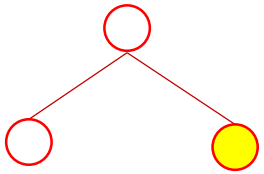
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





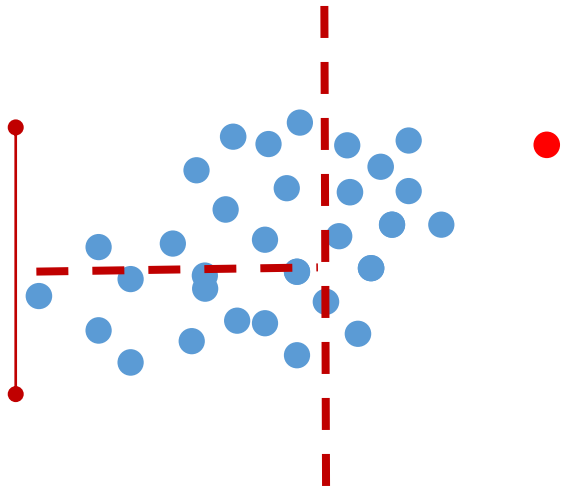
Repeat the procedure on each node: Randomly select a component

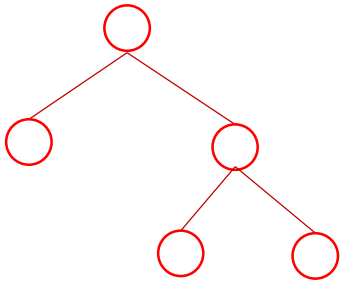
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





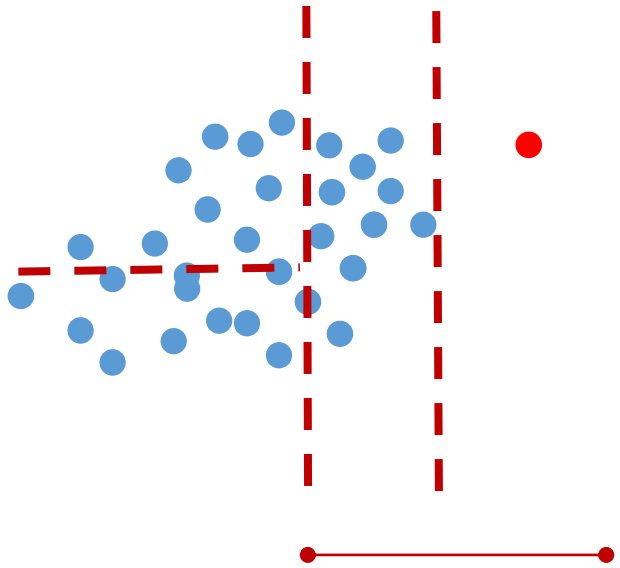
Repeat the procedure on each node: Randomly select a component and a cut point

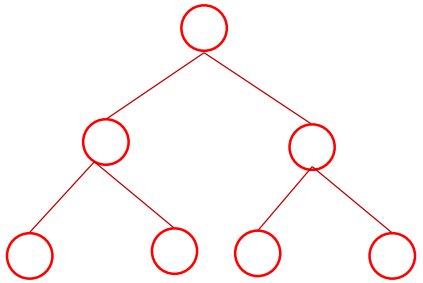
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





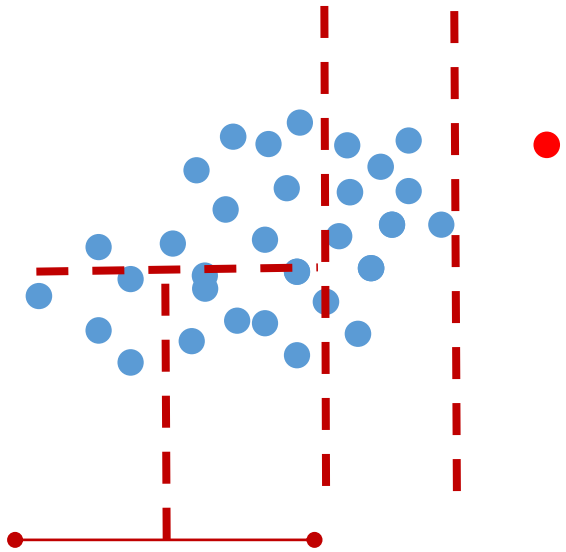
Randomly choose a component and a value within the range and define a splitting criteria

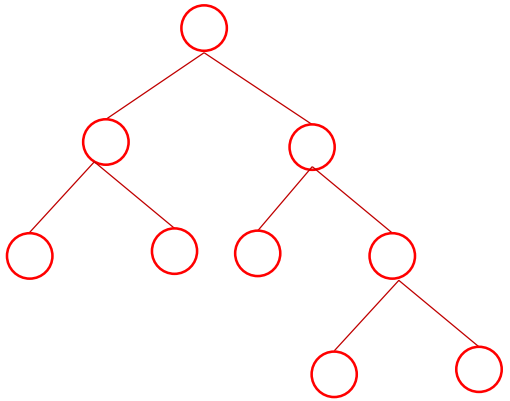
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





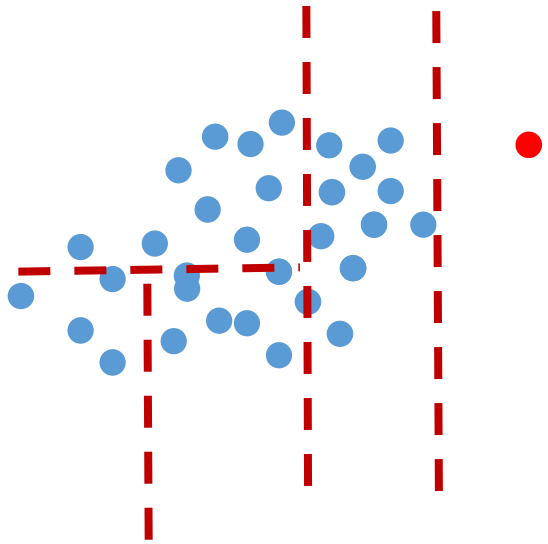
Repeat the procedure on the nodes:
Randomly select a component and a cut point

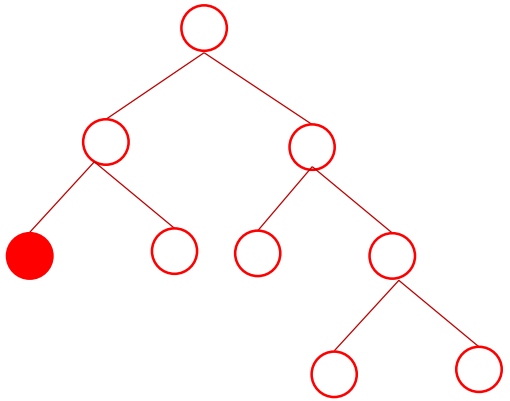
Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

Builds upon the rationale that «anomalies are easier to separate from the rest of normal data»

This idea is implemented very efficiently through a forest of binary trees that are constructed via an iterative procedure





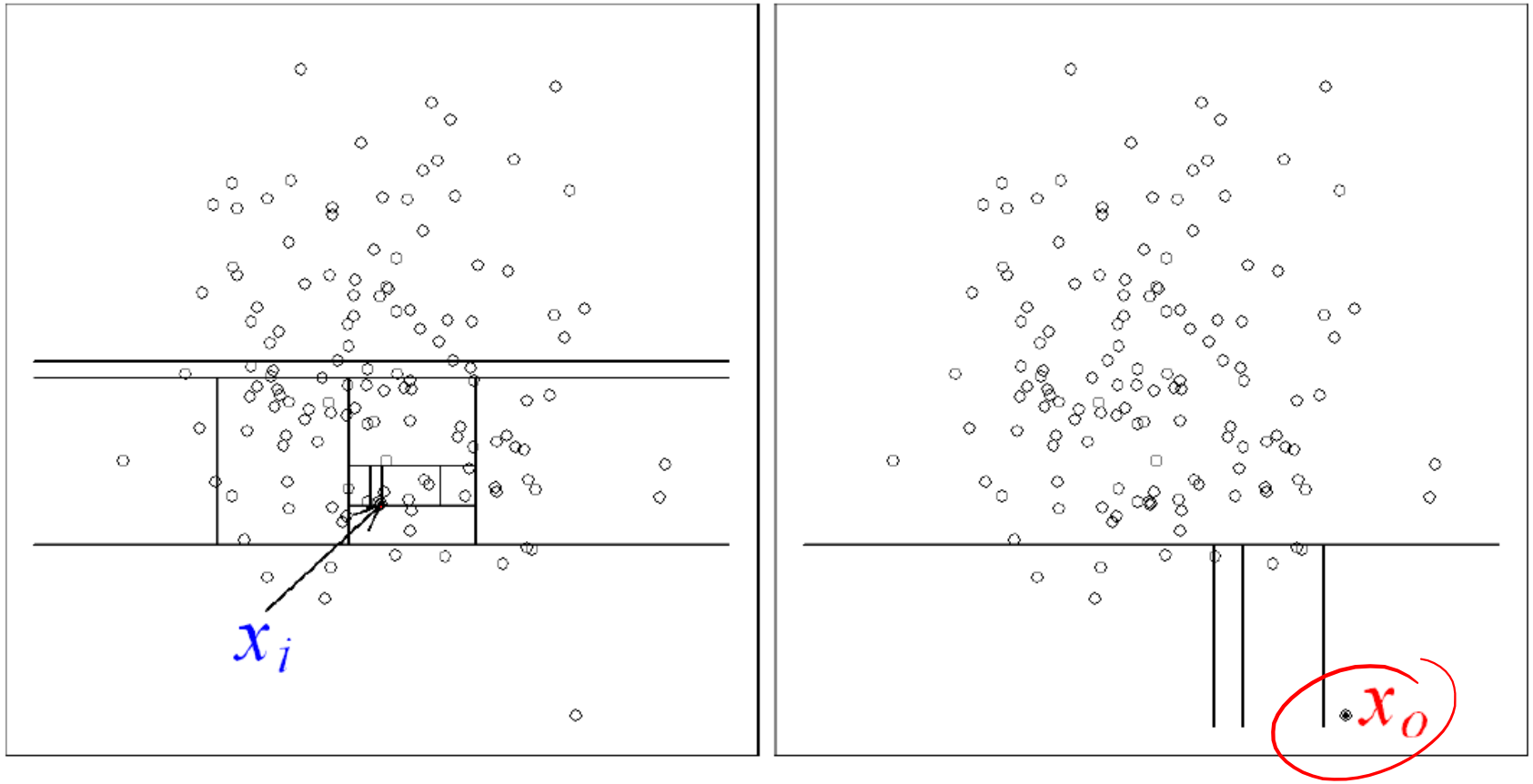
Anomalies lies in leaves close to the root.

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

An anomalous point (x_0) can be easily isolated

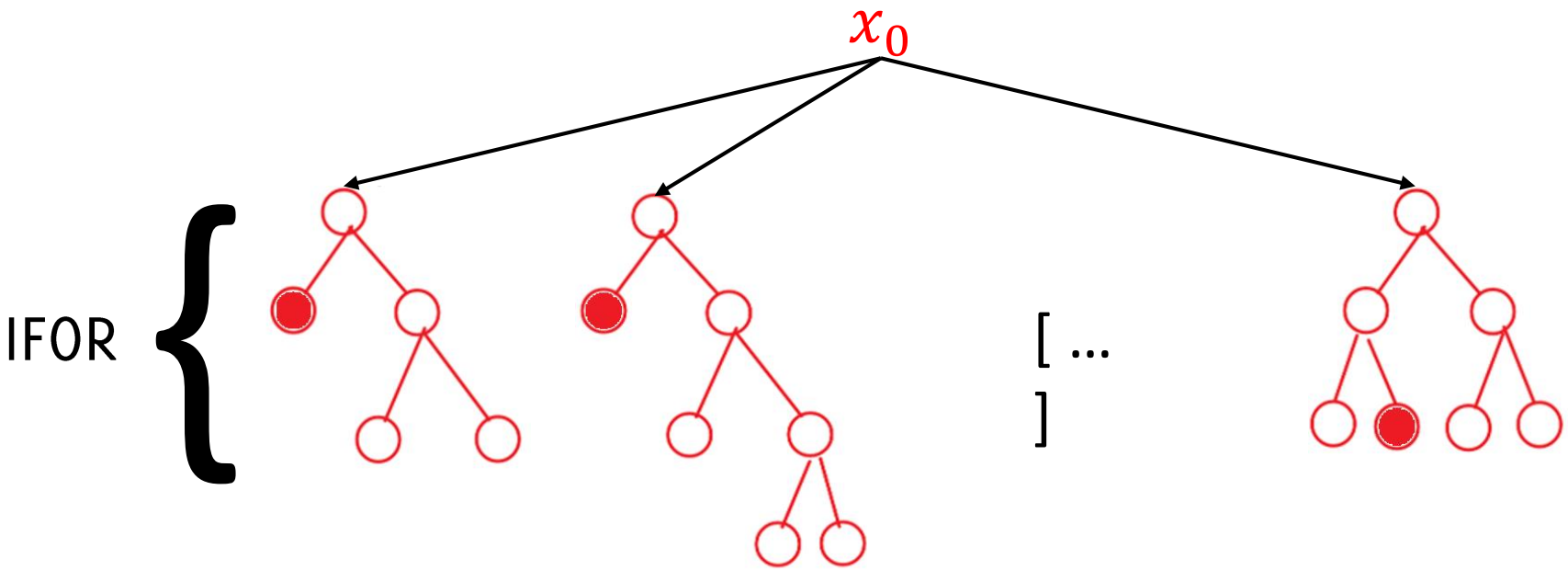
Genuine points (x_i) are instead difficult to isolate.



Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

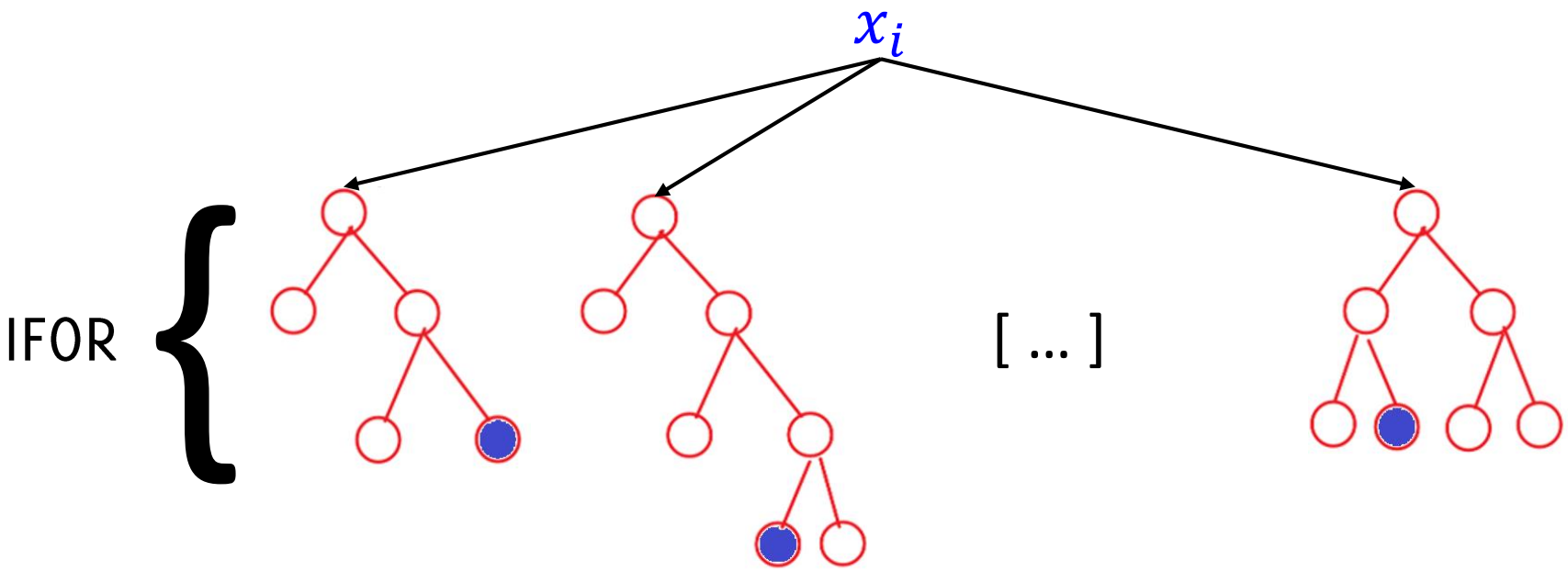
Anomalies



Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: Isolation Forest

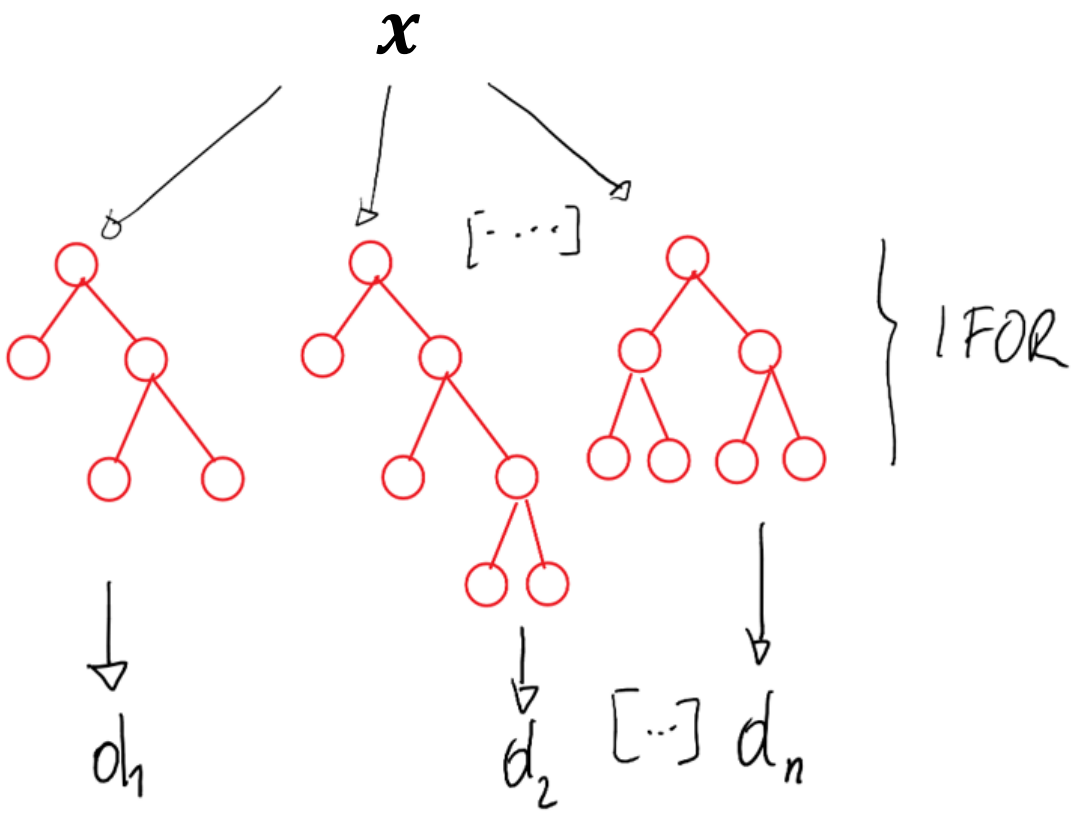
Normal data



Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: testing

Compute $E\big(h(\boldsymbol{x})\big)$, the **average path length among all the trees in the forest**, of a test sample $\boldsymbol{x}$



Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

IFOR: testing

A test sample is identified as **anomalous** when:

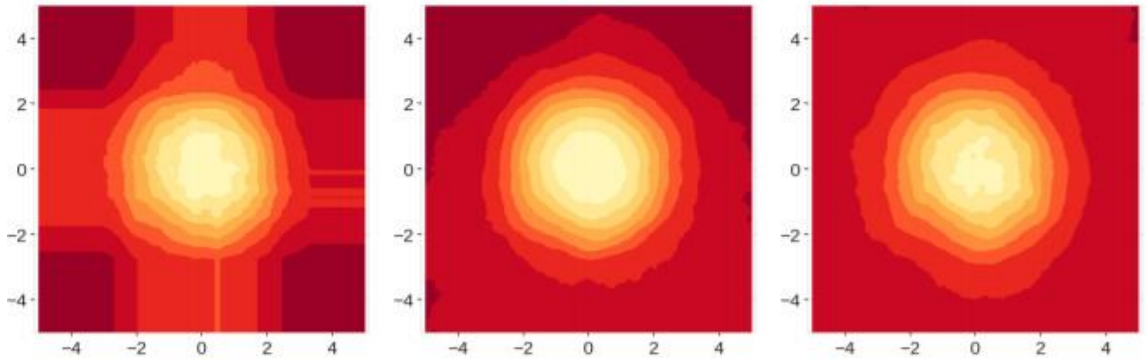
$$\mathcal{A}(\boldsymbol{x}) = 2^{-\frac{E(h(\boldsymbol{x}))}{c(n)}} > \gamma$$

- n: number of samples in TR
- $c(n)$: average path length of unsuccessful search in a binary tree

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*

Isolation Forest: testing

Several extensions including **EIF** (Extendend Isolation Forest) modify the splitting criteria to yield anomaly scores map that better conform to normal data



(a) Standard IF (b) Rotated IF (c) Extended IF

Fei Tony Liu, Kai Ming Ting and Zhi-Hua Zhou, *Isolation Forest, ICDM 2008*
S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest", TKDE, 2019.

Out of the «Random Variable World»

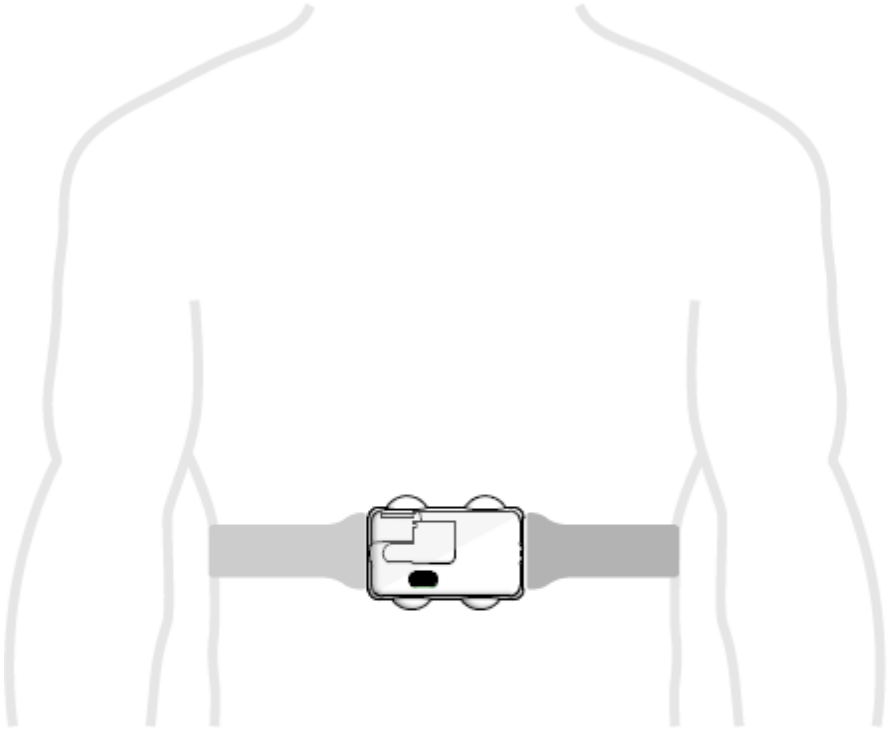
Anomaly Detection Methods for Signals and Images

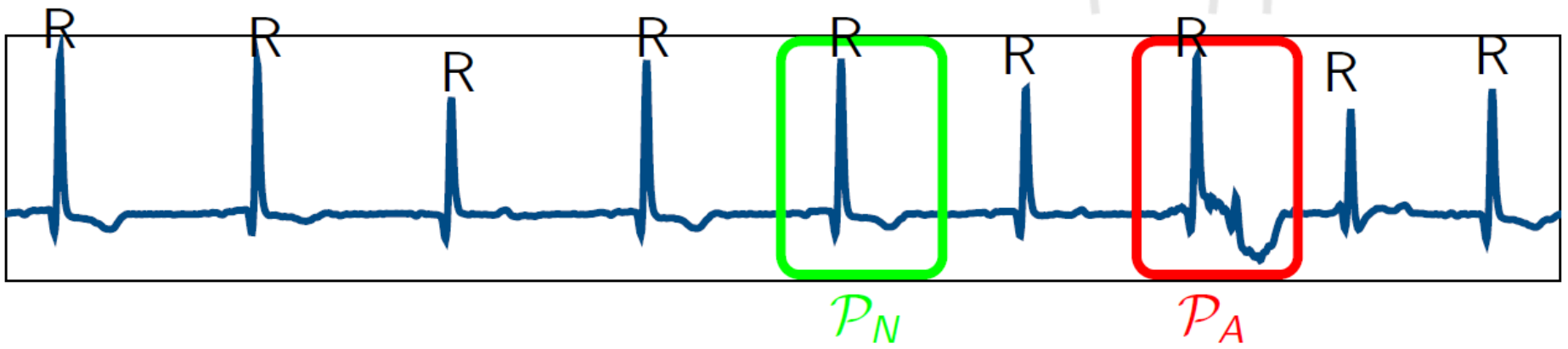
Boracchi, Trovò

… An Anomaly-Detection Problem

Health monitoring / wearable devices:

Automatically analyze EGC tracings to detect arrhythmias or incorrect device positioning

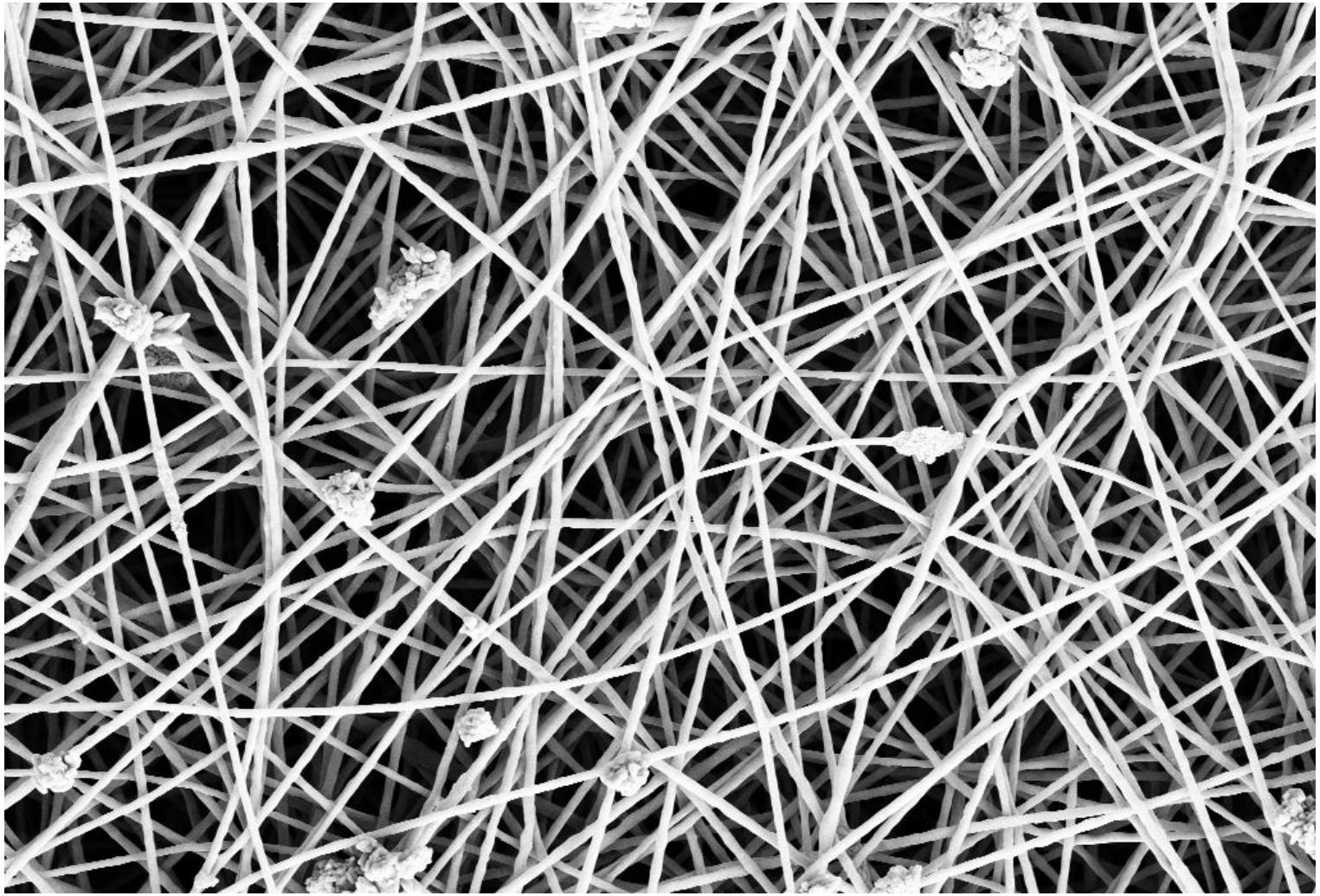




D. Carrera, B. Rossi, D. Zambon, P. Fragneto, and G. Boracchi "*ECG Monitoring in Wearable Devices by Sparse Models*" in Proceedings of ECML-PKDD 2016, 16 pages

… An Anomaly-Detection Problem

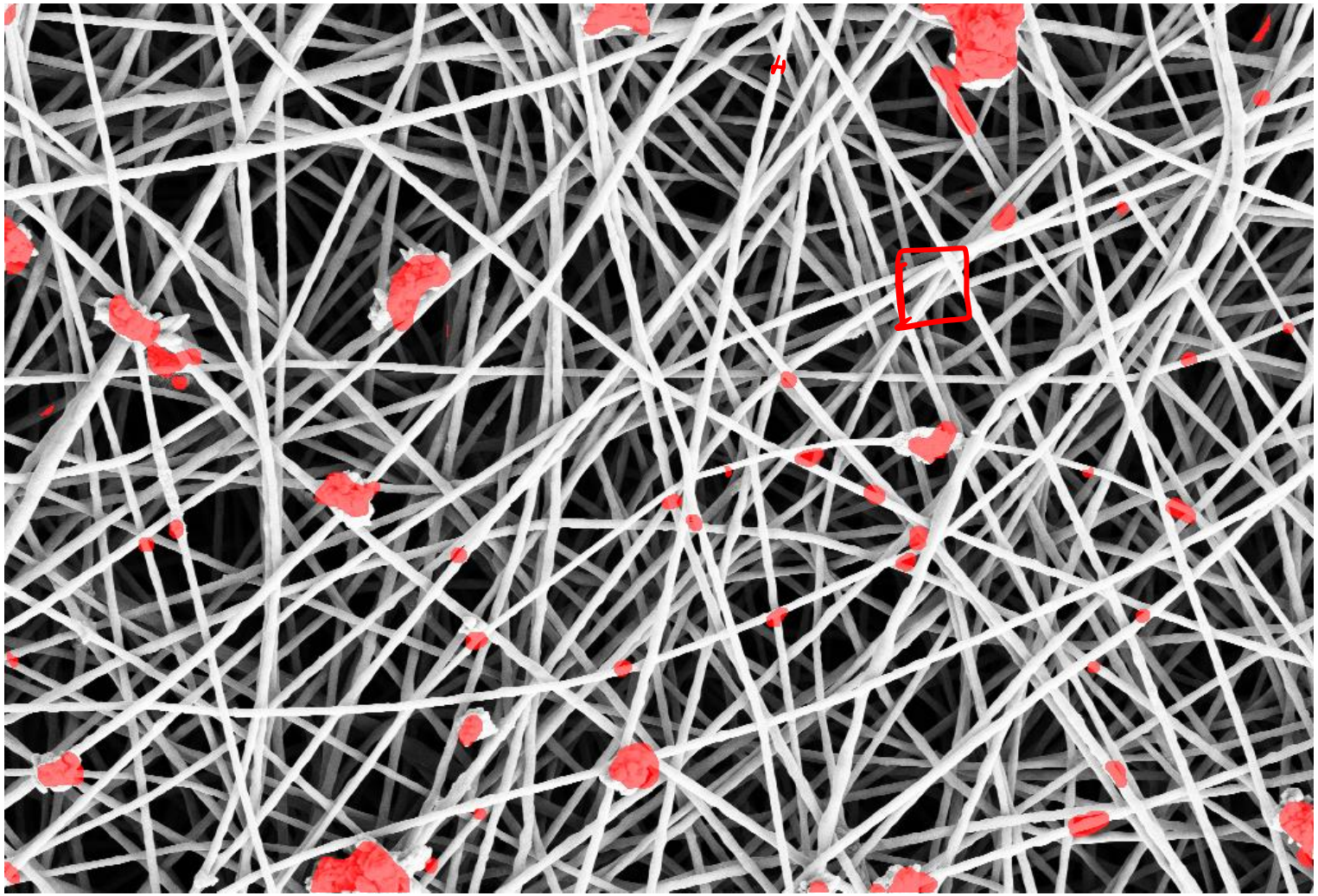
Quality Inspection Systems: monitoring the nanofiber production



Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

… An Anomaly-Detection Problem

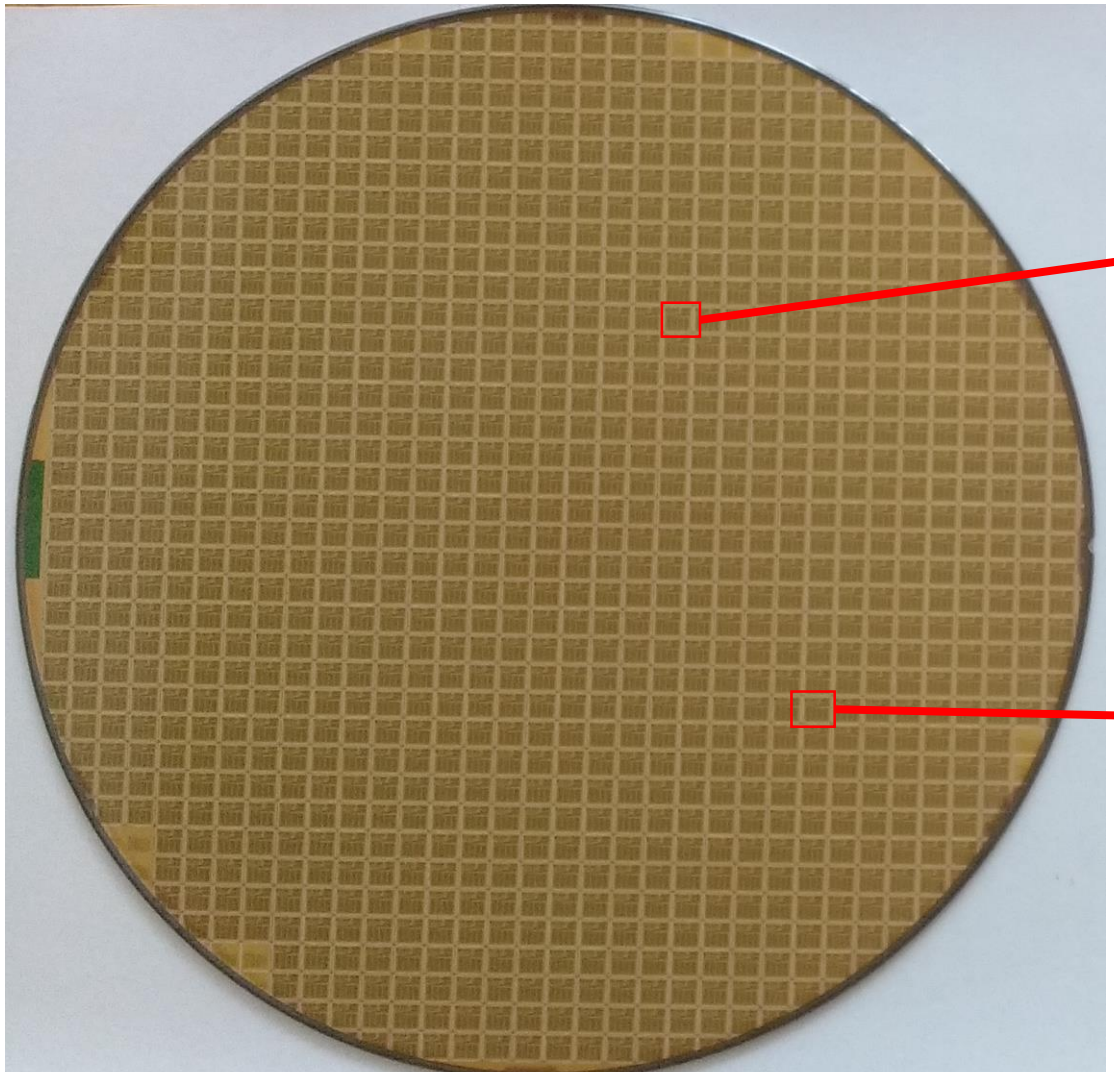
Quality Inspection Systems: monitoring the nanofiber production

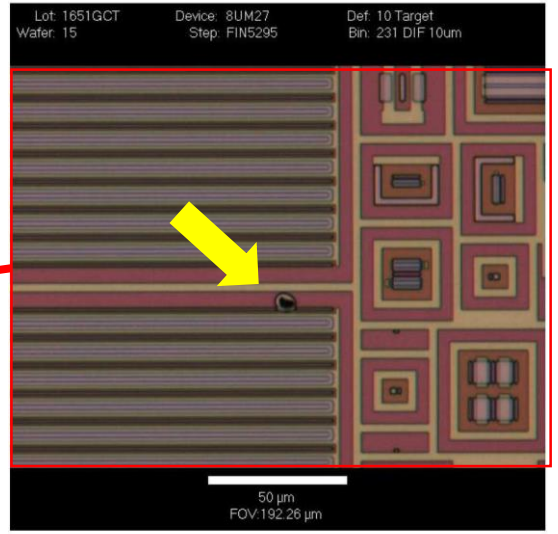


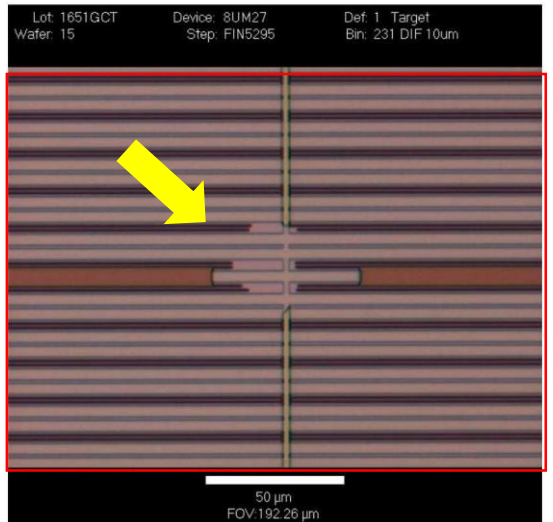
Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

… An Anomaly-Detection Problem

Detection of anomalies in chip production



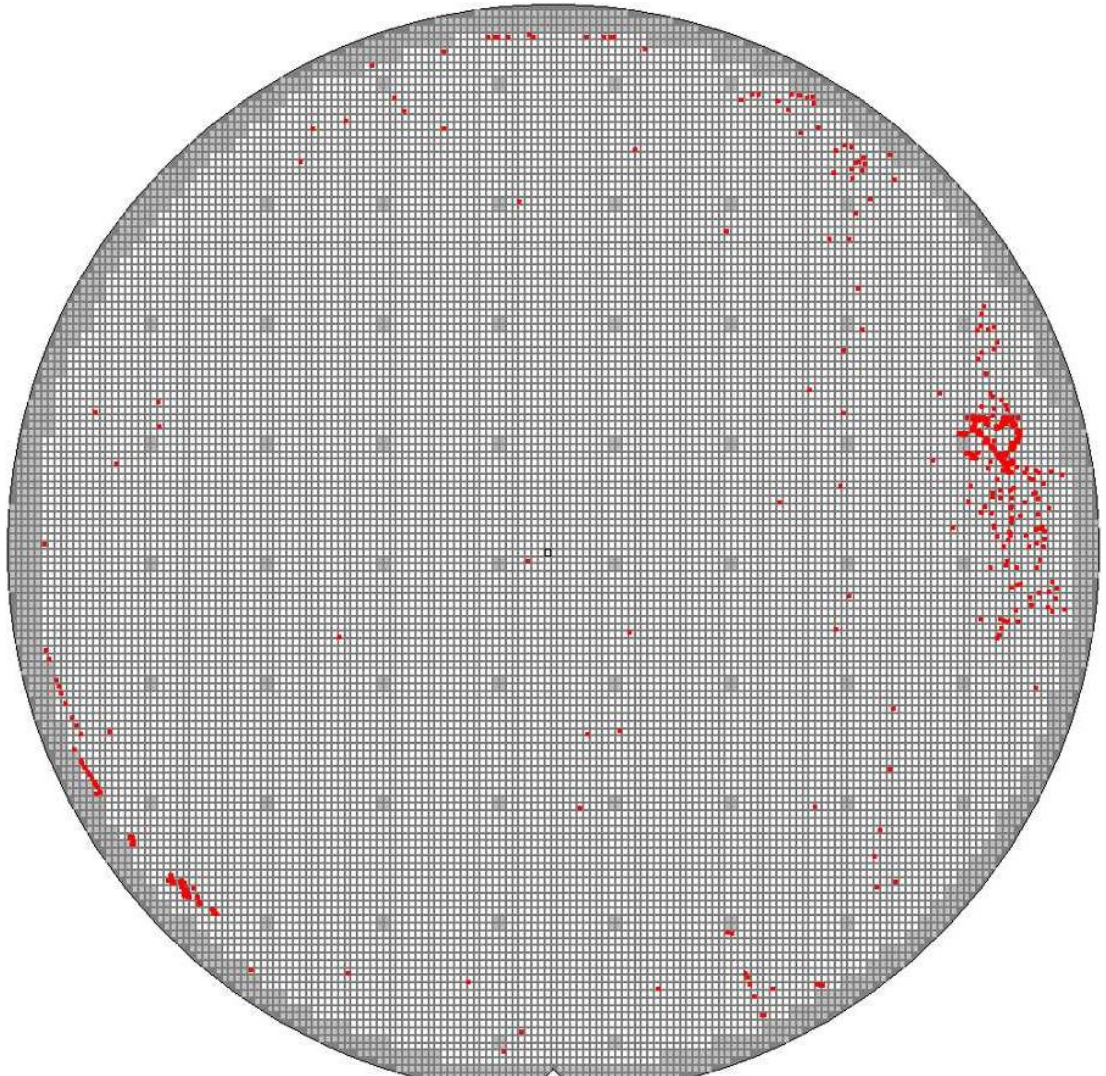




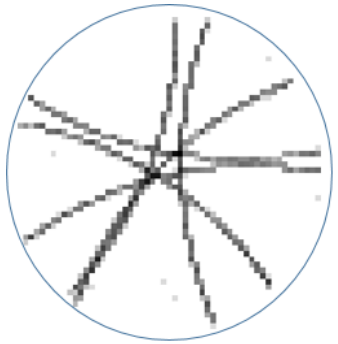
Boracchi, Trovò

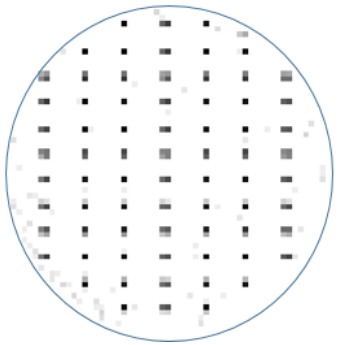
… An Anomaly-Detection Problem

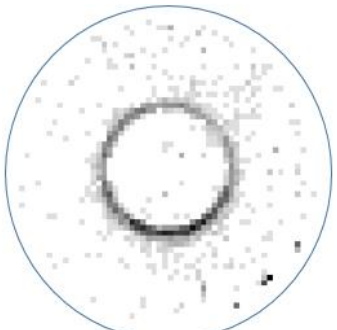
Detect **anomalous patterns** in the layout of defective chips, i.,e in the **wafer defect map.**

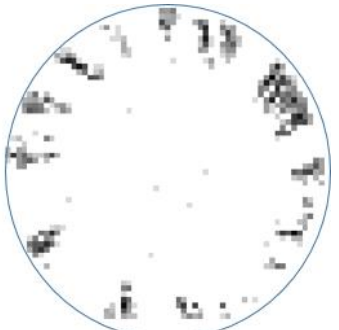


These might indicate faults, problems or malfunctioning in the chip production.



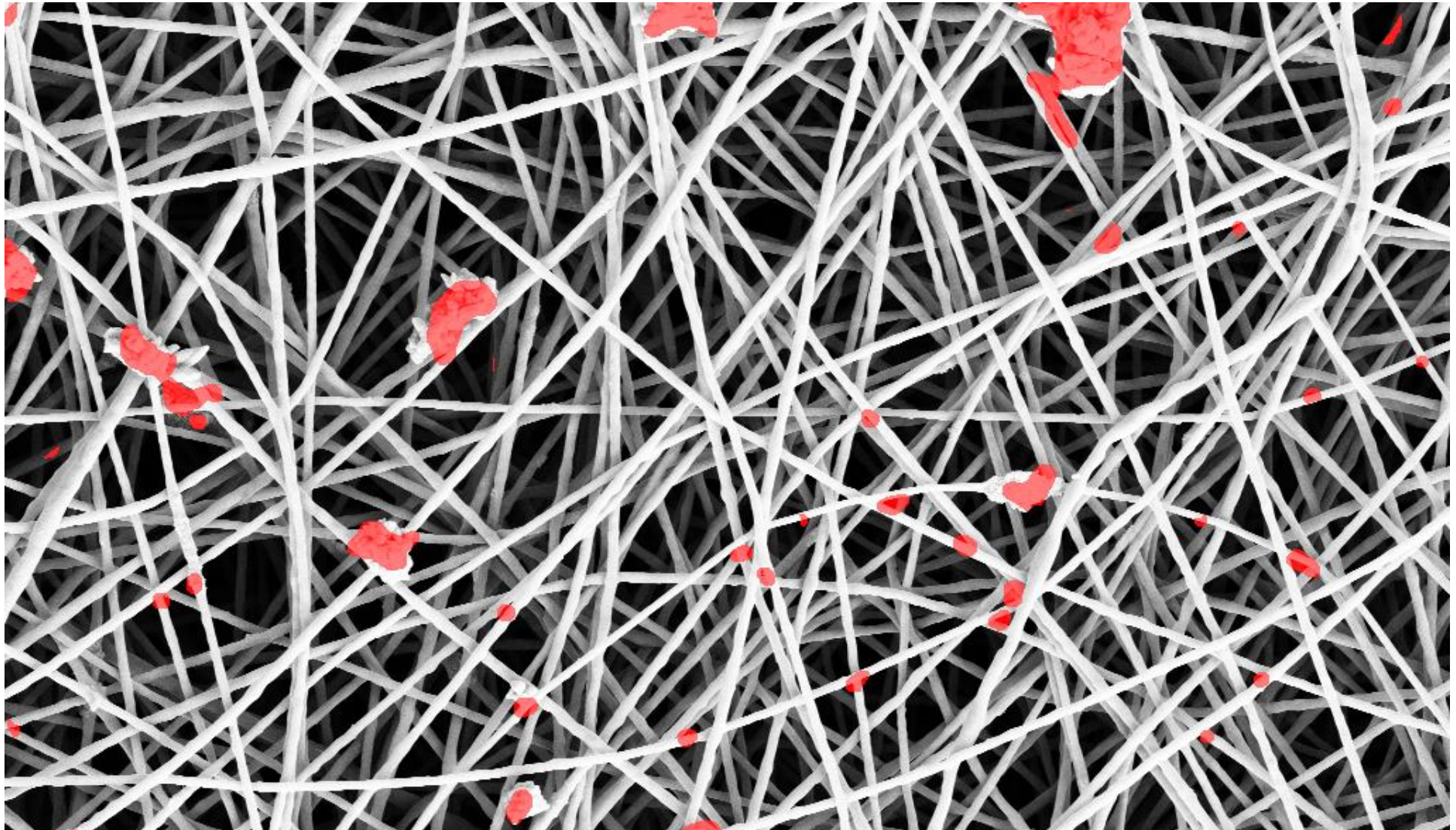






OUR Running example

Goal: Automatically measure area covered by defects



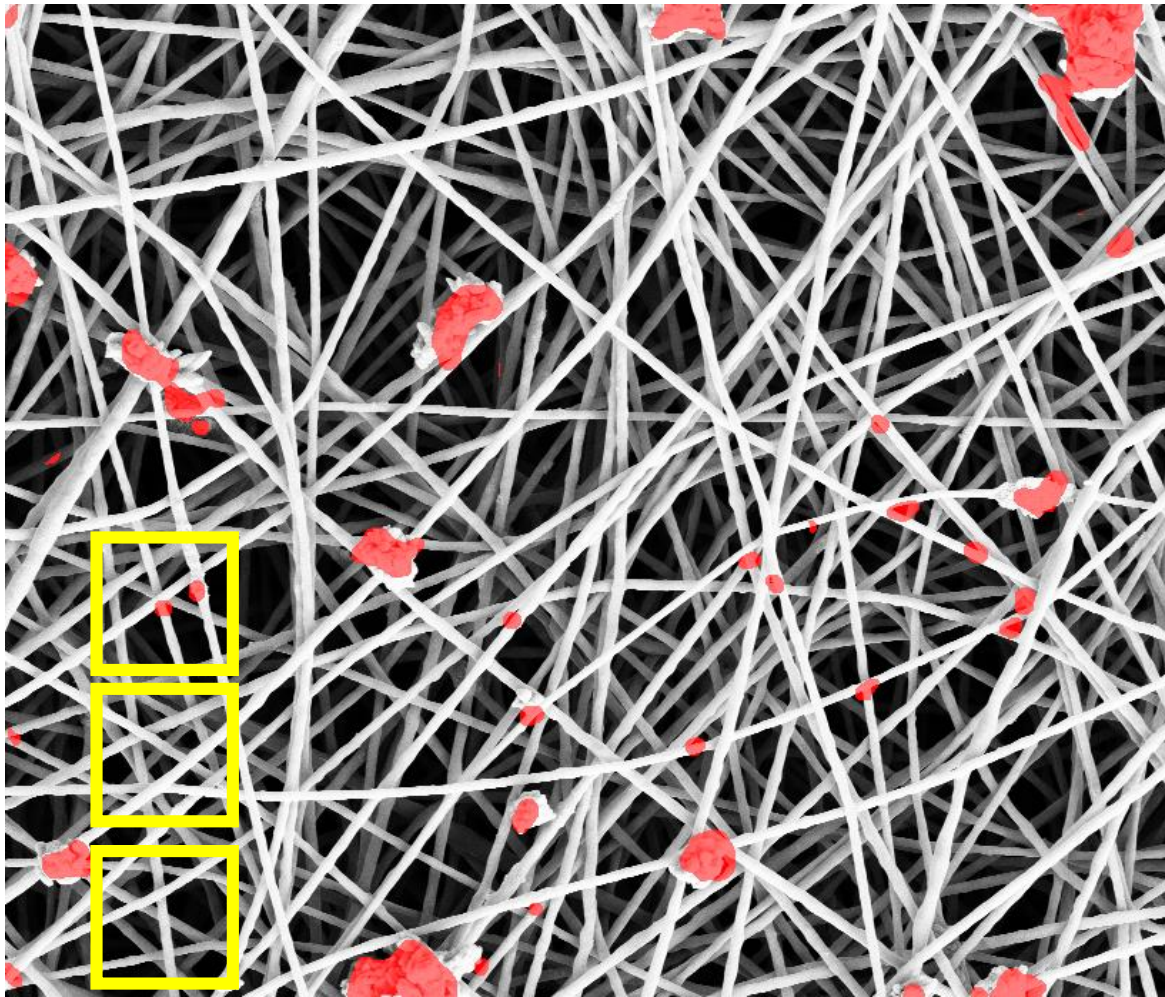
Boracchi, Trovò

Anomaly Detection in Images

The goal not determining whether the whole image is normal or anomalous, but **locate/segment possible anomalies**

Therefore, it is convenient to

1. **Analyze the image patch-wise**
2. Isolate regions containing patches that are detected as as anomalies



Can we pursue approaches meant for random variables on image patches?

Density-based approach on image patches

A density-based approach to AD would be:

Training

i. Split the normal image in patches $\boldsymbol{s}$

ii. Fit a statistical model $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ describing normal patches.

This model is rarely accurate on natural images. Small patches (e.g. 2×2 or 5×5) are typically preferred

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery.* IEEE Transactions on Geoscience and Remote sensing

X Xie, M Mirmehdi *"Texture exemplars for defect detection on random textures"* – ICPR 2005

Density-based approach on image patches

A density-based approach to AD would be:

Training

i. Split the normal image in patches $\boldsymbol{s}$

ii. Fit a statistical model $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ describing normal patches.

In some cases (textures) a Gaussian Mixture was used as a more general model

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery.* IEEE Transactions on Geoscience and Remote sensing

X Xie, M Mirmehdi *"Texture exemplars for defect detection on random textures"* – ICPR 2005

Density-based approach on image patches

A density-based approach to AD would be:

Training

i. Split the normal image in patches $\boldsymbol{s}$

ii. Fit a statistical model $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ describing normal patches.

Random selection procedures can be employed to minimize the risk of including outliers

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery.* IEEE Transactions on Geoscience and Remote sensing

Density-based approach on image patches

A density-based approach to AD would be:

Training

i. Split the normal image in patches $\boldsymbol{s}$

ii. Fit a statistical model $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ describing normal patches.

Testing

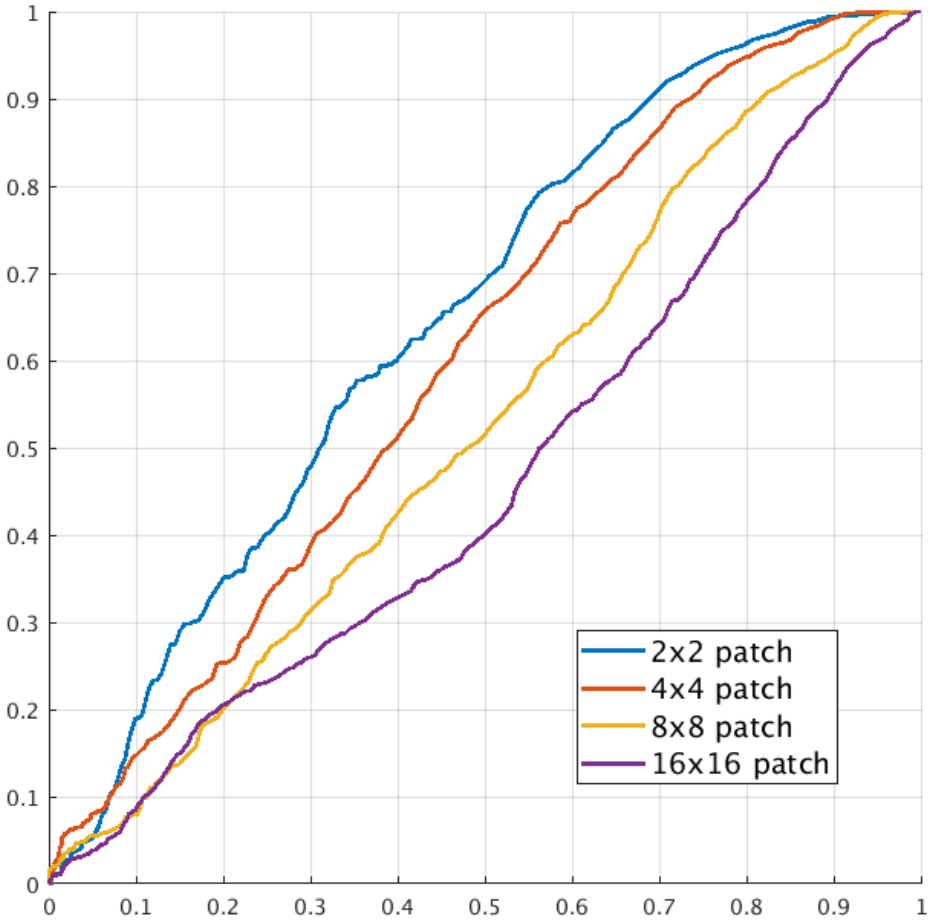
i. Split the test image in patches

ii. Compute $\hat{\phi}_0(\boldsymbol{s})$ the likelihood of each test patch $\boldsymbol{s}$

iii. Detect anomalies by thresholding the likelihood

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery.* IEEE Transactions on Geoscience and Remote sensing

The limitations of the Random variable model



This model is rarely accurate on natural images. Small patches (e.g. 2×2 or 5×5) are typically preferred

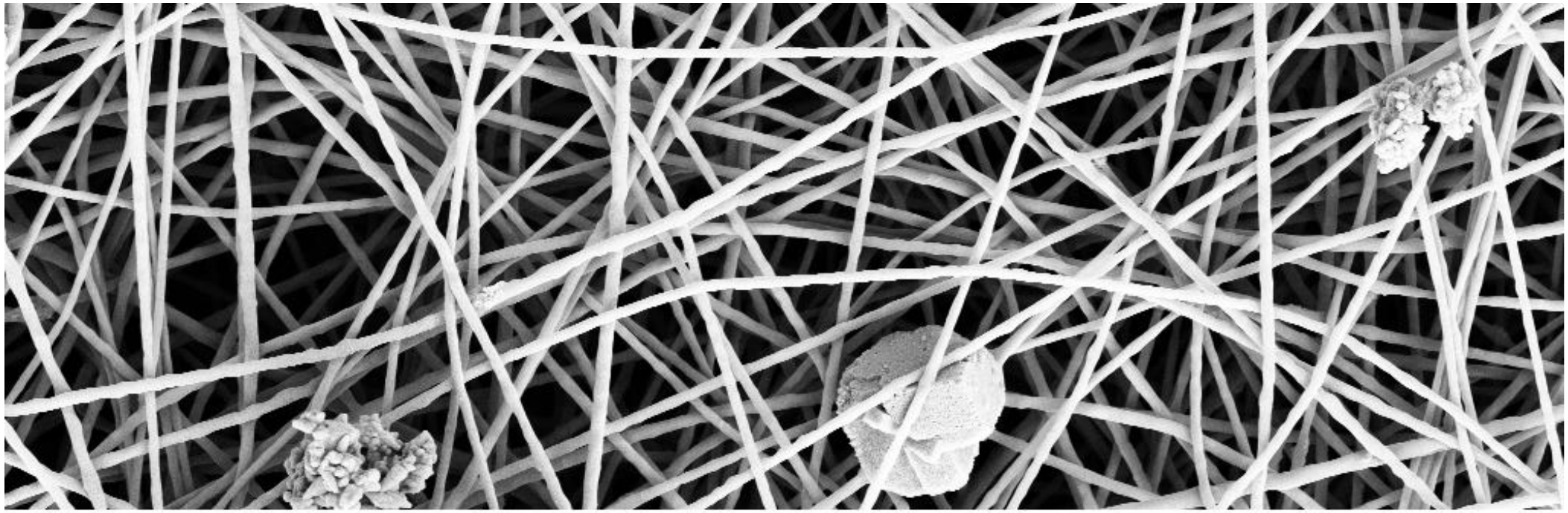
The model becomes even more unfit as the patch size increases

Du, B., Zhang, L.: *Random-selection-based anomaly detector for hyperspectral imagery*. IEEE Transactions on Geoscience and Remote sensing

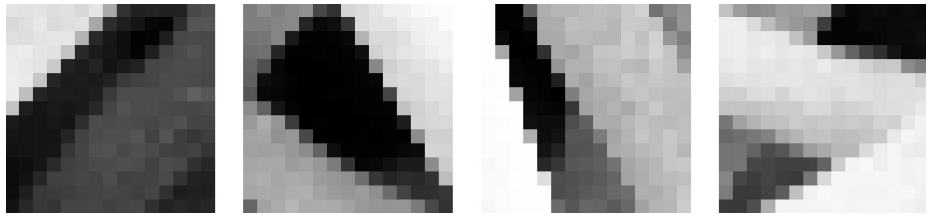
X Xie, M Mirmehdi *"Texture exemplars for defect detection on random textures"* – ICPR 2005

The limitations of the Random variable model

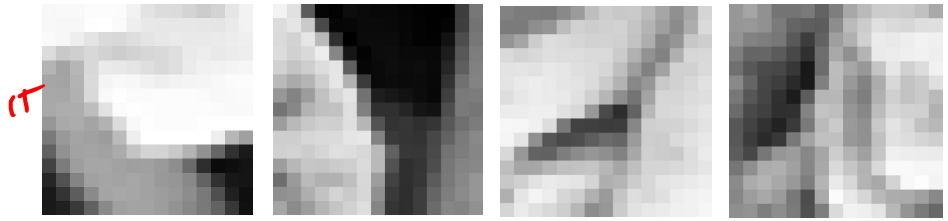
In many anomaly-detection problems in imaging, **normal regions exhibit peculiar structures and spatial correlation**



Normal regions



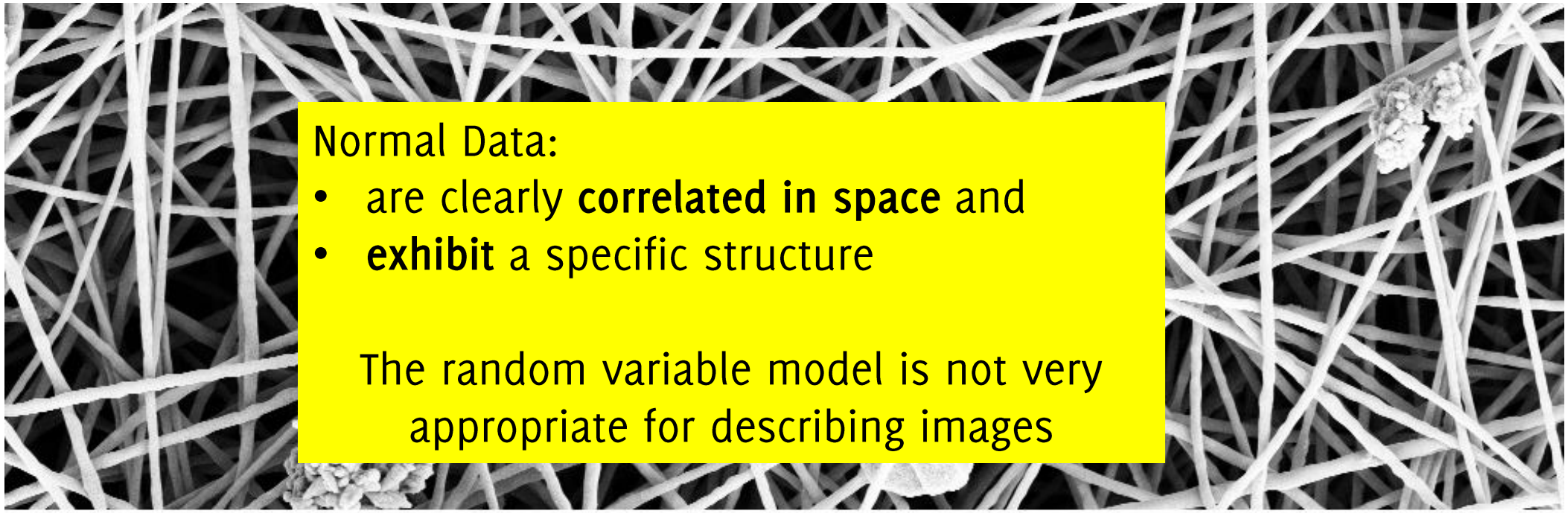
Anomalous regions



Boracchi, Trovò

The limitations of the Random variable model

In many anomaly-detection problems in imaging, **normal regions exhibit peculiar structures and spatial correlation**

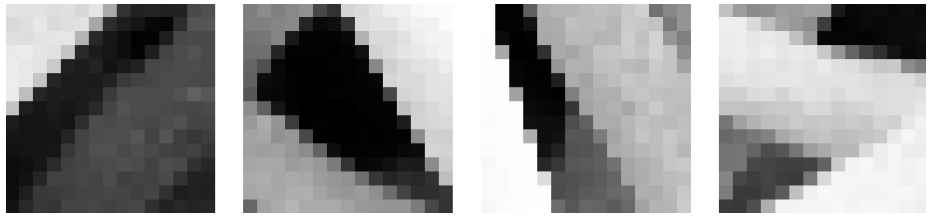


Normal Data:

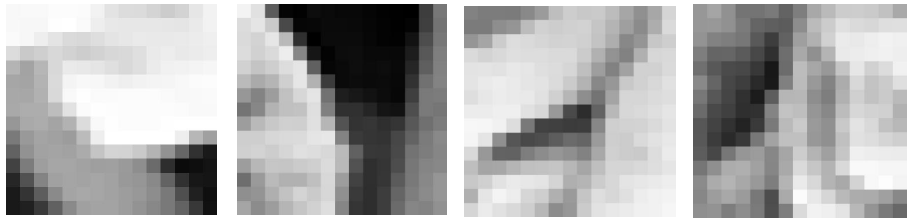
- are clearly **correlated in space** and
- **exhibit** a specific structure

The random variable model is not very appropriate for describing images

Normal regions



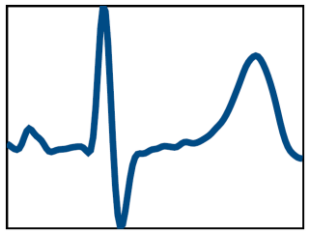
Anomalous regions

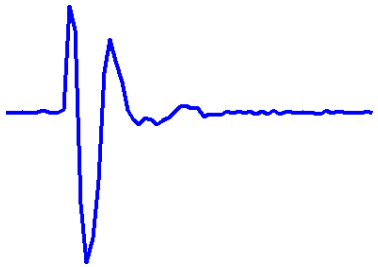


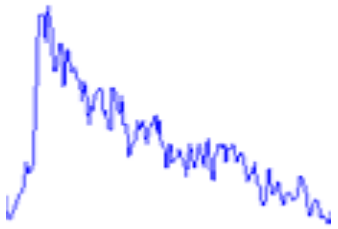
Boracchi, Trovò

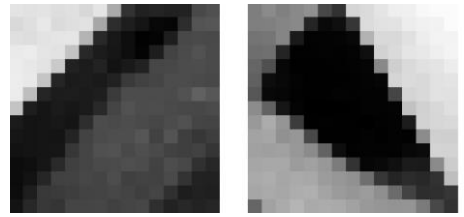
Real World Detection Problems

Random variable model **does not successfully apply to signals or images** (not even small portions)



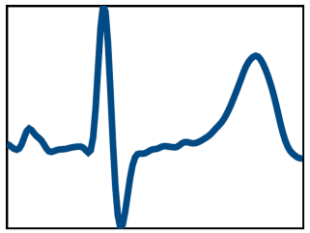
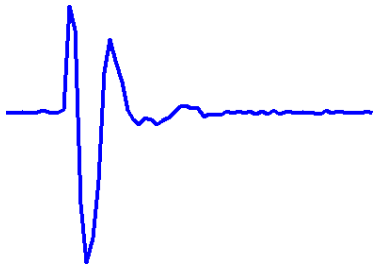
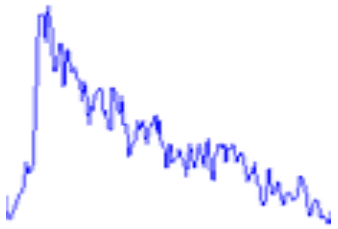
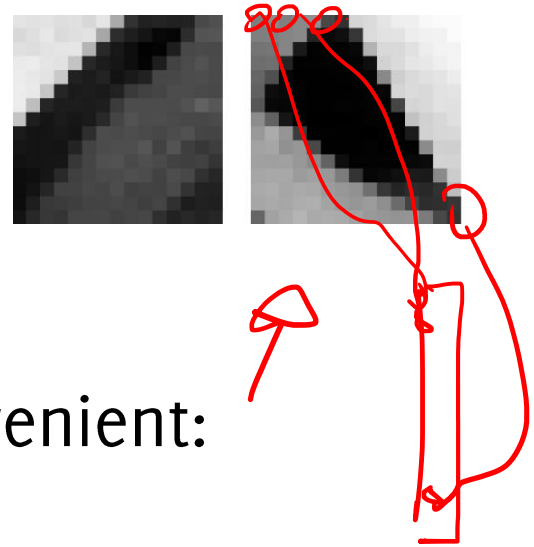






Real World Detection Problems

Random variable model **does not successfully apply to signals or images** (not even small portions)

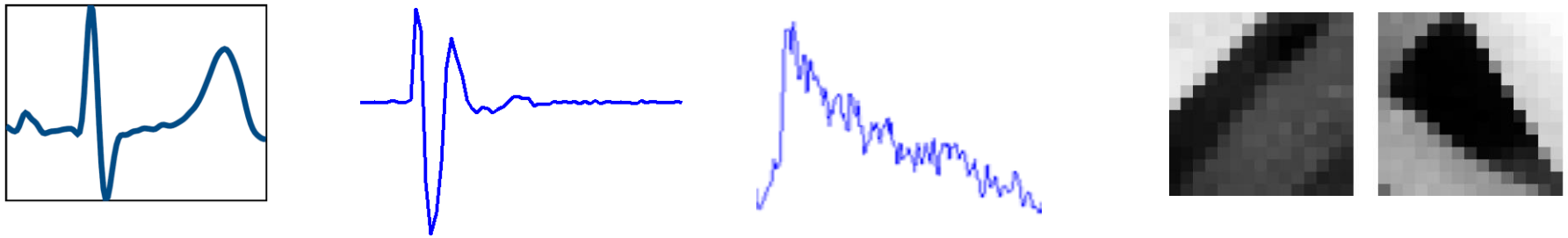





Stacking each signal $\boldsymbol{s} \in \mathbb{R}^d$ in a vector $\boldsymbol{x}$ is not convenient:

- **Data dimension** d can become **huge**
- **Correlation among components is difficult to model**

Real World Detection Problems

Random variable model **does not successfully apply to signals or images** (not even small portions)



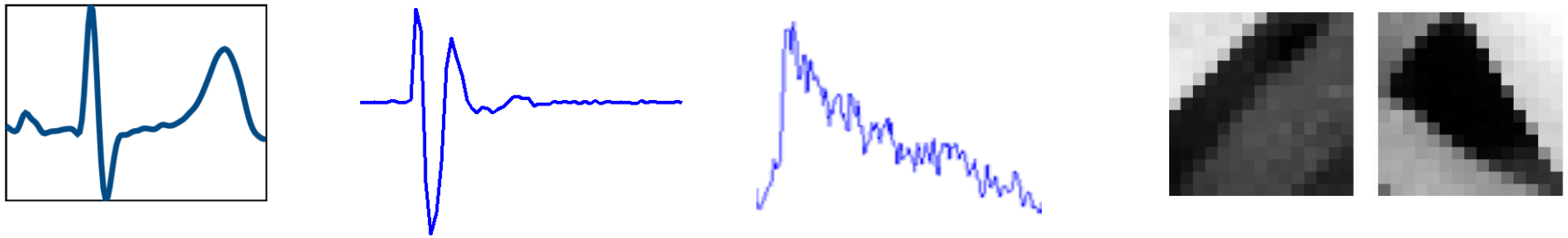
Stacking each signal $\boldsymbol{s} \in \mathbb{R}^d$ in a vector $\boldsymbol{x}$ is not convenient:

- **Data dimension** d can become **huge**
- **Correlation among components is difficult to model**

It is not easy to **estimate a density model** or threat these as **realizations of a random variable**

Real World Detection Problems

Random variable model **does not successfully apply to signals or images** (not even small portions)



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

- **Data dimension** d can become **huge**
- **Correlation among components is difficult to model**

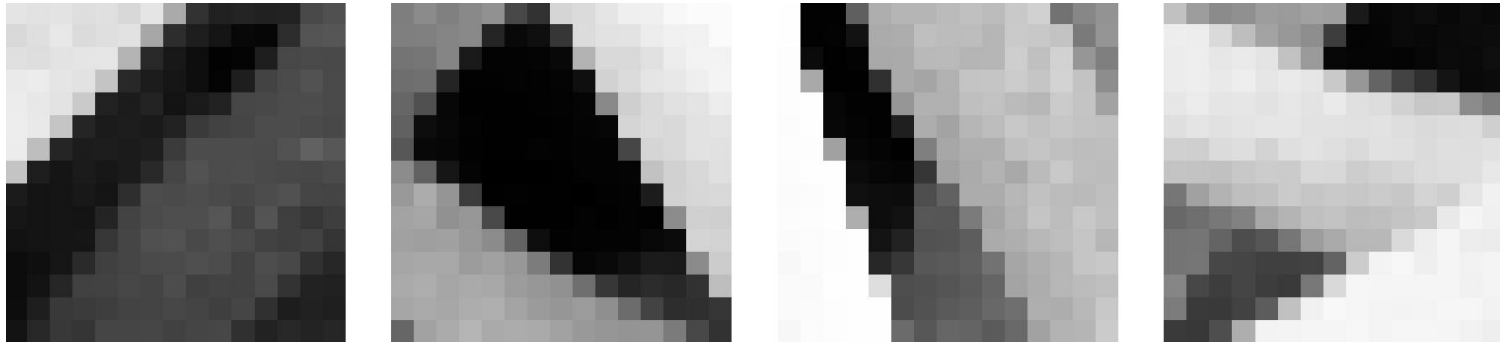
It is not easy to **estimate a density model** or threat these as **realizations of a random variable**

Moreover, when **normal data** exhibit a peculiar **structure**, we are interested in **detecting changes/anomalies affecting that structure**

Real World Detection Problems

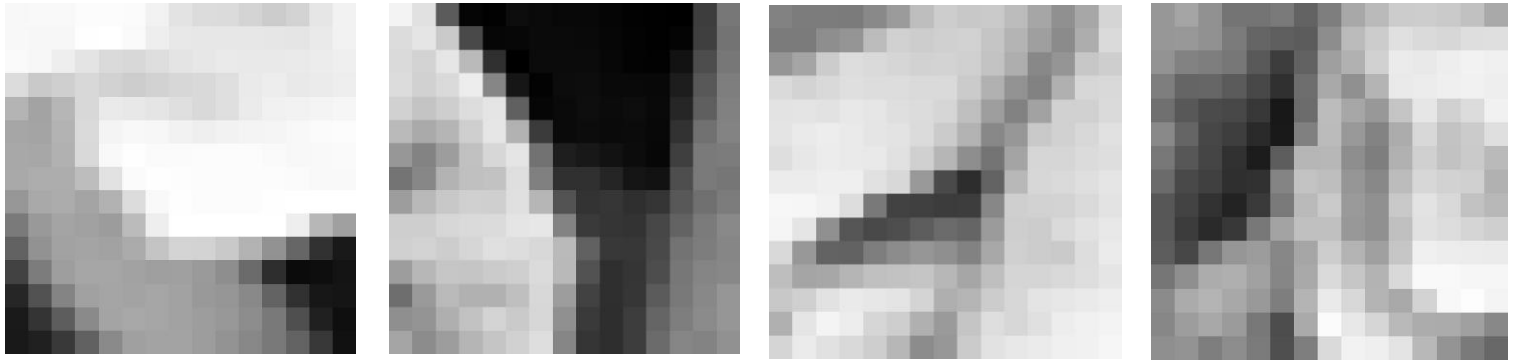
Normal patches -> background

- Exhibit a specific structure (geometry) or intensities



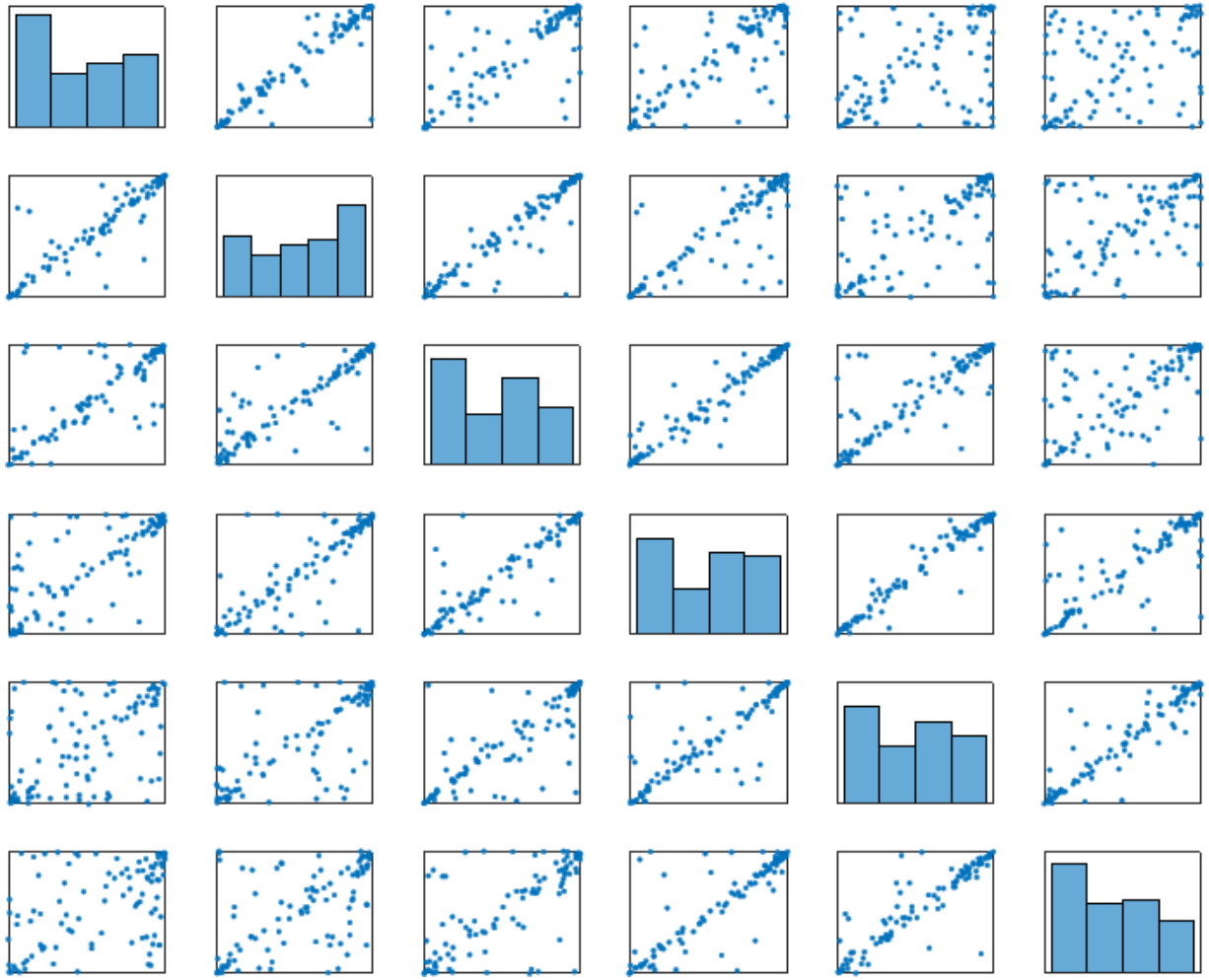
Anomalous patches:

- Are rare elements that do not confrom with the background



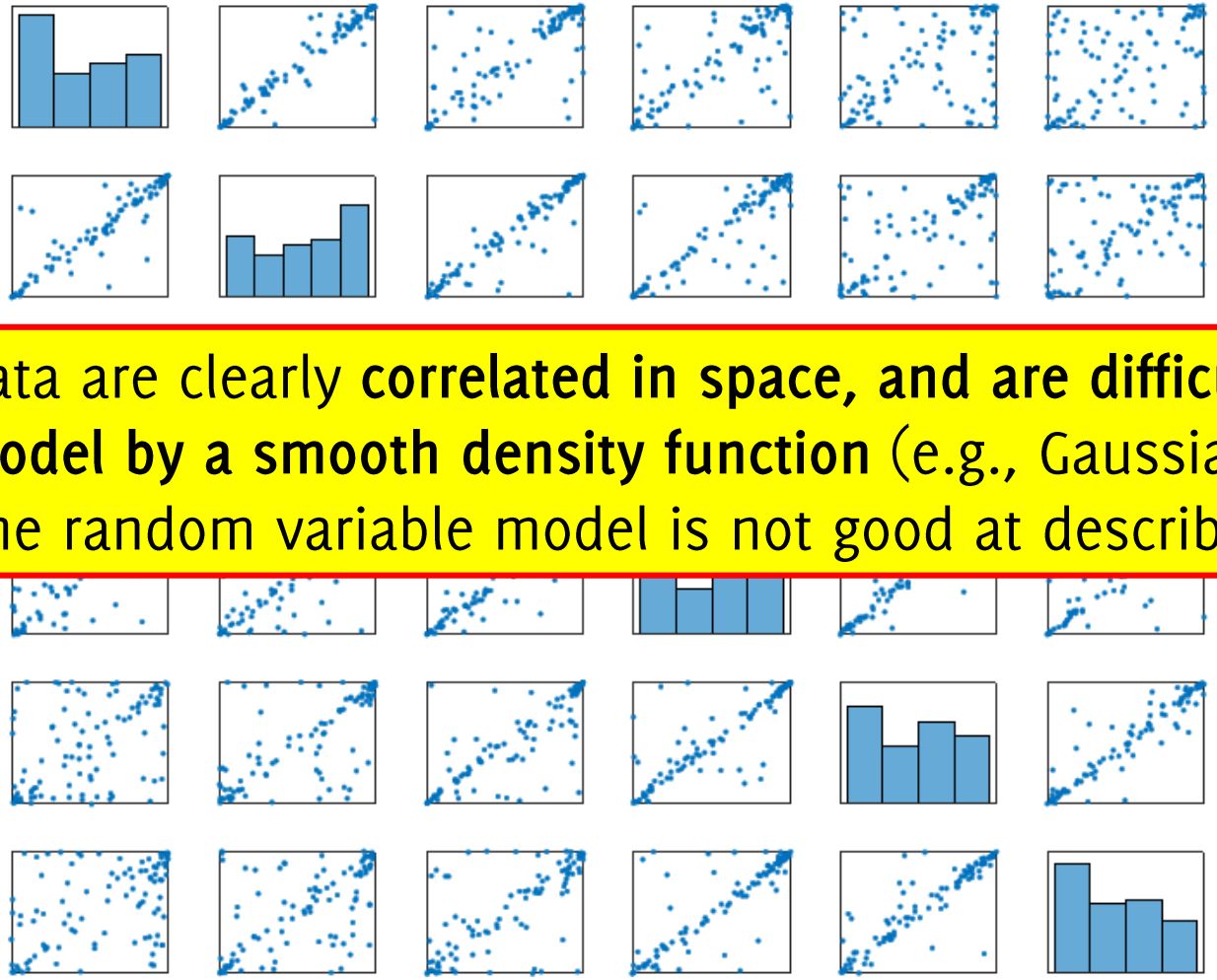
The limitations of the Random variable model

Distribution of adjacent pixel values inside a patch:



The limitations of the Random variable model

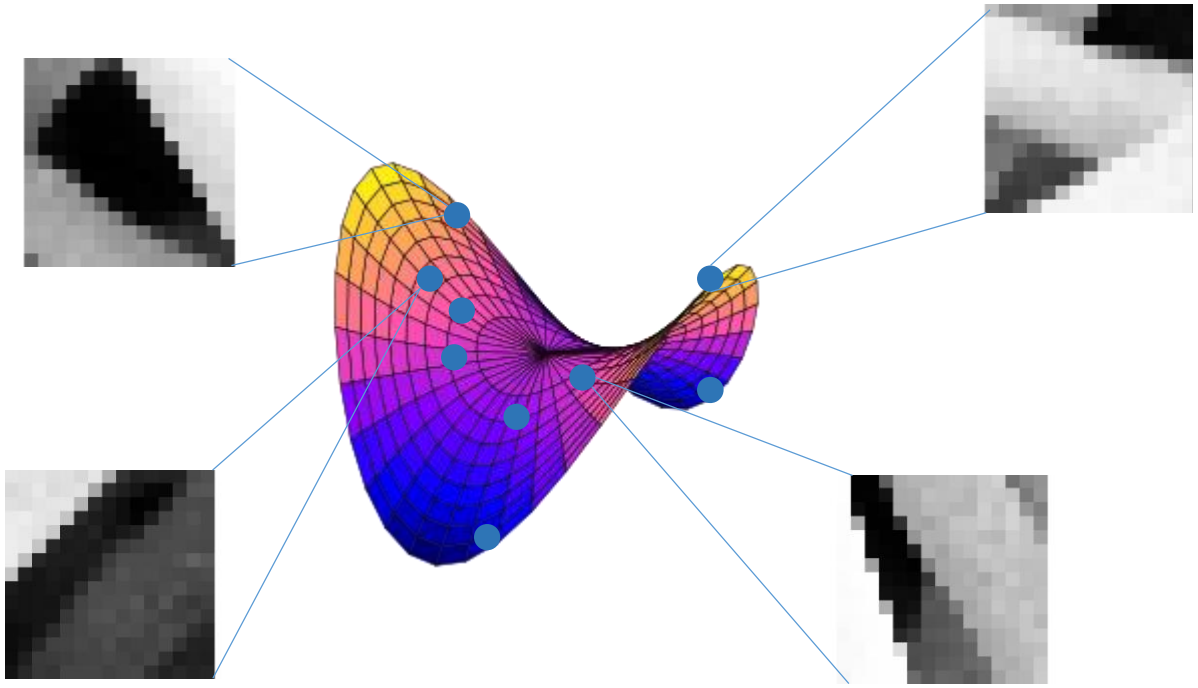
Distribution of adjacent pixel values inside a patch:



Data are clearly **correlated in space, and are difficult to model by a smooth density function** (e.g., Gaussians)
The random variable model is not good at describing images

THE LIMITATIONS OF THE RANDOM VARIABLE MODEL

Patches from natural images live close to a low dimensional manifold

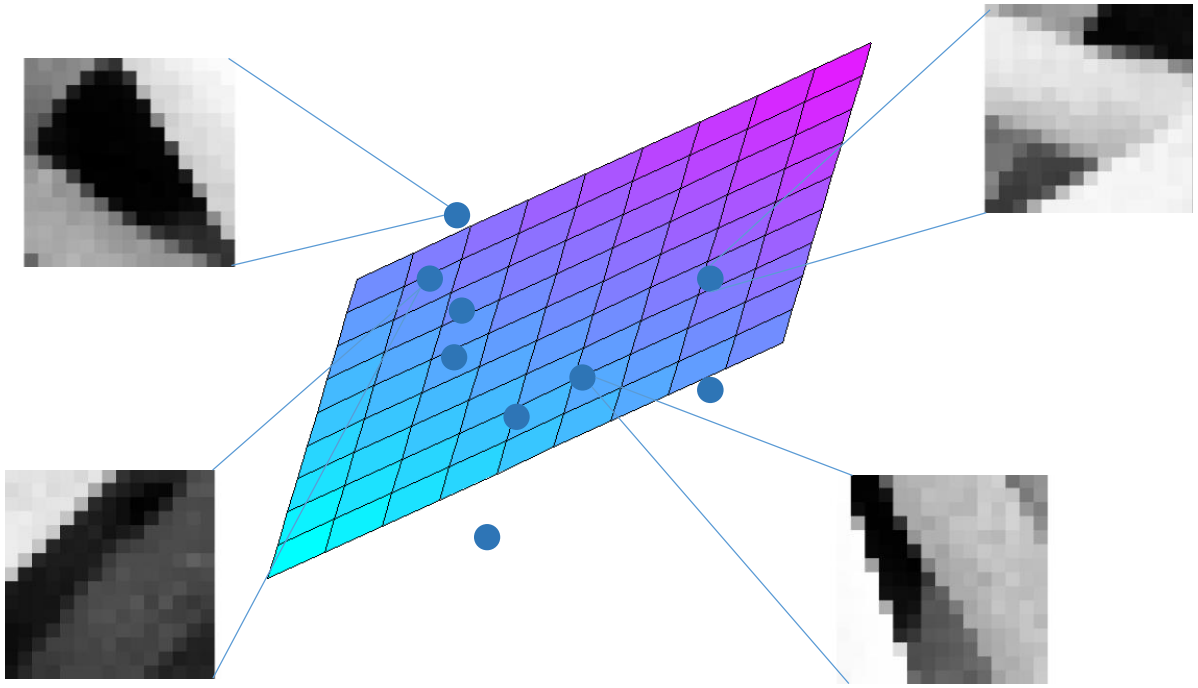


These means that patches can be well described by few latent variables

Bengio, Courville, Vincent, Representation Learning: A Review and New Perspectives, IEEE Pattern Analysis and Machine Intelligence ' 2013

A simple experiment

Let’s approximate this manifold with the simplest one: a linear subspace

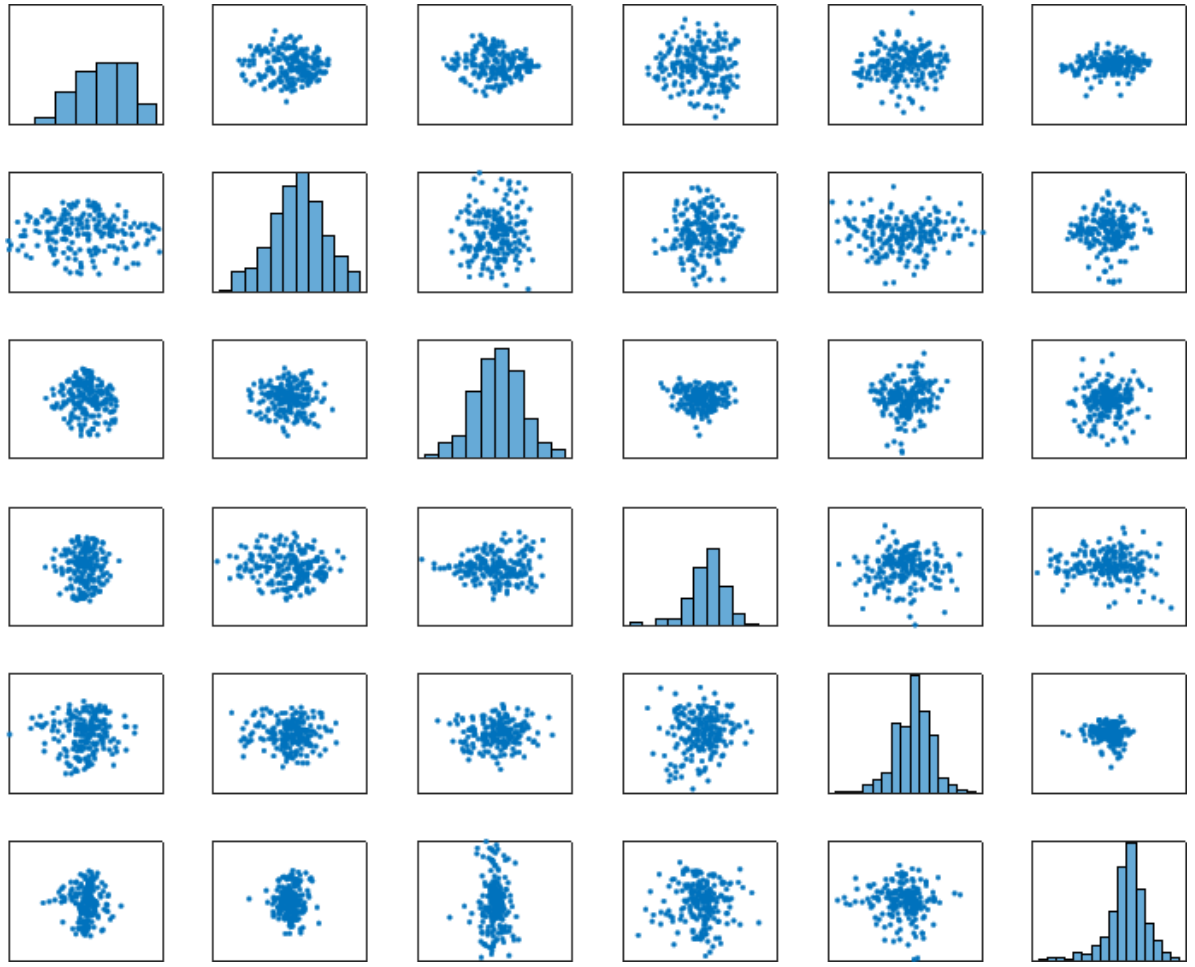


In practice, we compute the PCA of training patches and consider the PCA score as latent variables

Boracchi, Trovò

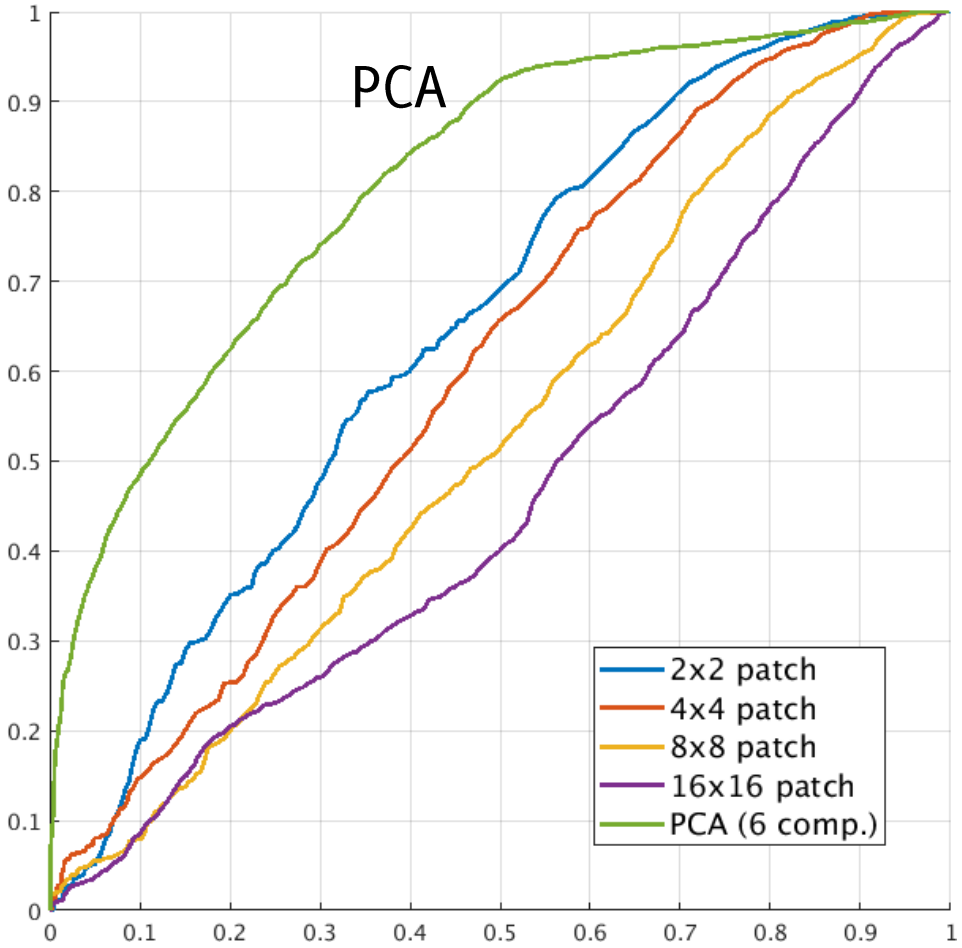
A simple experiment

Distribution of first 6 PCA coefficients:



A simple experiment

We fit a $\hat{\phi}_0 = \mathcal{N}(\mu, \Sigma)$ on PCA components to describe normal patches, and perform anomaly detection



Boracchi, Trovò

Anomaly Detection Out of the “Random Variable” World

Model-based approaches for images

Boracchi, Trovò

The three major ingredients

Most detection algorithms have three major ingredients:

- The **background model** $\mathcal{M}$, learned from normal data
- The **statistic / anomaly score**: $\mathrm{err}(\boldsymbol{s}), \mathcal{L}(\boldsymbol{s}), \mathcal{A}(\boldsymbol{s}), \ldots$
- **Decision rule** to detect, e.g. $\mathrm{err}(\boldsymbol{s}) \gtrless \gamma$ possibly controlling the FPR, as in other statistical detection methods

The Typical Approach

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Provide, for each test sample, an **anomaly score,** or measure of rareness, w.r.t. the learned model
3. Apply a **decision rule** to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Remark: Statistical-based approaches seen before use as background model the statistical distribution $\hat{\phi}_0$ and a statistic as anomaly score

The Typical Approach

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Remark: Statistical-based approaches seen before uses as background model the statistical distribution $\hat{\phi}_0$ and a statistic as anomaly score

The Typical Approach

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Remark: Statis ground model the sta score

The background model is used to **bring an image patch into the "random variable world"**

The Typical Approach

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Re

m

Once "having applied" the background model, one can use **anomaly detection methods for the "random variable world".**
This might require **fitting an additional (density) model** in the random variable world

The Typical Approach

Most of the considered methods

1. **Estimate** a **model** describing **normal data** (background model)
2. Use the background model to provide, for each test signal/patch, an **anomaly score,** or measure of rareness
3. Apply a **decision rule** to the anomaly score to detect anomalies (typically thresholding)
4. **[optional]** Perform **post-processing** operations to enforce smooth detections and avoid isolated pixels that are not consistent with neighbourhoods

Rem
mod

And it is important to control the False Positive Rate or the ARL_0 of the overall monitoring scheme

The Typical Approach

Different options to learn the background model

- **semi-supervised approach,** background model is learned exclusively normal data
- **unsupervised approach**, background model is fit to both normal and anomalous but it is robust to outliers

Semi-supervised AD methods out of the RVW

Out of the "Random Variable" world

- Detrending-based methods
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features

Semi-supervised AD methods out of the RVW

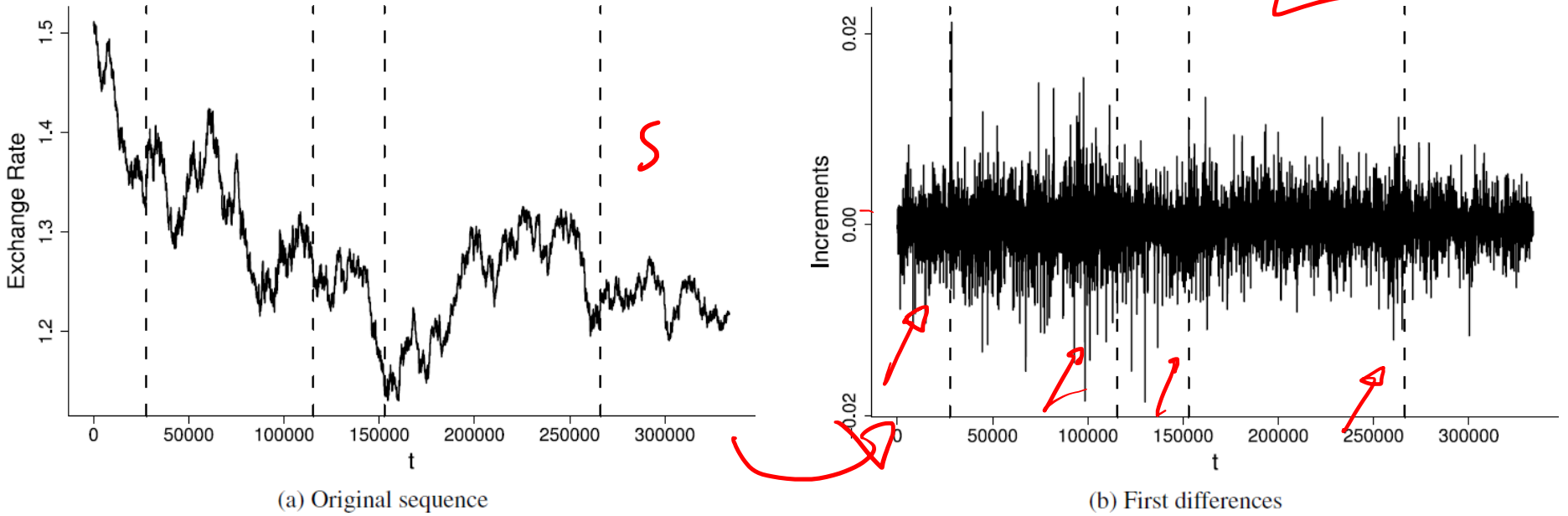
Out of the "Random Variable" world

- Detrending-based methods
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features

… Out of the random variable world

We can "get rid of the structure" by **detrending/filtering:**

- removing the deterministic/correlated components of the data (e.g. by computing derivatives, or by polynomial fit)



(a) Original sequence

(b) First differences

But this might not always apply, since we get rid of "all the structures", thus also those from anomalous signals.

G.J. Ross, D.K. Tasoulis, N.M. Adams *"Nonparametric monitoring of data streams for changes in location and scale"* Technometrics 53 (4), 379-389, 2011

Semi-supervised AD methods out of the RVW

Out of the "Random Variable" world

- Detrending-based methods
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features

Reconstruction-based Methods

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

Detection is performed by using a model $\mathcal{M}$ which **represents normal data**:

- **During training:** learn the model $\mathcal{M}$ from training set TR
- **During testing:**
  - Reconstruct each test signal $\boldsymbol{s}$ through $\mathcal{M}$.
  - Assess the **residuals** between $\boldsymbol{s}$ and its reconstruction

The rationale is that $\mathcal{M}$ can **reconstruct only normal data**, thus anomalies are expected to yield large reconstruction errors.

Bengio, Y., Courville, A., Vincent, P. *"Representation learning: A review and new perspectives"*. IEEE TPAMI 2013

Reconstruction-based Methods

Popular models are:

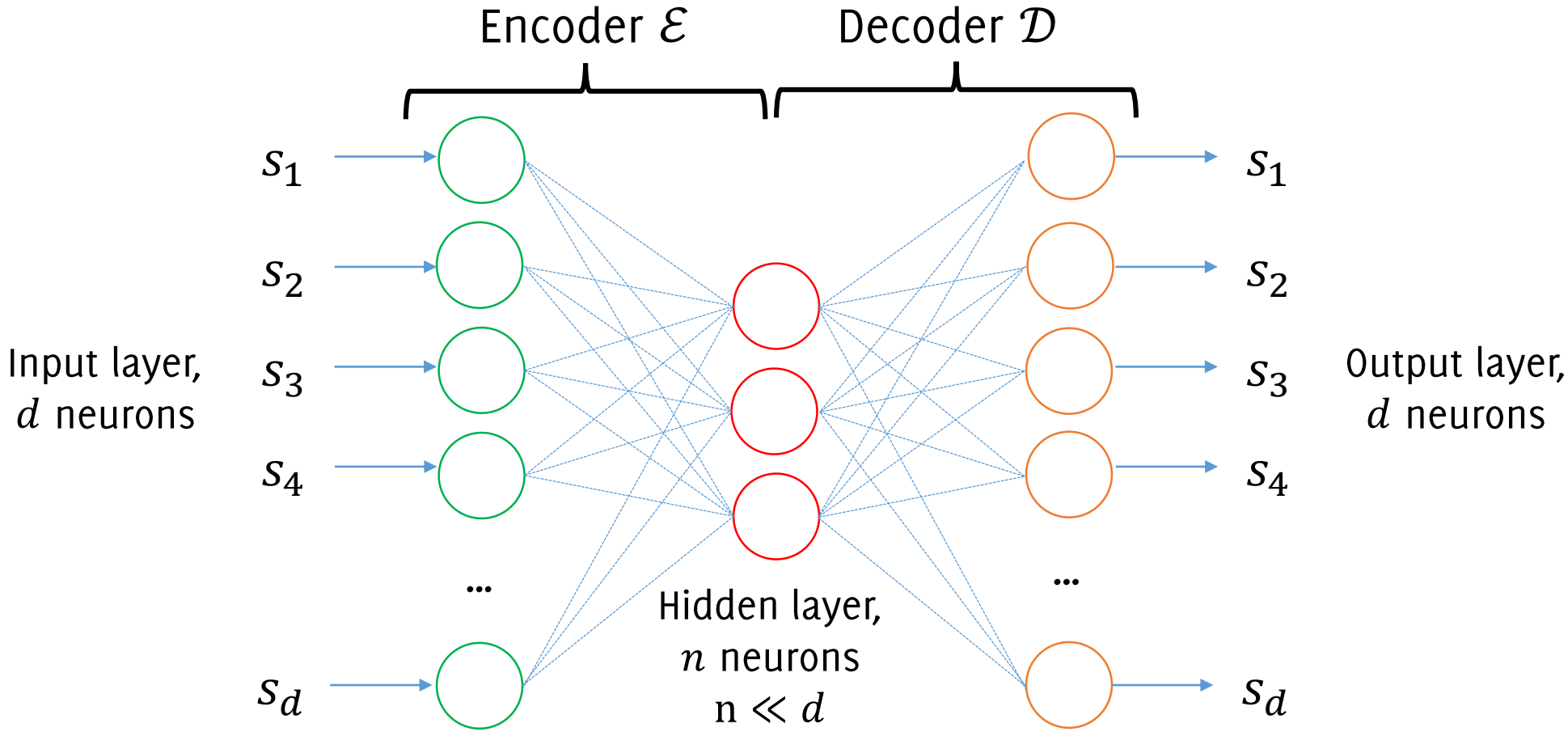
- autoregressive models for time series (ARMA, ARIMA…)
- neural networks, in particular auto-encoders, for higher dimensional data
- projection on subspaces / manifolds
- dictionaries yielding sparse-representations
- Deep Learning Models

Projection and Dictionaries can be also interpreted as subspace methods

Reconstruction-based Methods

Autoencoders are non-parametric models (neural networks) trained to reconstruct data in a training set.

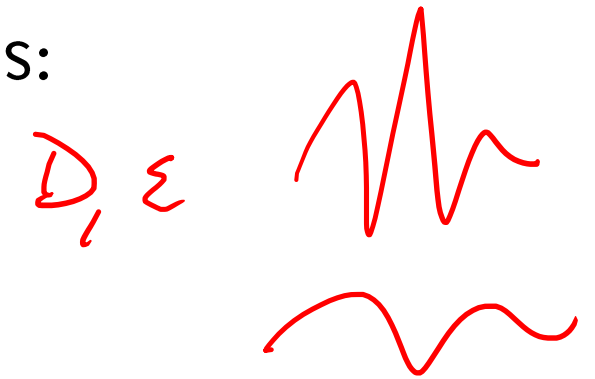
The typical structure of an autoencoder is:



Reconstruction-based Methods

Autoencoders are non-parametric models (neural networks) trained to reconstruct data in a training set. The typical loss function is:

$$\sum_{\boldsymbol{s}\in S} \|\boldsymbol{s} - \mathcal{D}(\mathcal{E}(\boldsymbol{s}))\|_2$$



and training of $\mathcal{D}(\mathcal{E}(\cdot))$ is performed through standard backpropagation algorithms (e.g. SGD)

$$\|s - D(\mathcal{E}(s))\|_2$$

Remarks

- AE typically does not provide exact reconstruction since $n \ll d$.
- Additional regularization terms might be included in the loss function

Bengio, Y., Courville, A., Vincent, P. *"Representation learning: A review and new perspectives"*. IEEE TPAMI 2013

Mishne, G., Shaham, U., Cloninger, A., & Cohen, I. *Diffusion nets*. Applied and Computational Harmonic Analysis (2017).

Monitoring the Reconstruction Error

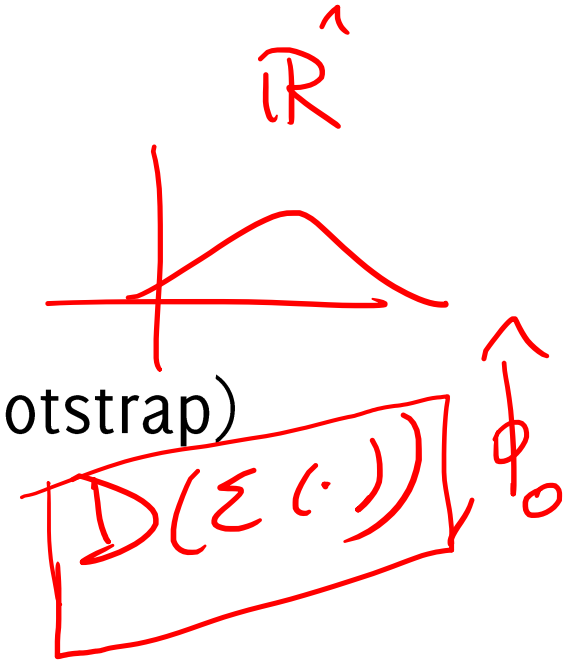
Detection by reconstruction error monitoring (AE notation)

Training (Monitoring the Reconstruction Error):

1. Train the model $\mathcal{D}(\mathcal{E}(\cdot))$ from the training set TR
2. Learn the distribution of reconstruction errors

$$\mathrm{err}(\boldsymbol{s}) = \left\|\boldsymbol{s} - \mathcal{D}\left(\mathcal{E}(\boldsymbol{s})\right)\right\|_2, \qquad \boldsymbol{s} \in V$$

over a validation set $V \neq TR$ and define a threshold γ (bootstrap)

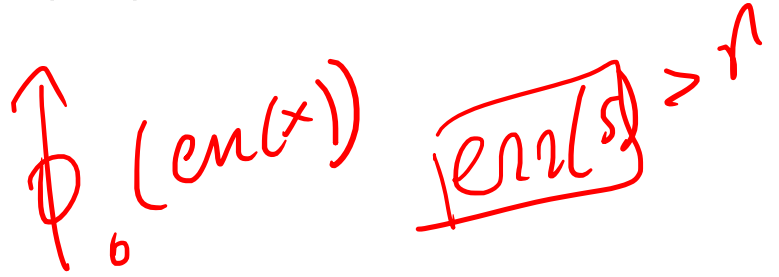


Testing (Monitoring the Reconstruction Error):

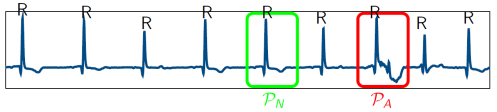
1. Perform encoding and compute the reconstruction error

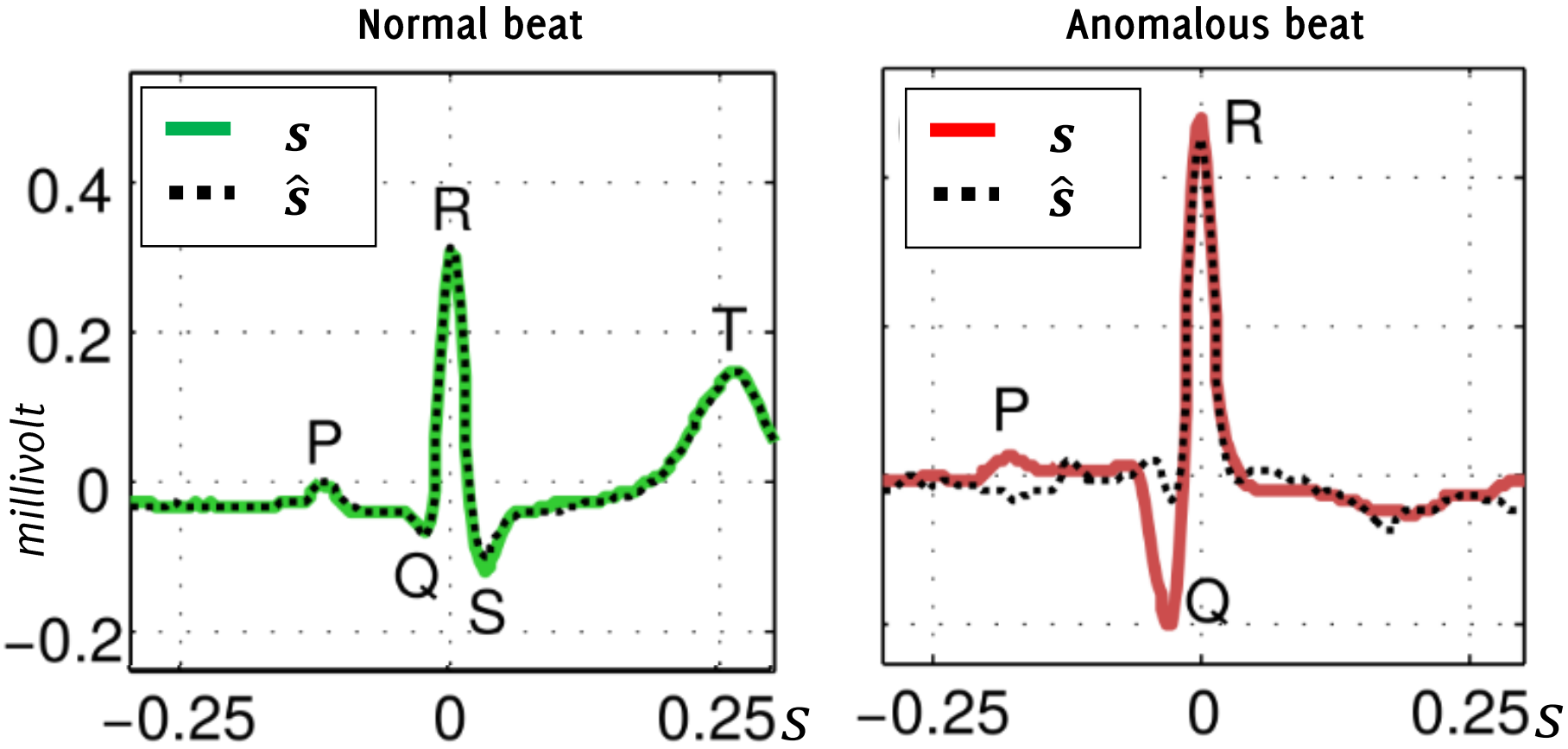
$$\mathrm{err}(\boldsymbol{s}) = \left\|\boldsymbol{s} - \mathcal{D}\left(\mathcal{E}(\boldsymbol{s})\right)\right\|_2$$

2. Consider $\boldsymbol{s}$ anomalous when $\mathrm{err}(\boldsymbol{s}) > \gamma$



Reconstruction-based monitring



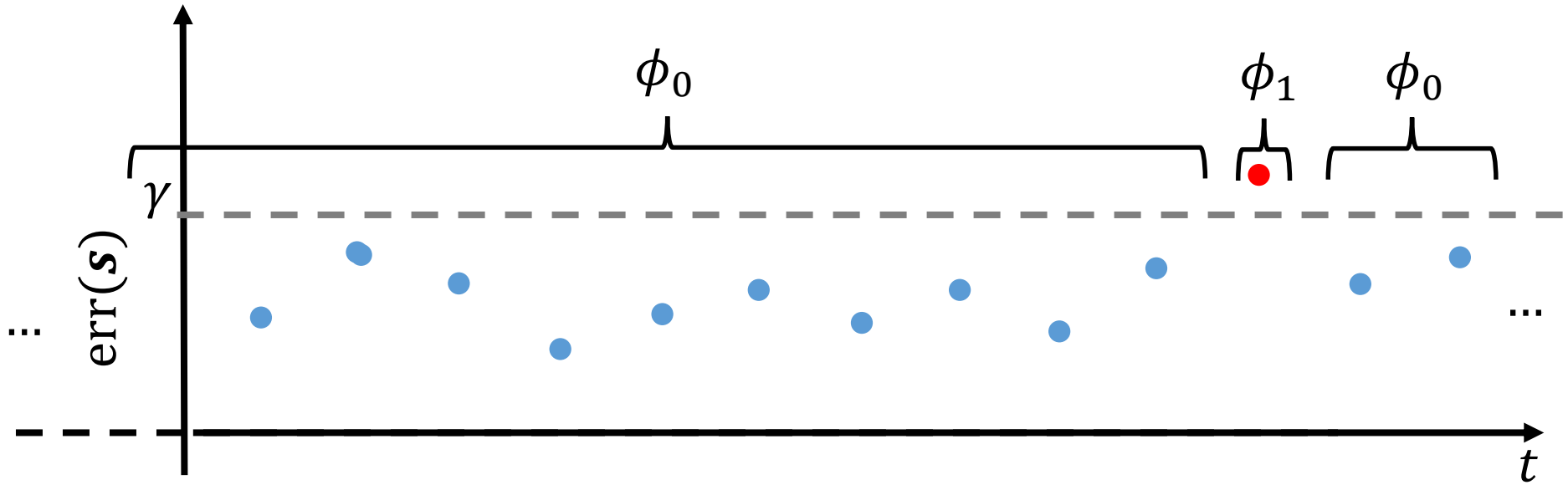


Carrera D., Rossi B., Fragneto P., Boracchi G. "*Online Anomaly Detection for Long-Term ECG Monitoring using Wearable Devices*", Pattern Recognition, Elsevier 2019

Monitoring the Reconstruction Error

Normal data are expected to yield values of $\mathrm{err}(\boldsymbol{s})$ that **are low**, while anomalies do not. This property holds **when the model $\mathcal{M}$ was specifically learned to describe normal data**

Outliers can be detected by a threshold on $\mathrm{err}(\boldsymbol{s})$



Semi-supervised AD methods out of the RVW

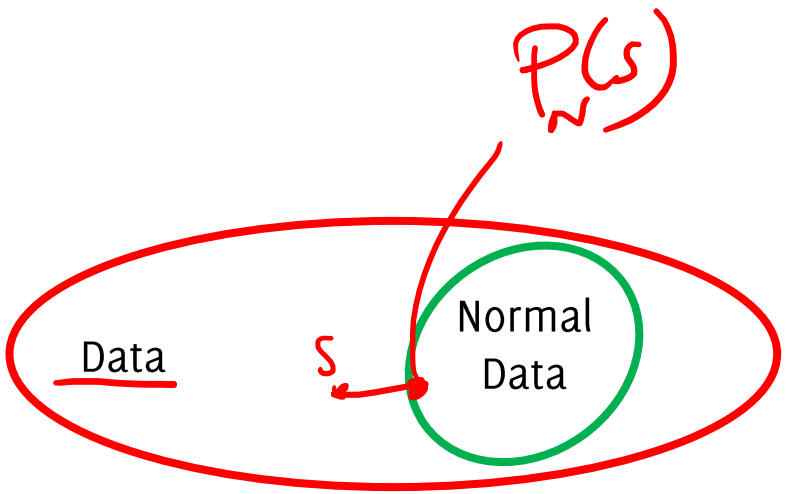
Out of the "Random Variable" world

- Detrending-based methods
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features

Subspace Methods

The underlying assumption is that

- **normal data live in a subspace** that can be identified by TR
- **anomalies can be detected by projecting test data** in such subspace and by monitoring the reconstruction error (distance with the projection)

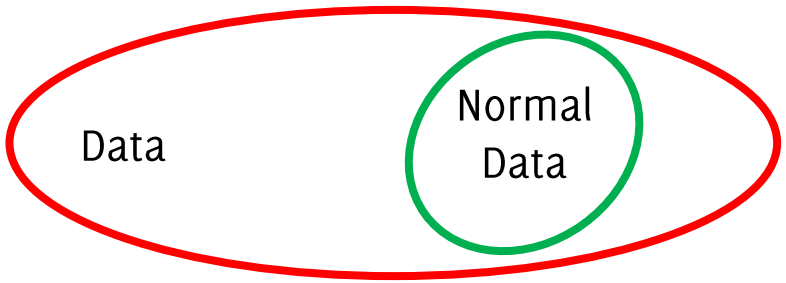


V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

Subspace Methods

A few example of **models used for describing normal patches**:

- **Fourier transform**: normal data can be expressed by a few specific frequencies
- **PCA**: normal data live **in the linear subspace of the first components**.
- Robust PCA: defined on the ℓ^1 distance to be insensitive to outliers in normal data.
- Kernel PCA: normal patches live in a non-linear manifold.
- **Random projections**



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

Subspace Methods

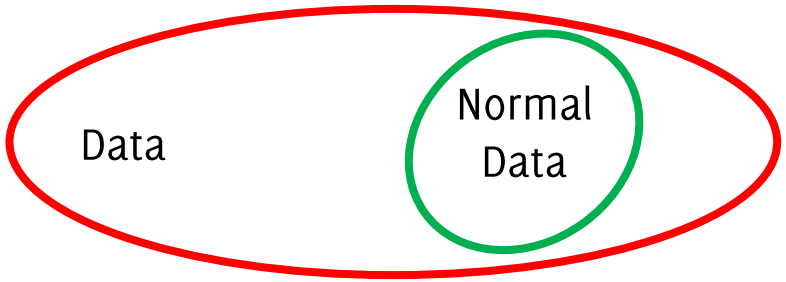
A few example of **models used for describing normal patches:**

- **Fourier transform:** normal data can be expressed by a few specific frequencies

- **PCA:** normal data live **in the linear subspace of the first components.**
- Robust PCA: defined on the ℓ^1 distance to be insensitive to outliers in normal data.
- Kernel PCA: normal patches live in a non-linear manifold.

Data Driven

- **Random projections**



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

Subspace Methods: Statistics

Compute the **the projection over a subspace characterizing normal data,**

$$\boldsymbol{x} = P^T \boldsymbol{s}, \qquad P \in \mathbb{R}^{m \times d}, \qquad m \ll d$$

which is the projection over the first m principal components and a way to reduce data-dimensionality. When $m = d$ you get perfect reconstruction on any normal and anomalous data!

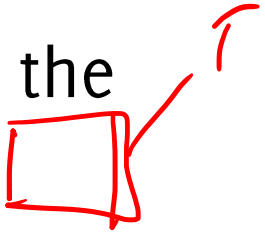
$P^{-1} = P^T$

- Monitor the **reconstruction error**:

  $$\text{err}(\mathbf{s}) = \|\boldsymbol{s} - PP^T \boldsymbol{s}\|_2$$

  which is the distance between $\boldsymbol{s}$ and its projection $PP^T \boldsymbol{s}$ over the subspace of normal patches

$P \in \mathbb{R}^d$



- Alternatively, monitor the **least-principal component only**, which like an anomaly score should be low in normal data.
- Alternatively, **monitor projections** coefficient $\boldsymbol{x} = P^T \boldsymbol{s}$ via multivariate statistical model/test

$m = 4$

Fifth Matlab Assignment

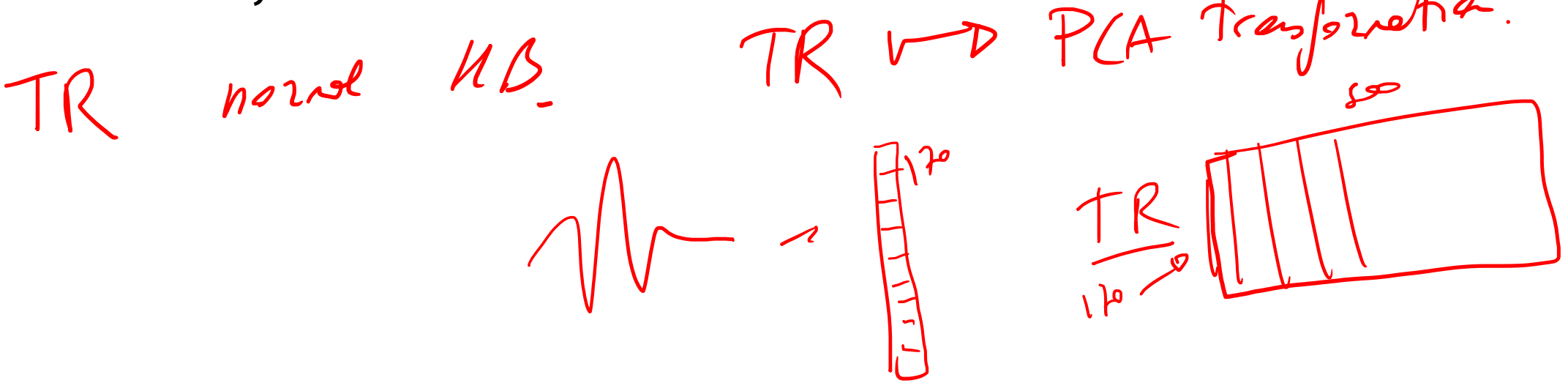
Boracchi, Trovò

Fifth Matlab Assignment

Goal: You have to implement a model based on PCA for detecting anomalous heart-beats.

Data: You will be provided with both normal and anomalous ECG signals, already split and aligned in portions corresponding to an heartbeat.

Annotation are also provided but these can be used for performance assessment only

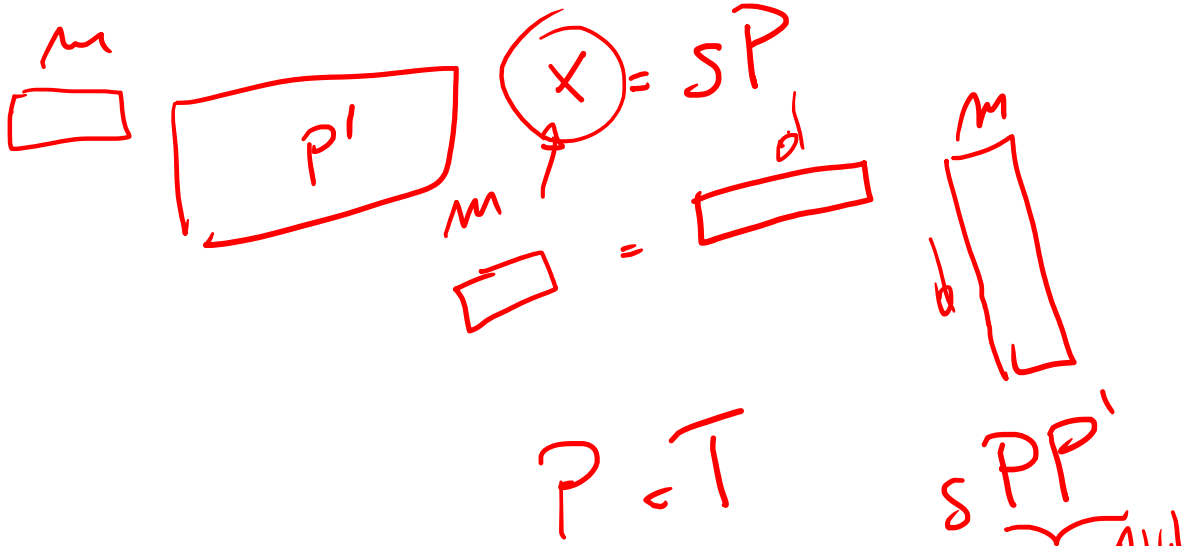


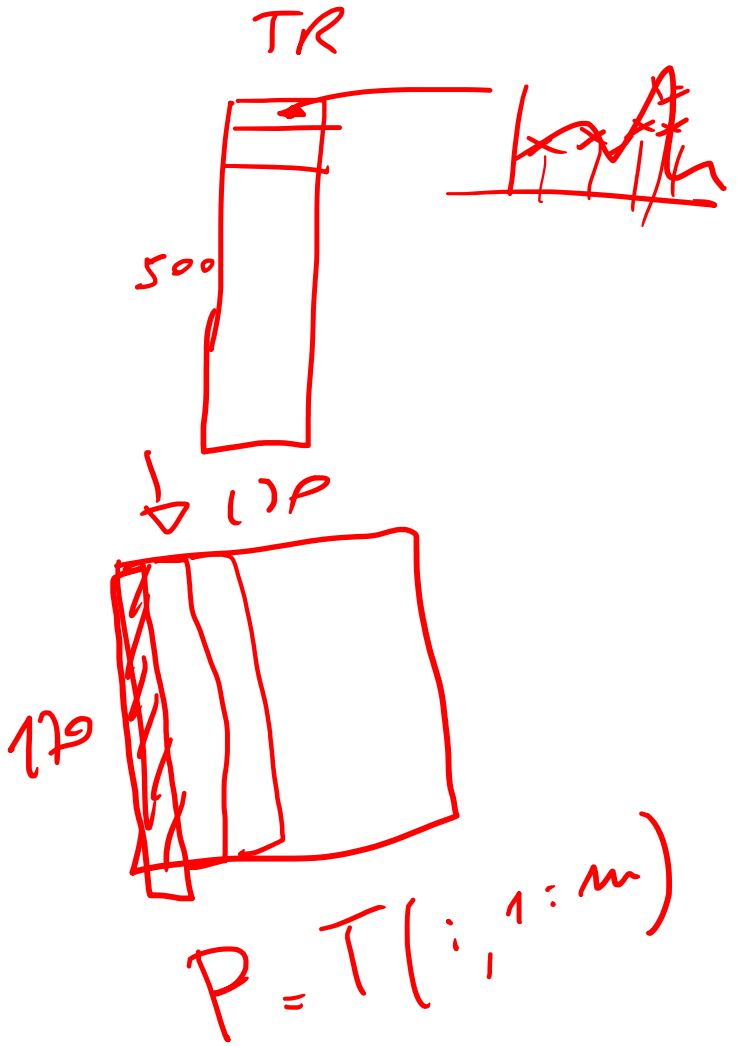
TR → Pca projection

$[T, scor, L] \equiv PCA(TR)$

$L(1-m)$ $\quad s \in \mathbb{R}^{170}$

$t \quad xP' \quad sT$





$P = T(:, 1:m)$

$\varepsilon(s)$

The Data

Vectors arranged in matrices corresponding to normal/anomalous HB

Watch out:

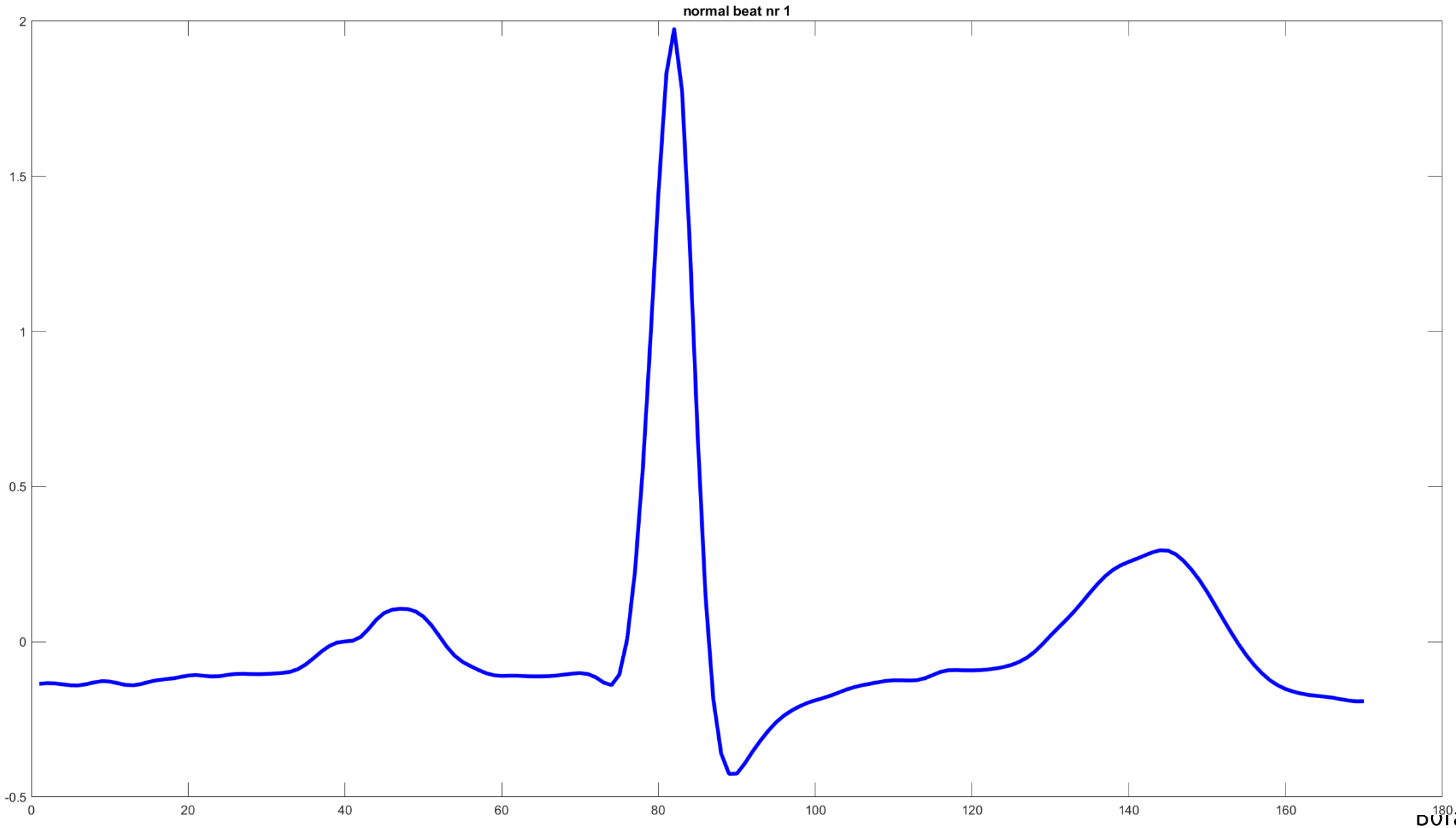
- Each signal is a **row vector, as this is how PCA operates (we used column vectors instead so far)**
- It is convenient to **center each sample before applying transformation.** This can be done by subtracting the mean of each sample.

Training and Test set has been already separated in two matrices.

Annotated labels are provided on the test set.

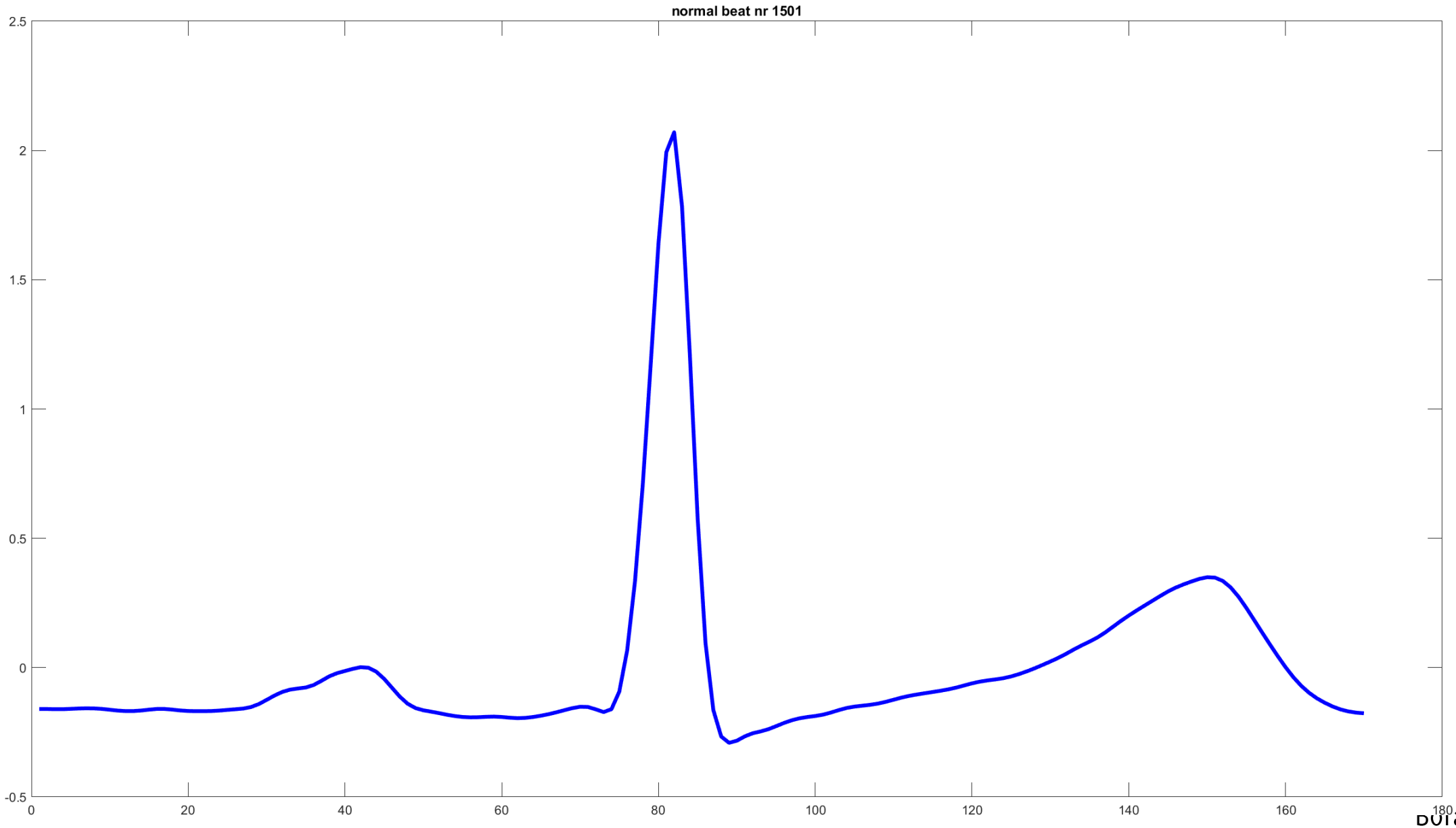
Training set is made of normal data only (semi-supervised settings)

Normal HB

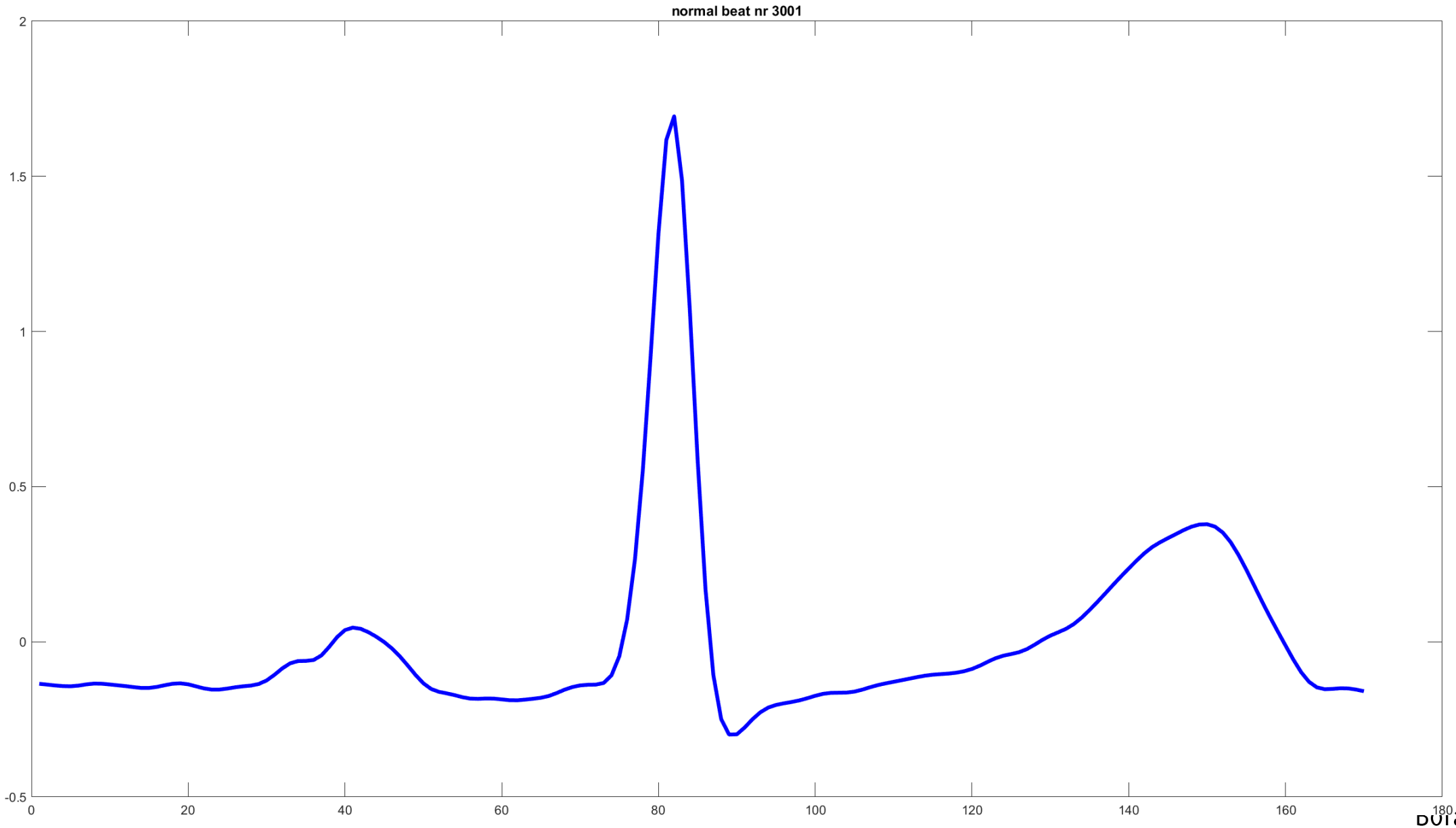


Bonacchi, Trovò

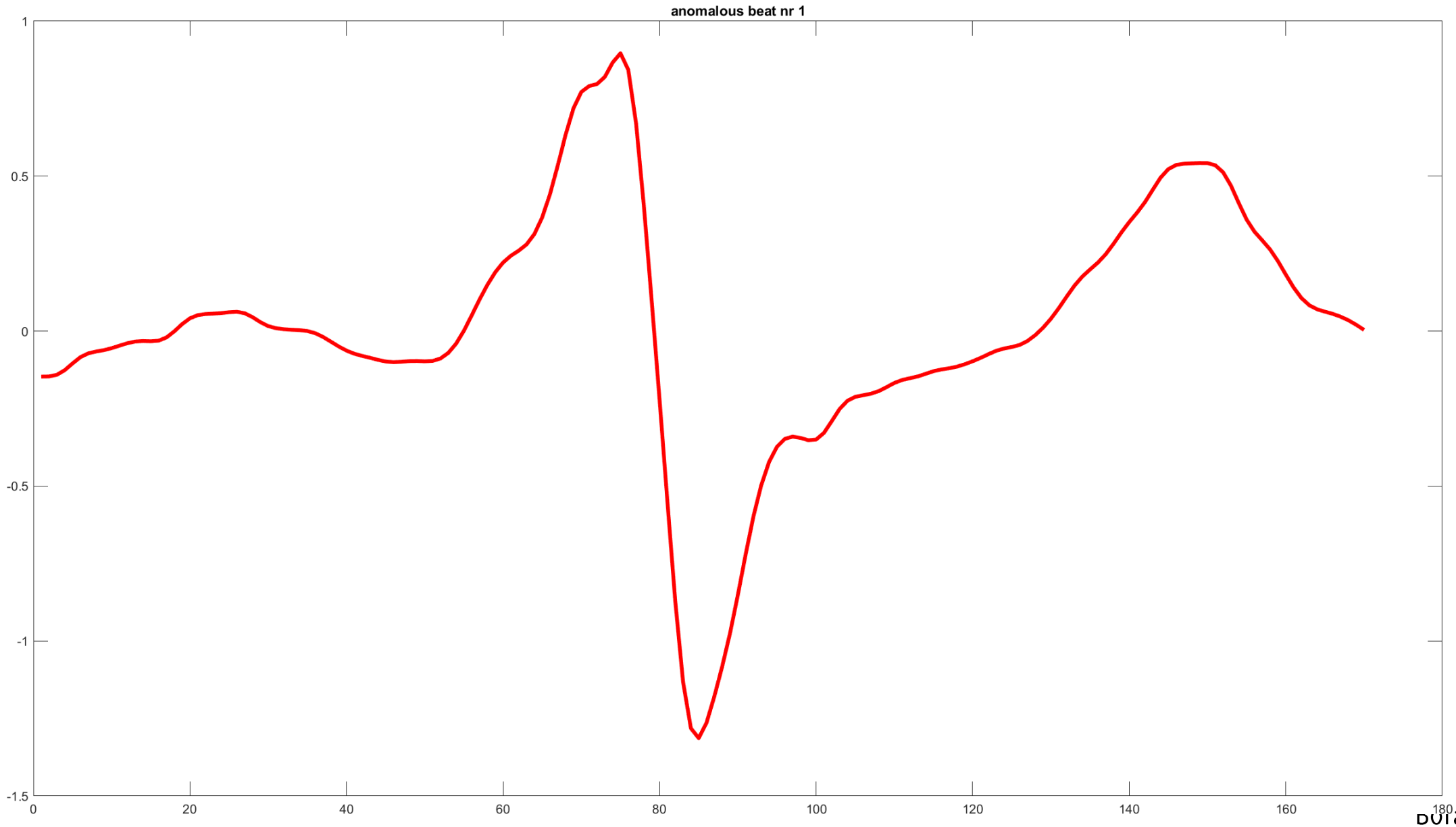
Normal HB


normal beat nr 1501
2.5
2
1.5
1
0.5
0
-0.5
0
20
40
60
80
100
120
140
160
180

Normal HB

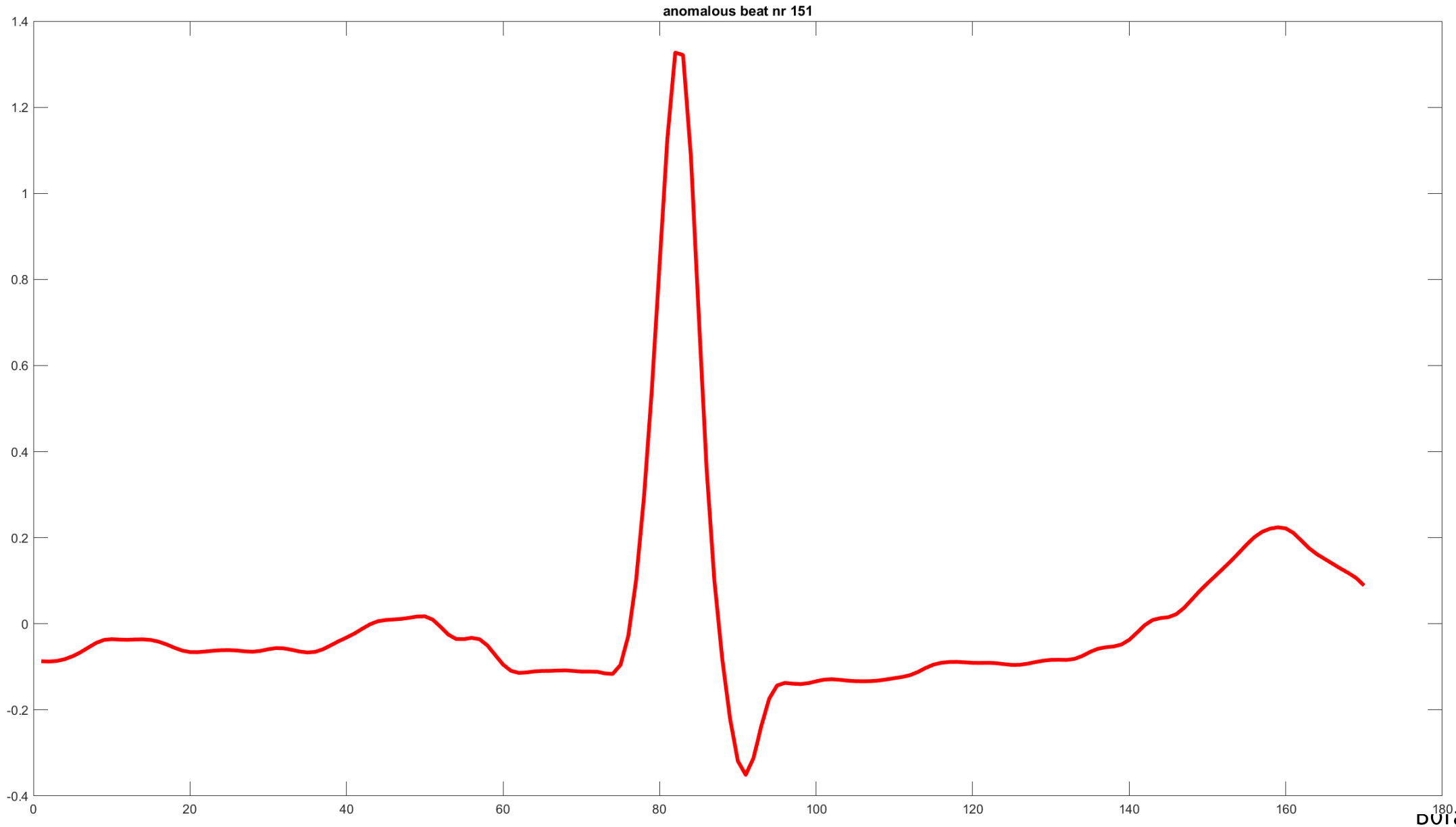


Anomalous HB



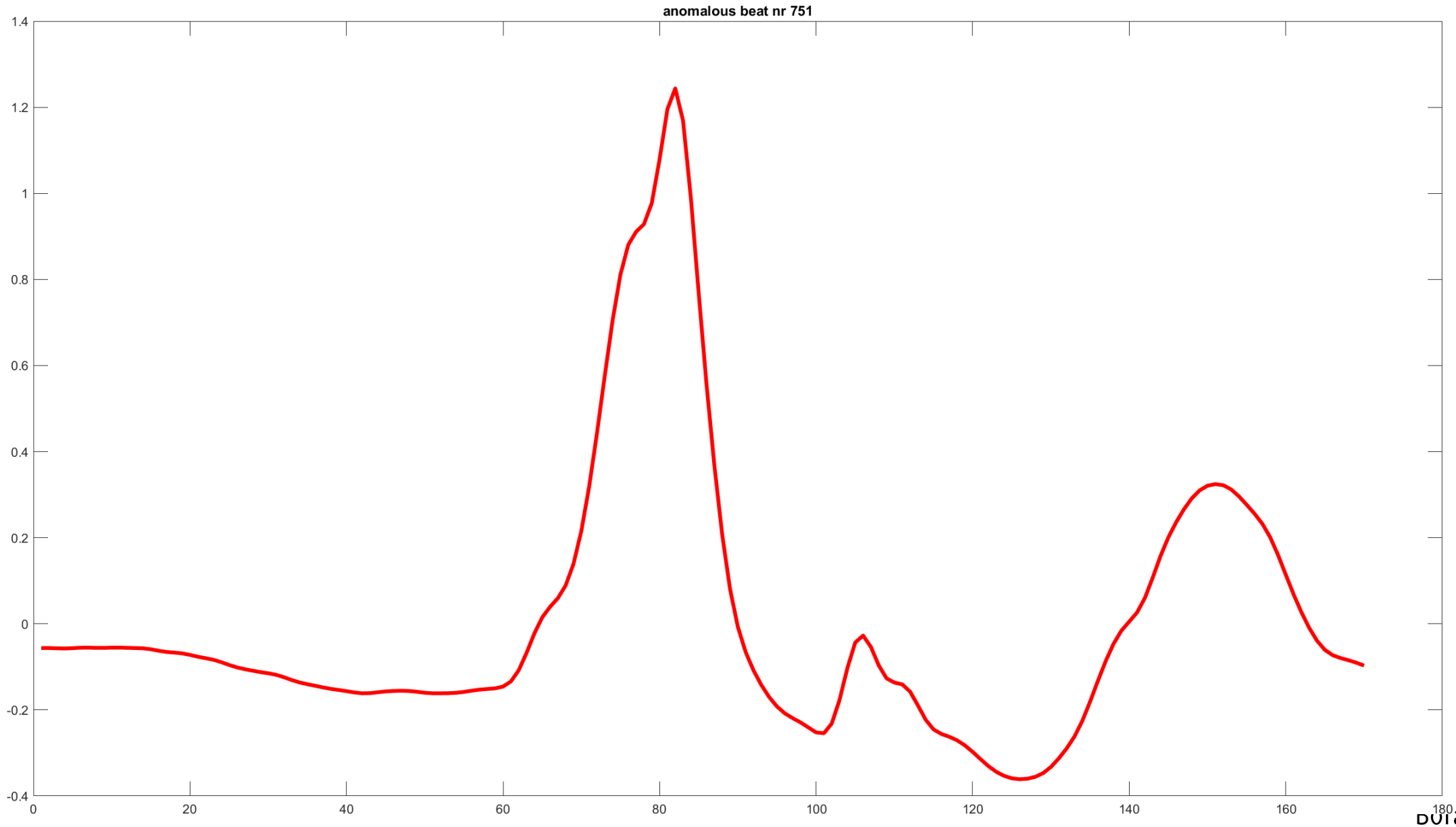
Bonacchi, Trovò

Anomalous HB



Bonacchi, Trovò

Anomalous HB

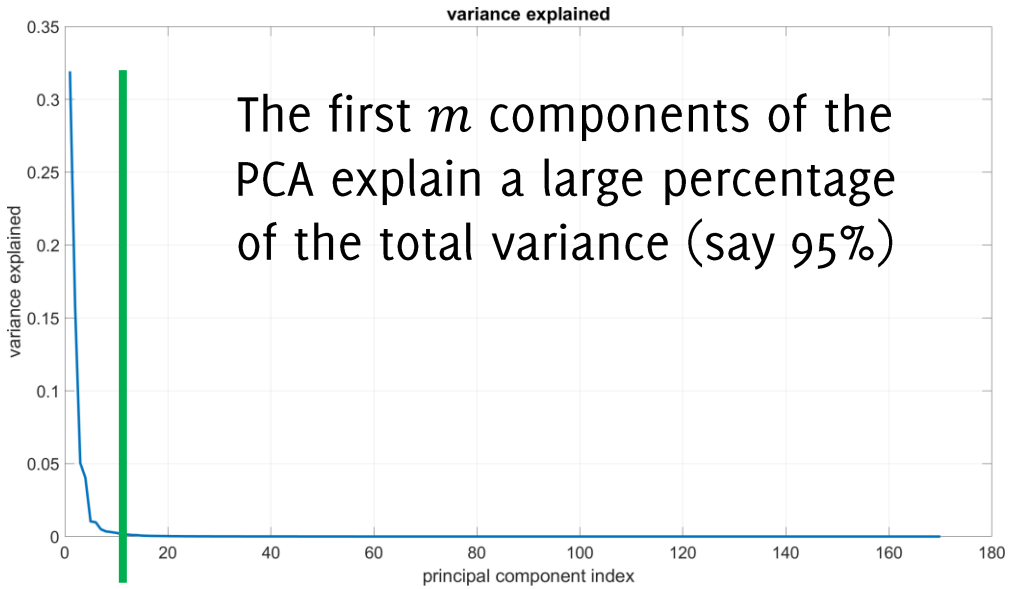


Bonacchi, Trovò

The Model $\mathcal{M}$ for normal data

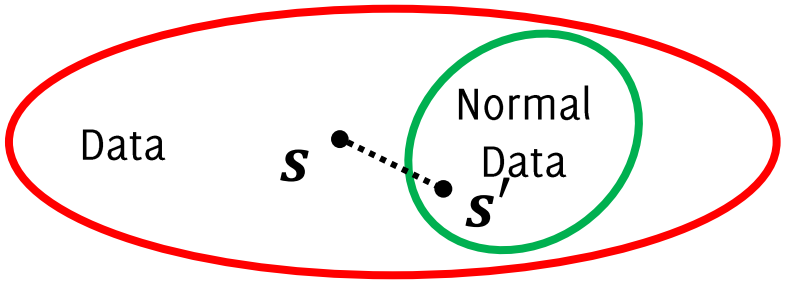
We assume normal data lives in a m −dimensional space that can be learned from the TS with $m \ll d$

We define this projection by **truncating the PCA transformation** computed over the training set TS



The first m components of the PCA explain a large percentage of the total variance (say 95%)

This is the subspace of normal HB that is spanned by the first m PCs



The Projection

Once the first m components of the PCA have been identified, it is possible to compute the coefficients of the projection over the PCA subspace $\forall \boldsymbol{s} \in \mathbb{R}^{1,d}$

$$\boldsymbol{s} \rightarrow \boldsymbol{x} = \boldsymbol{s}\, P$$

Being $P \in \mathbb{R}^{d,m}, m \ll d$, $\boldsymbol{x} \in \mathbb{R}^{m}$ are the first m principal components (i.e. the m columns of the PCA transformation matrix)

Remember **now signals are arrange row-wise in vectors**

The Reconstruction

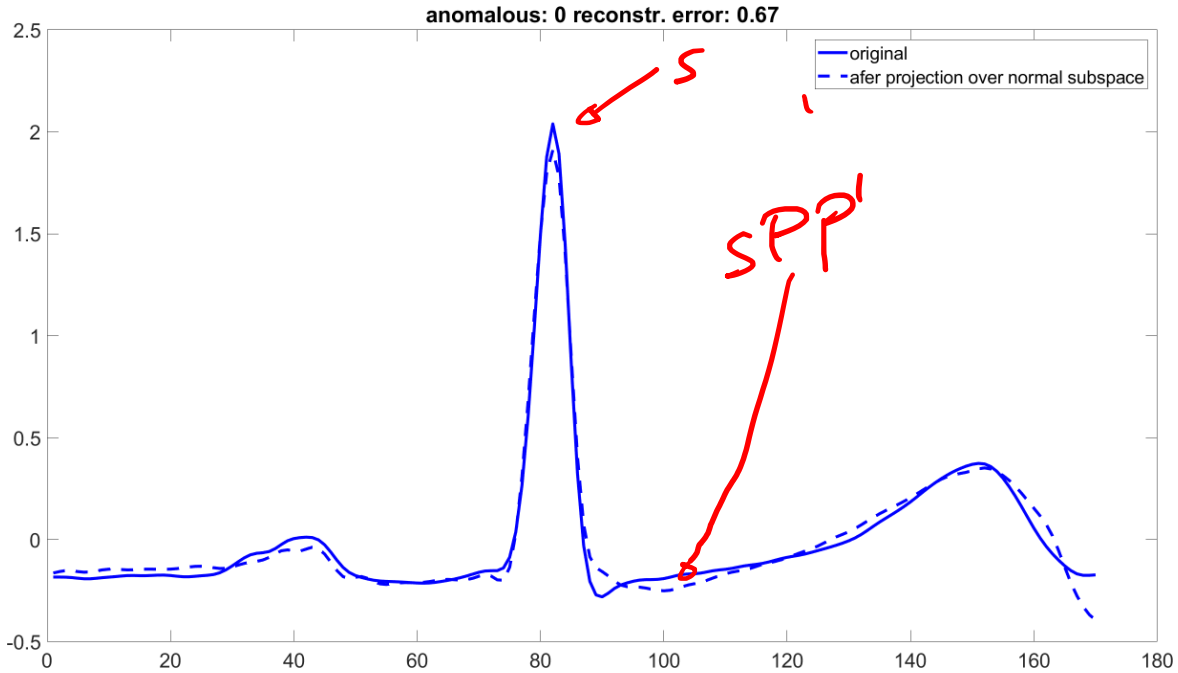
The reconstructed signal from the projection is:

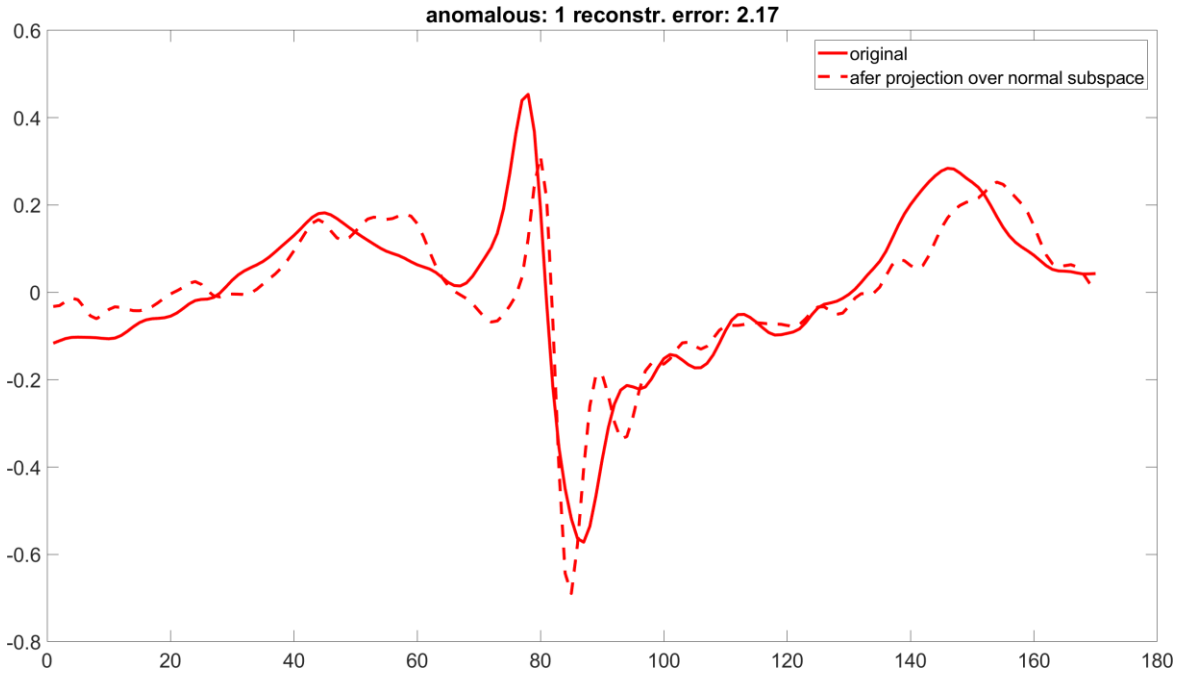
$$\boldsymbol{x}P^T \in \mathbb{R}^d$$

Which can be compared with the original signal

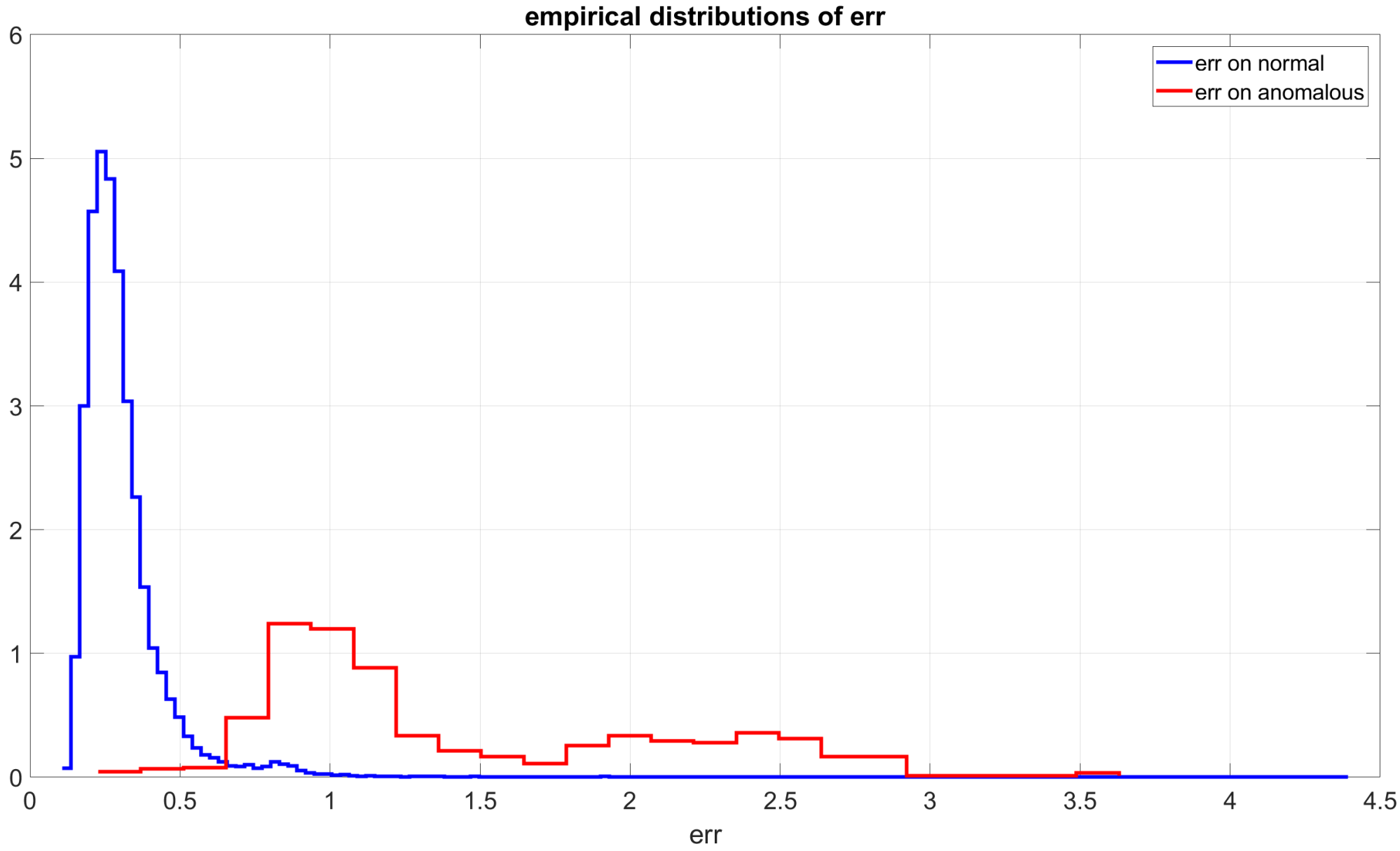
$$\text{err}(\boldsymbol{s}) = \left|\left|\boldsymbol{s} - \boldsymbol{s}PP^T\right|\right|_2$$

That can be used as an anomaly score



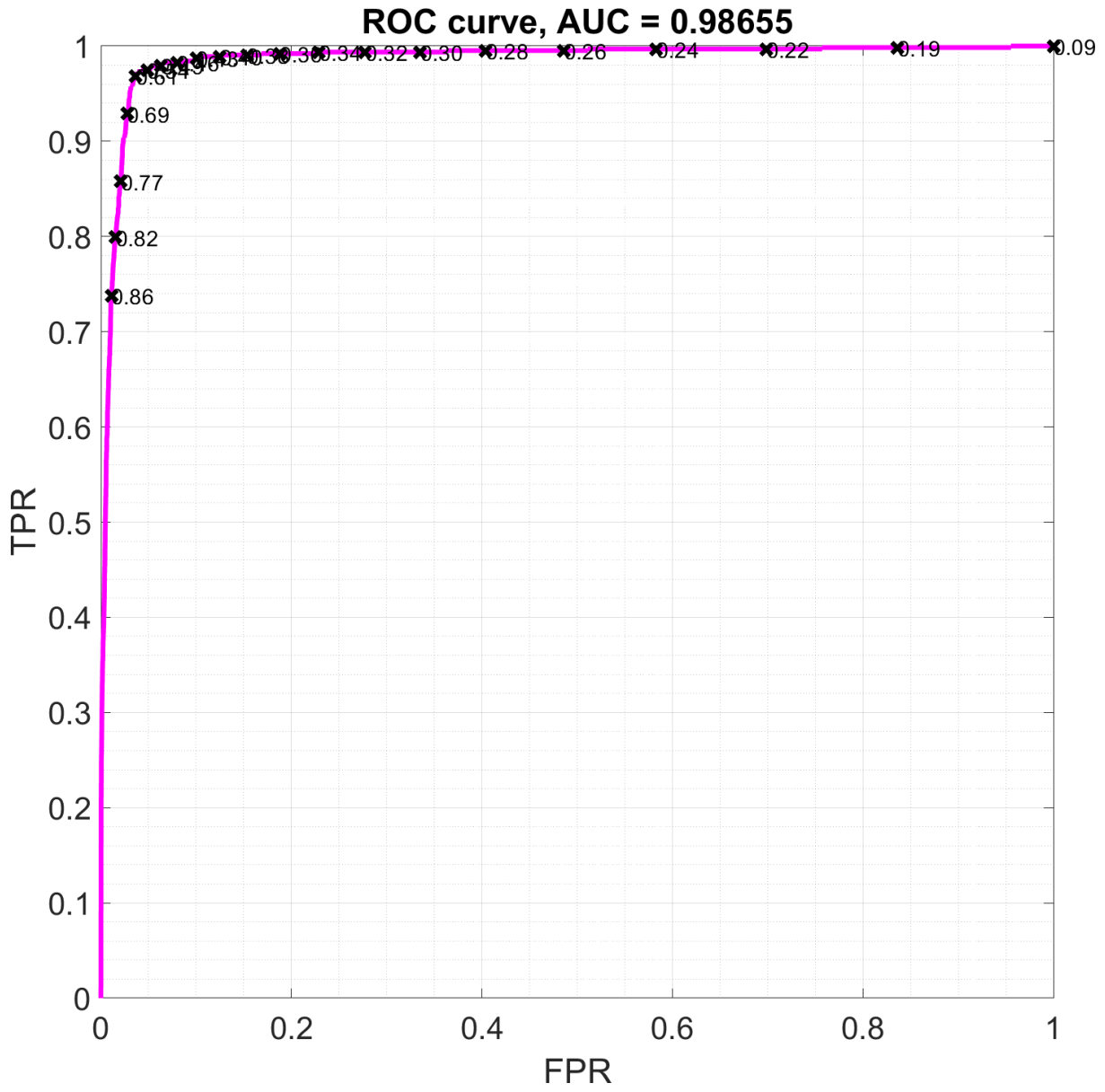


Different Distributions of $\mathrm{err}(\boldsymbol{s})$



Boracchi, Trovò

Good Detection by Thresholding



Anomaly Detection Based on Sparse Representations

Subspace Methods: Sparse Representations

Basic assumption: normal data live in **a union of low-dimensional subspaces** of the input space

The model learned from S is a matrix: the **dictionary** D.

Each signal is decomposed as **a sum of a few dictionary atoms** (representation is constrained to be **sparse**).

Atoms represent the many **building blocks** that can be used to reconstruct normal signals.

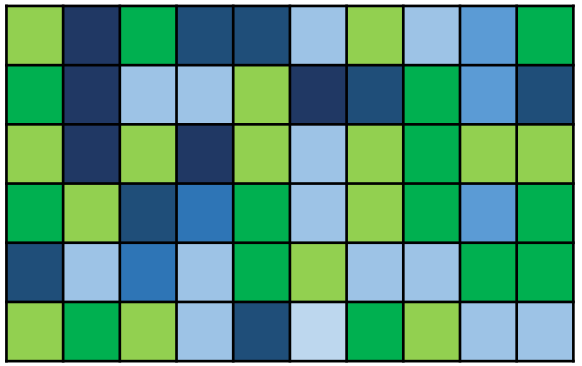
There are typically **more atoms** than the signal dimension.

Effective as long as the learned **dictionary** D is **very specific for normal data**

M. Elad *"Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing"*, Springer, 2010

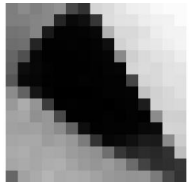
Dictionaries Yielding Sparse Representations

Dictionaries are just matrices! $D \in \mathbb{R}^{d \times m}$

$D =$ 

Dictionaries Yielding Sparse Representations

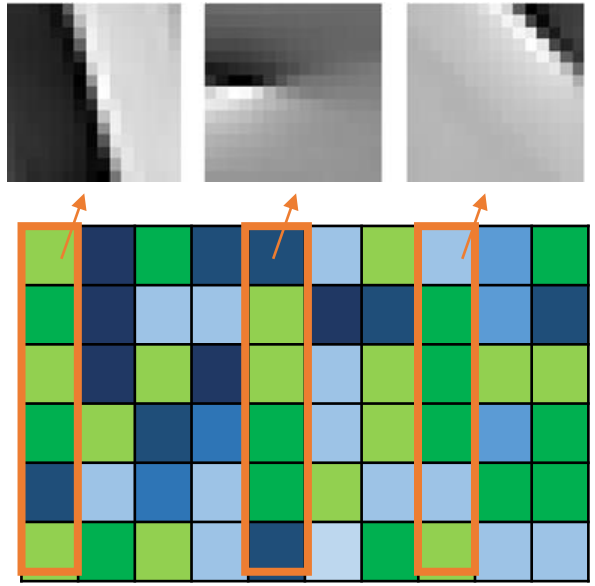
Dictionaries are just matrices! $D \in \mathbb{R}^{d \times m}$



s

Each column is an atom:

- lives in the input space
- it is one of the learned building blocks to reconstruct the input signal



$D =$

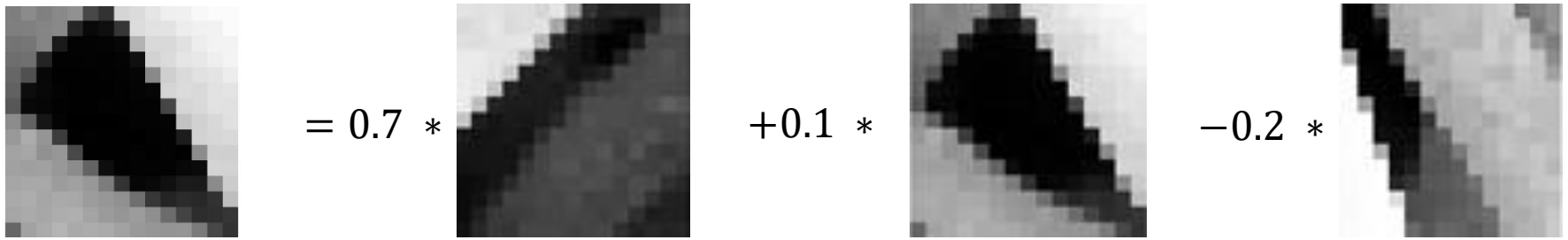
Sparse Representations

Let $\boldsymbol{s} \in \mathbb{R}^n$ be the input signal, a sparse representation is

$$\boldsymbol{s} = \sum_{i=1}^{M} \alpha_i \, \boldsymbol{d_i}$$

a linear combination of **few dictionary atoms** $\{\boldsymbol{d_i}\}$, i.e., most of coefficients are such that $\alpha_i = 0$

An illustrative example in case of our patches

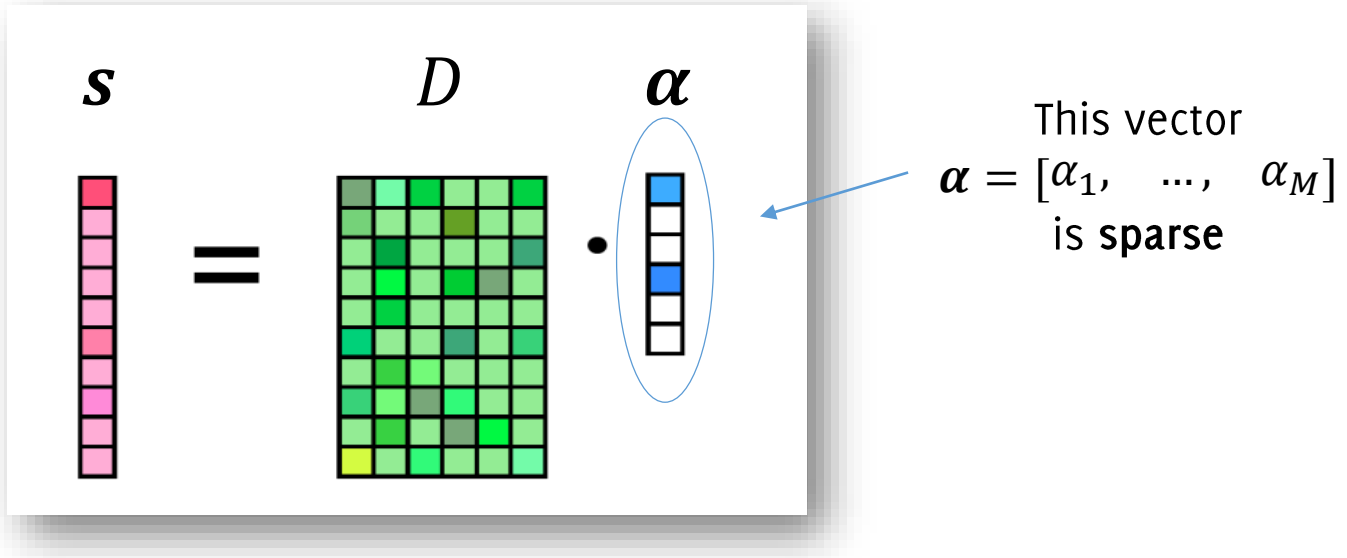


Sparse Representations… Matrix Expression

Let $\boldsymbol{s} \in \mathbb{R}^n$ be the input signal, a sparse representation is

$$\boldsymbol{s} = \sum_{i=1}^{M} \alpha_i \, \boldsymbol{d_i} = D\boldsymbol{\alpha}$$

a linear combination of **few dictionary atoms** $\{\boldsymbol{d_i}\}$ and $\|\boldsymbol{\alpha}\|_0 < L$, i.e. only a few coefficients are nonzero, i.e. $\boldsymbol{\alpha}$ is sparse.



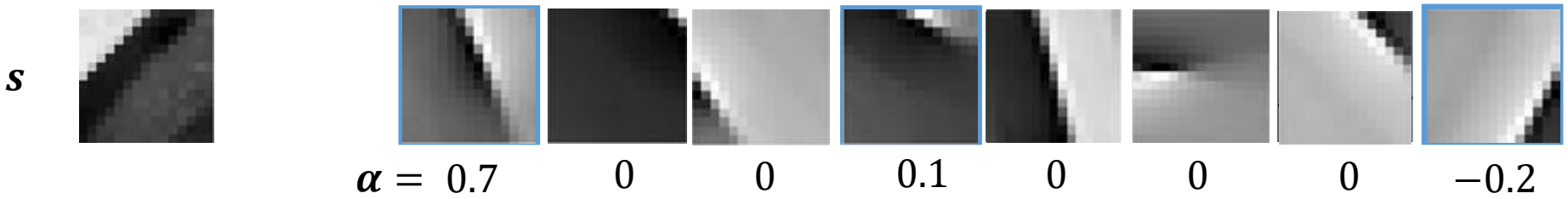
Sparse Coding…

Sprase Coding: computing the sparse representation for an input signal $\boldsymbol{s}$ w.r.t. D

$$\boldsymbol{s} \in \mathbb{R}^d \quad \Rightarrow \quad \boldsymbol{\alpha} \in \mathbb{R}^n$$

It is solved as the following optimization problem, (e.g. via the Orthogonal Matching Pursuit, OMP)

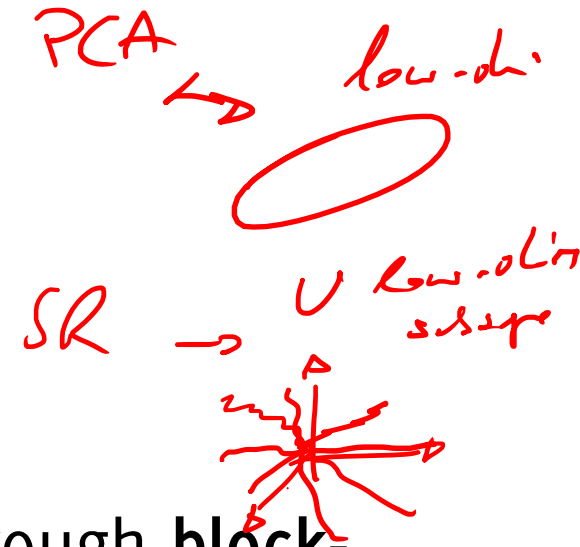
$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha} \in \mathbb{R}^n}{\operatorname{argmin}} \|D\boldsymbol{\alpha} - s\|_2 \quad \text{s.t.} \quad \|\boldsymbol{\alpha}\|_0 < L$$



In the previous illustration $\boldsymbol{\alpha} = [0.7, 0, 0, 0.1, 0, 0, 0, -0.2]$

Pati, Y.; Rezaiifar, R.; Krishnaprasad, P. *Orthogonal Matching Pursuit: recursive function approximation with application to wavelet decomposition*. Asilomar Conf. on Signals, Systems and Comput. 1993

… and Dictionary Learning

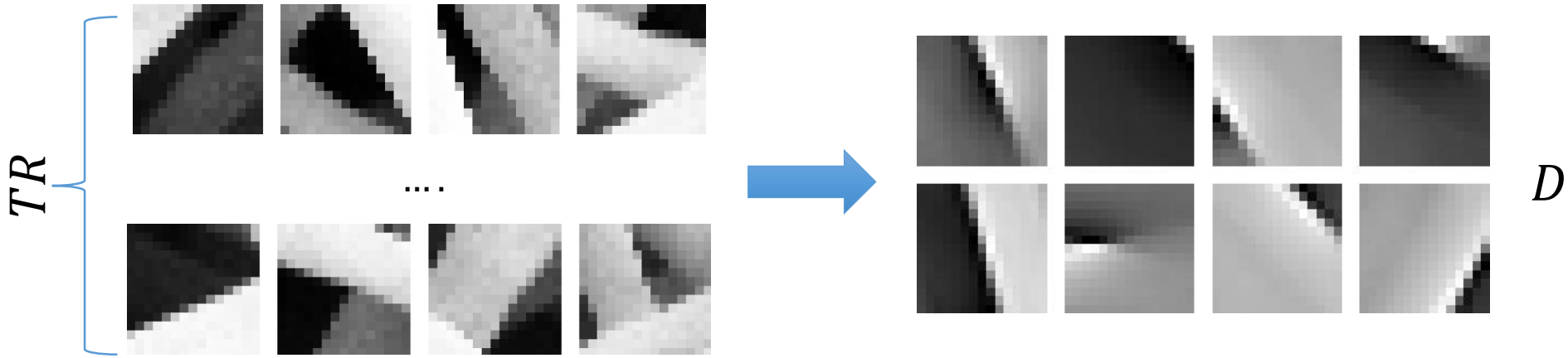


Dictionary Learning: estimate D from a training set of M

$$S = \{\boldsymbol{s_1}, \dots \boldsymbol{s_M}\} \in \mathbb{R}^{d \times n} \quad \Rightarrow \quad D \in \mathbb{R}^{d \times n}$$

It is solved as the following optimization problem typically through **block-coordinates descent** (e.g. KSVD algorithm)

$$[D, X] = \underset{A \in \mathbb{R}^{d \times n},\, Y \in \mathbb{R}^{n \times M}}{\operatorname{argmin}} \|AY - S\|_2 \ \text{s.t.} \ \|\boldsymbol{y_i}\|_0 < L, \quad \forall \boldsymbol{y_i}$$



Aharon, M.; Elad, M. Bruckstein, A. *K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation* IEEE TSP, 2006

Sparse representation monitoring: statistics

Anomalies can be directly **detected during the sparse coding** stage, by changing the functional being optimized.

A set of test signals is modeled as:

$$S = DX + E + V$$

where X is sparse, V is a noise term, and E is a matrix having most columns set to zero. Columns $\boldsymbol{e}_i \neq \mathbf{0}$ indicate anomalies, as they do not admit a sparse representation w.r.t. D

A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, *"Sparse coding with anomaly detection"* Journal of Signal Processing Systems, vol. 79, no. 2, pp. 179–188, 2015.

Sparse representation monitoring: statistics

Anomalies can be detected by solving (through ADMM) the following sparse coding problem

$$\underset{X,E}{\operatorname{argmin}} \left(\frac{1}{2} \|S - DX - E\|_F^2 + \lambda \|X\|_1 + \mu \|E\|_{2,1} \right)$$

Data-fidelity for normal data

Sparsity

Group sparsity regularization, only a few columns can be nonzero

.. and identifying as anomalies the signals corresponding to columns of E that are nonzero.

A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, *"Sparse coding with anomaly detection"* Journal of Signal Processing Systems, vol. 79, no. 2, pp. 179–188, 2015.

If you want to know more:
«*Mathematical Models and Methods for Image Processing*»
Spring 2023

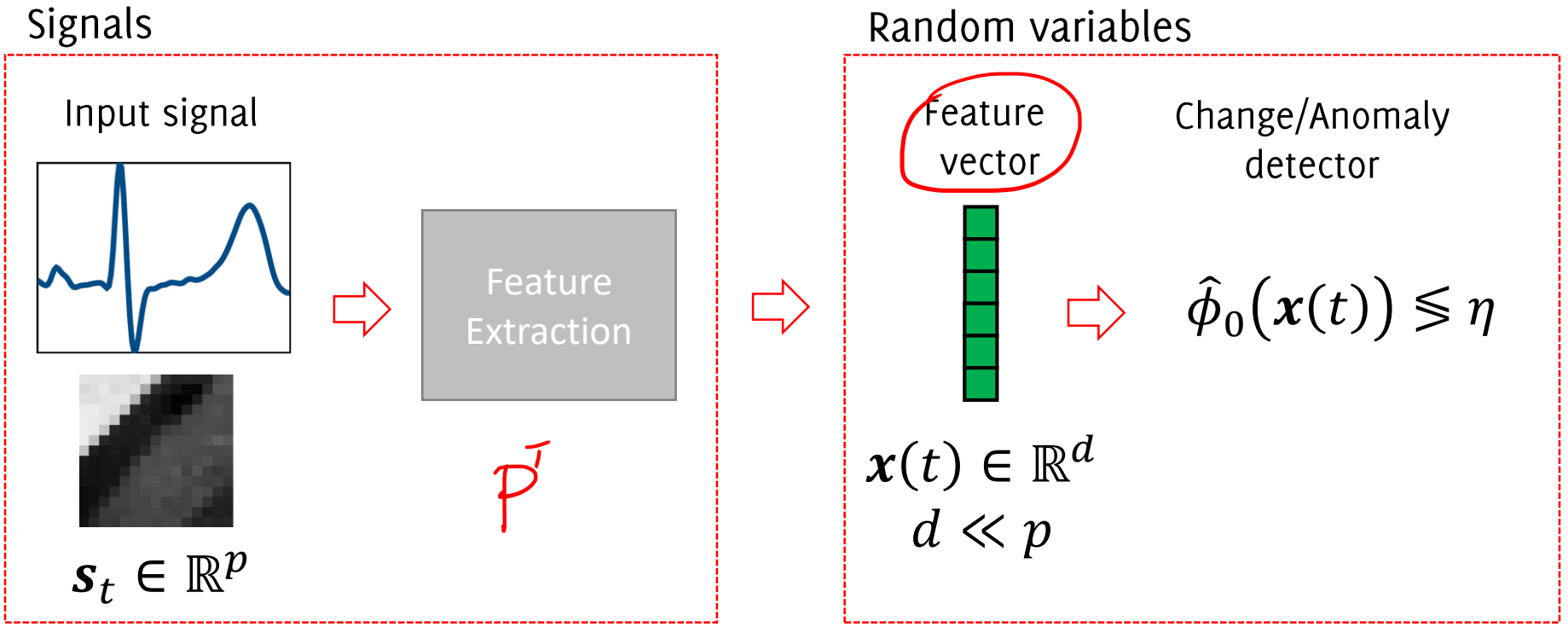
Semi-supervised AD methods out of the RVW

Out of the "Random Variable" world

- Detrending-based methods
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features

Monitoring Features

Feature extraction: meaningful indicators to be monitored which have a known / controlled response w.r.t. normal data



Feature Extraction: signal processing, a priori information, learning methods

The customary framework for change / anomaly detection

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

Feature Extraction

The peculiar structures of normal images and signals suggest that **normal data live in a manifold** having **lower dimension** than the input domain

Data dimensionality can be reduced by extracting features

Good features should:

- Yield a **stable response** w.r.t. **normal data**
- Yield **unusual response** on **anomalies** / when data change

Reconstruction error and representation coefficients can be considered features.

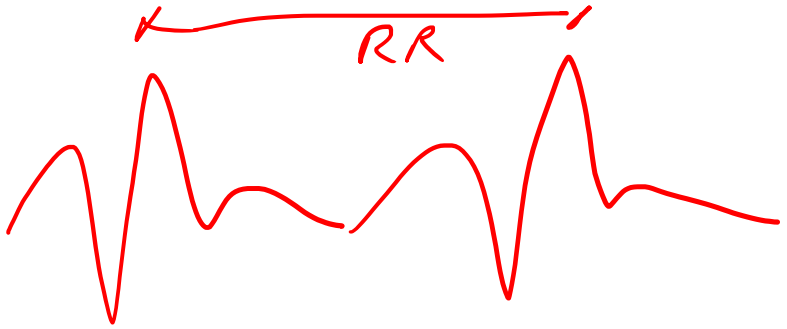
Features can be monitored in either one-shot/sequential monitoring schemes.

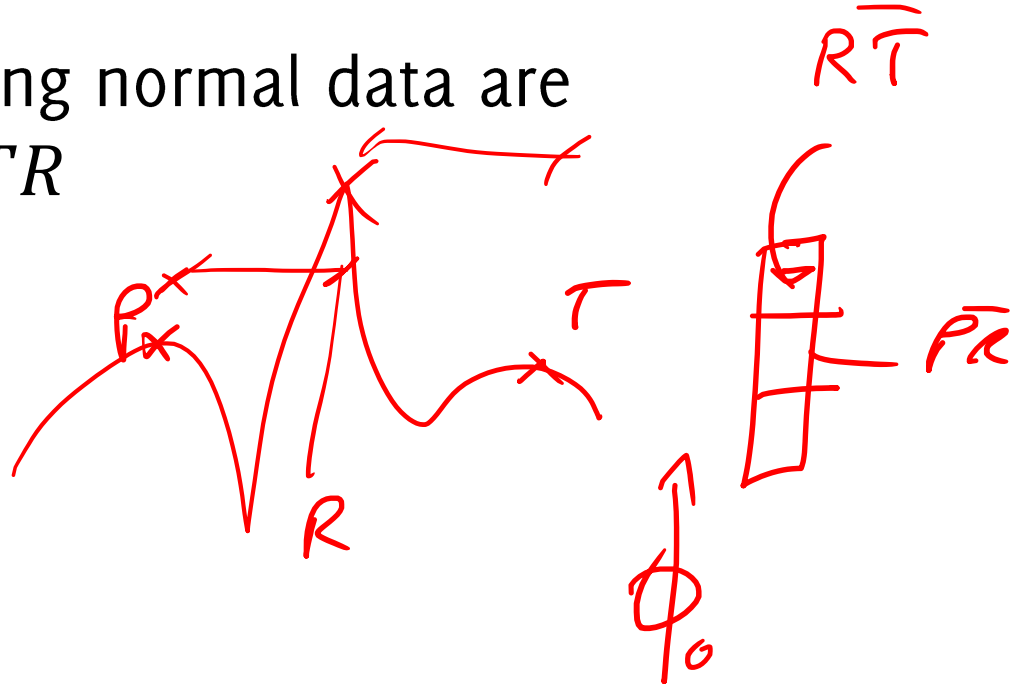
V. Chandola, A. Banerjee, V. Kumar. "*Anomaly detection: A survey*". ACM Comput. Surv. 41, 3, Article 15 (2009), 58 pages.

Feature Extraction approaches

There are two major approaches for extracting features:

- **Expert-driven (hand-crafted) features**: computational expressions that are **manually designed by experts** to distinguish between normal and anomalous data
- **Data-driven features**: **features** characterizing normal data are automatically **learned from training data** TR





Semi-supervised AD methods out of the RVW

Out of the "Random Variable" world

- Detrending-based methods
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features

Examples of Expert-Driven Features

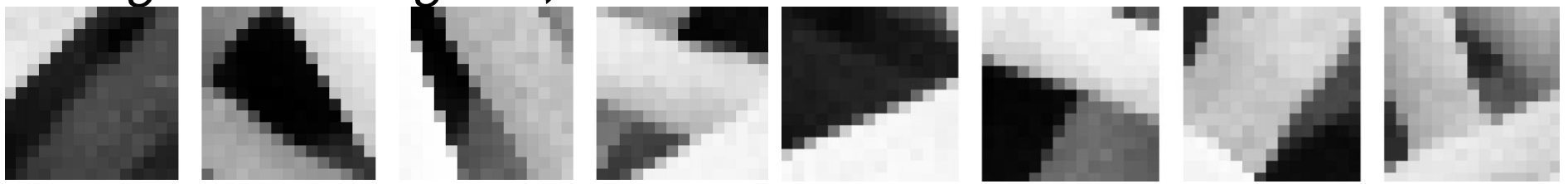
Expert-driven features: each patch of an image s

$$\mathbf{s}_c = \{s(c+u), u \in \mathcal{U}\}$$

Example of features are:

- the average,
- the variance,
- the total variation (the energy of gradients)

These can hopefully **distinguish normal** and **anomalous** patches (since image in anomalous region is expected to be flat or without edges characterizing normal regions)



Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017

Semi-supervised AD methods out of the RVW

Out of the "Random Variable" world

- Detrending-based methods
- Reconstruction-based methods
  - Subspace methods
- Feature-based monitoring
  - Expert-driven Features
  - Data-driven Features

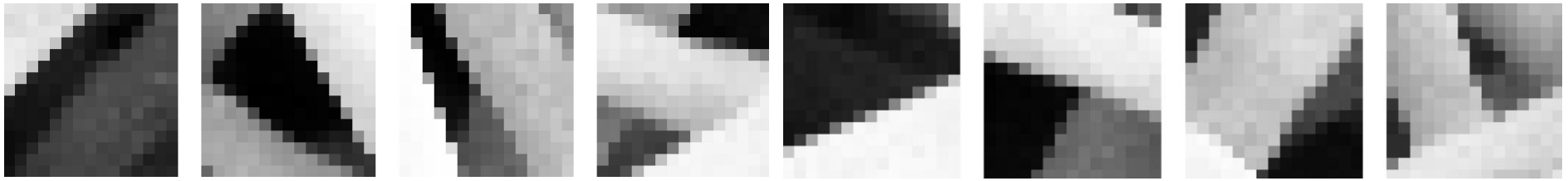
Examples of Data-Driven Features

Analyze each patch of an image s

$$\mathbf{s}_c = \{s(c+u), u \in \mathcal{U}\}$$

and determine whether it is normal or anomalous.

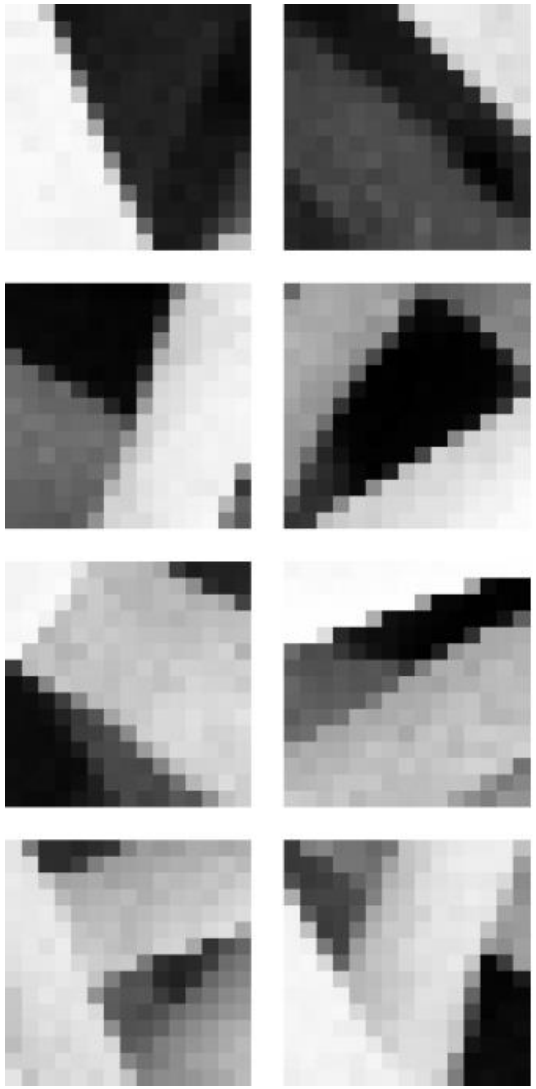
Data driven features: expressions to **quantitatively assess whether test patches conform** or not **with the model**, learned from normal data.



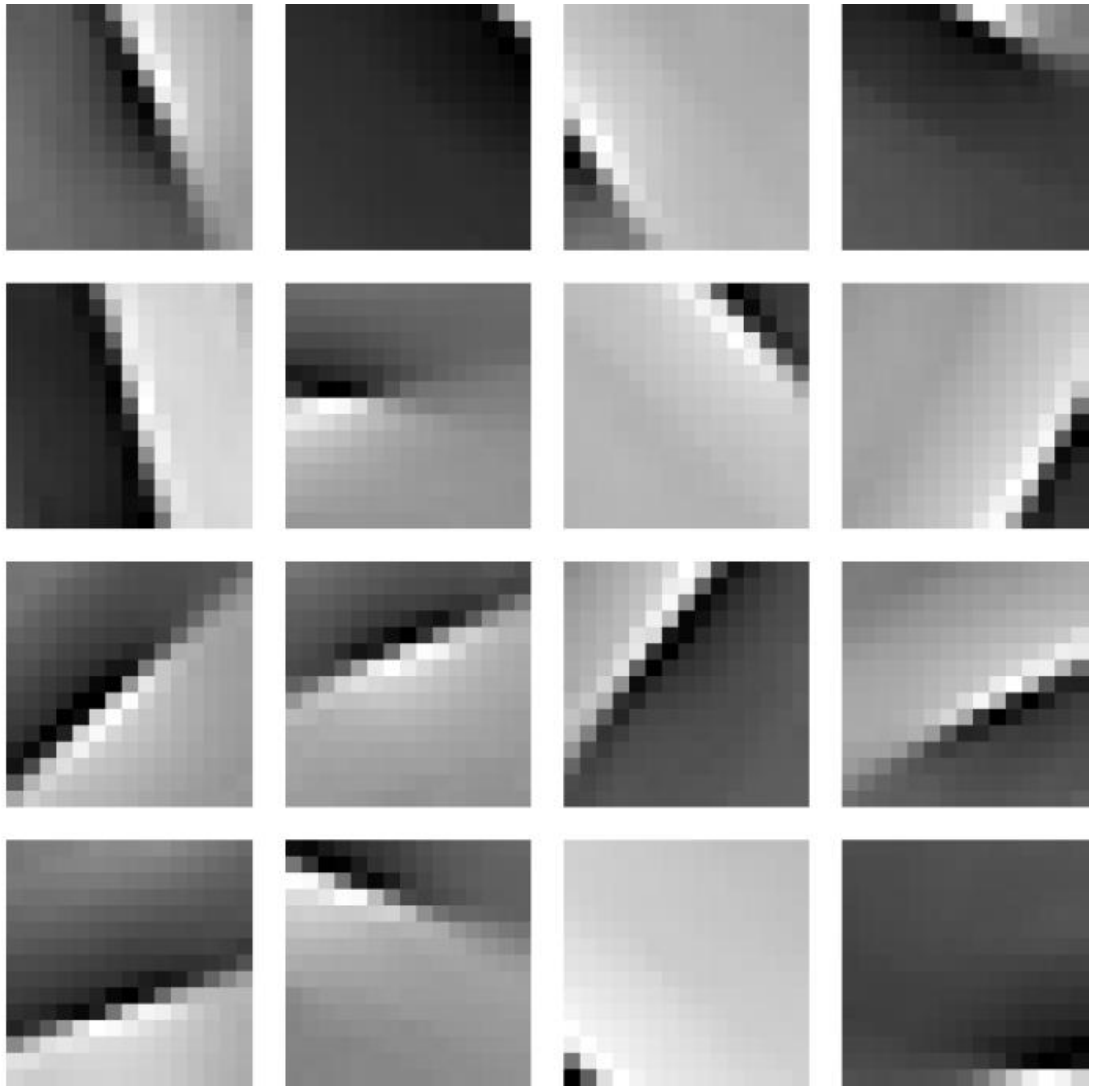
Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

A Learned Dictionary from normal patches

Example of training patches



Few learned atoms (BPDN-based learning)



Data-Driven Features

To assess the conformance of $\boldsymbol{s}_c$ with D we solve the following

Sparse coding:

$$\boldsymbol{\alpha} = \underset{\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^n}{\operatorname{argmin}} \|D\tilde{\boldsymbol{\alpha}} - \mathbf{s}\|_2^2 + \lambda \|\tilde{\boldsymbol{\alpha}}\|_1, \qquad \lambda > 0$$

which is the BPDN formulation and we solve using ADMM.

The penalized ℓ^1 formulation has more degrees of freedom in the reconstruction, **the conformance of $\boldsymbol{s}$ with $\boldsymbol{D}$ have to be assessed monitoring both terms of the functional**

Boyd S., Parikh N., Chu E., Peleato B., Eckstein J., *"Distributed optimization and statistical learning via the alternating direction method of multipliers"* 2011

Data-driven features

Features then include both the **reconstruction error**

$$\mathrm{err}(\boldsymbol{s}) = \|D\boldsymbol{\alpha} - \mathbf{s}\|_2^2$$

and **the sparsity** of the representation

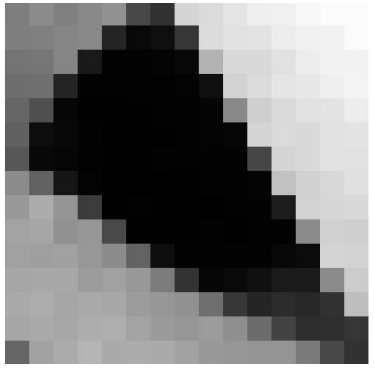
$$\|\boldsymbol{\alpha}\|_1$$

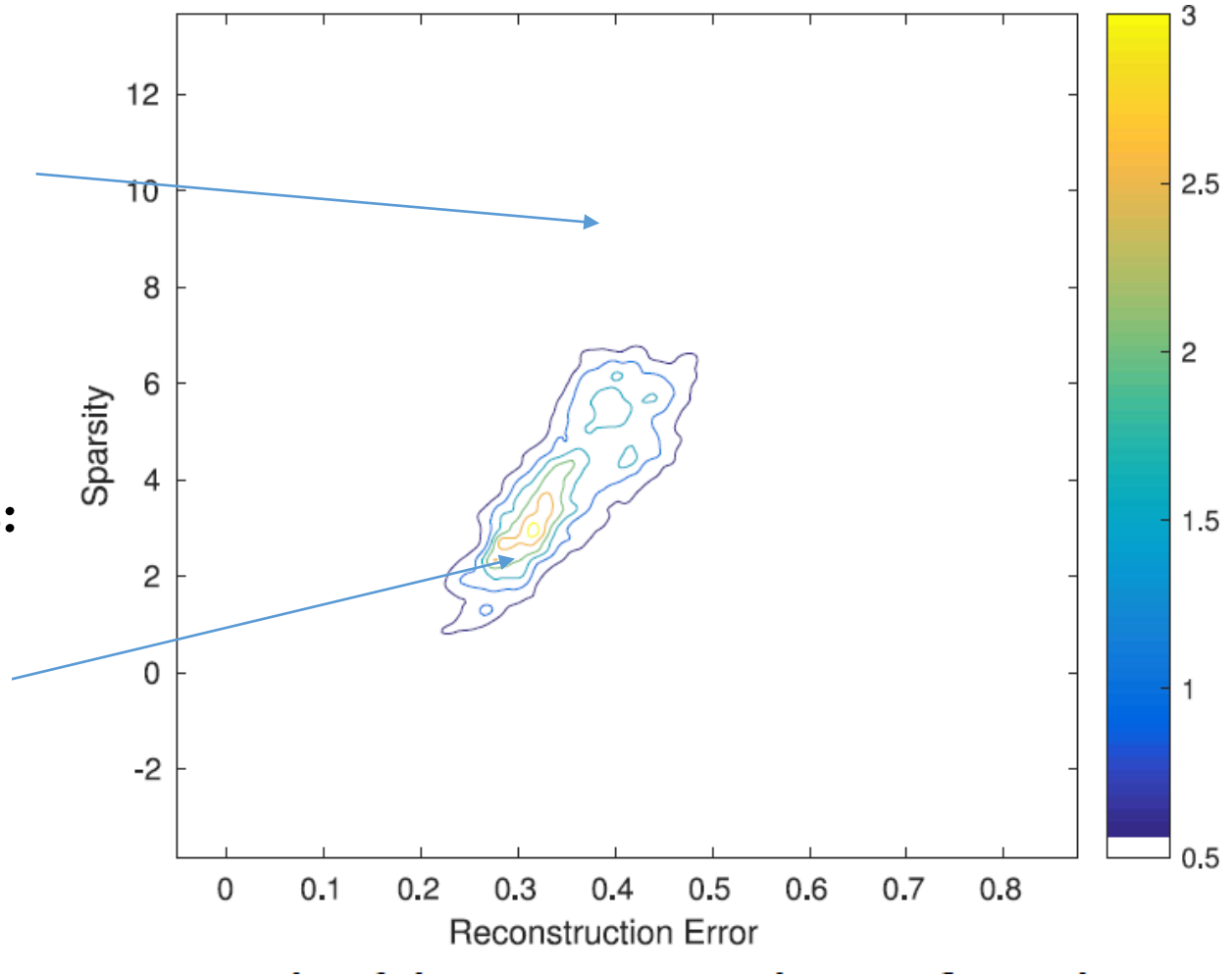
Thus obtaining **a data-driven feature vector** $\boldsymbol{x} = \begin{bmatrix} \|D\boldsymbol{\alpha} - \mathbf{s}\|_2^2 \\ \|\boldsymbol{\alpha}\|_1 \end{bmatrix}$

Density-based monitoring on Data-driven features

Anomalies

Normal patches:





Data-driven features

Training:

- **Learn** from $TR\backslash V$ the dictionary D
- **Learn** from V, the distribution $\hat{\phi}_0$ of normal features vectors $\boldsymbol{x}$.

Testing:

- Compute feature vectors $\boldsymbol{x}$ via sparse coding
- Detect anomalies when $\hat{\phi}_0(\boldsymbol{x}) < \eta$

Carrera D., Manganini F., Boracchi G., Lanzarone E. *"Defect Detection in SEM Images of Nanofibrous Materials"*, IEEE Transactions on Industrial Informatics 2017, 11 pages, doi:10.1109/TII.2016.2641472

Data-driven features

Training:

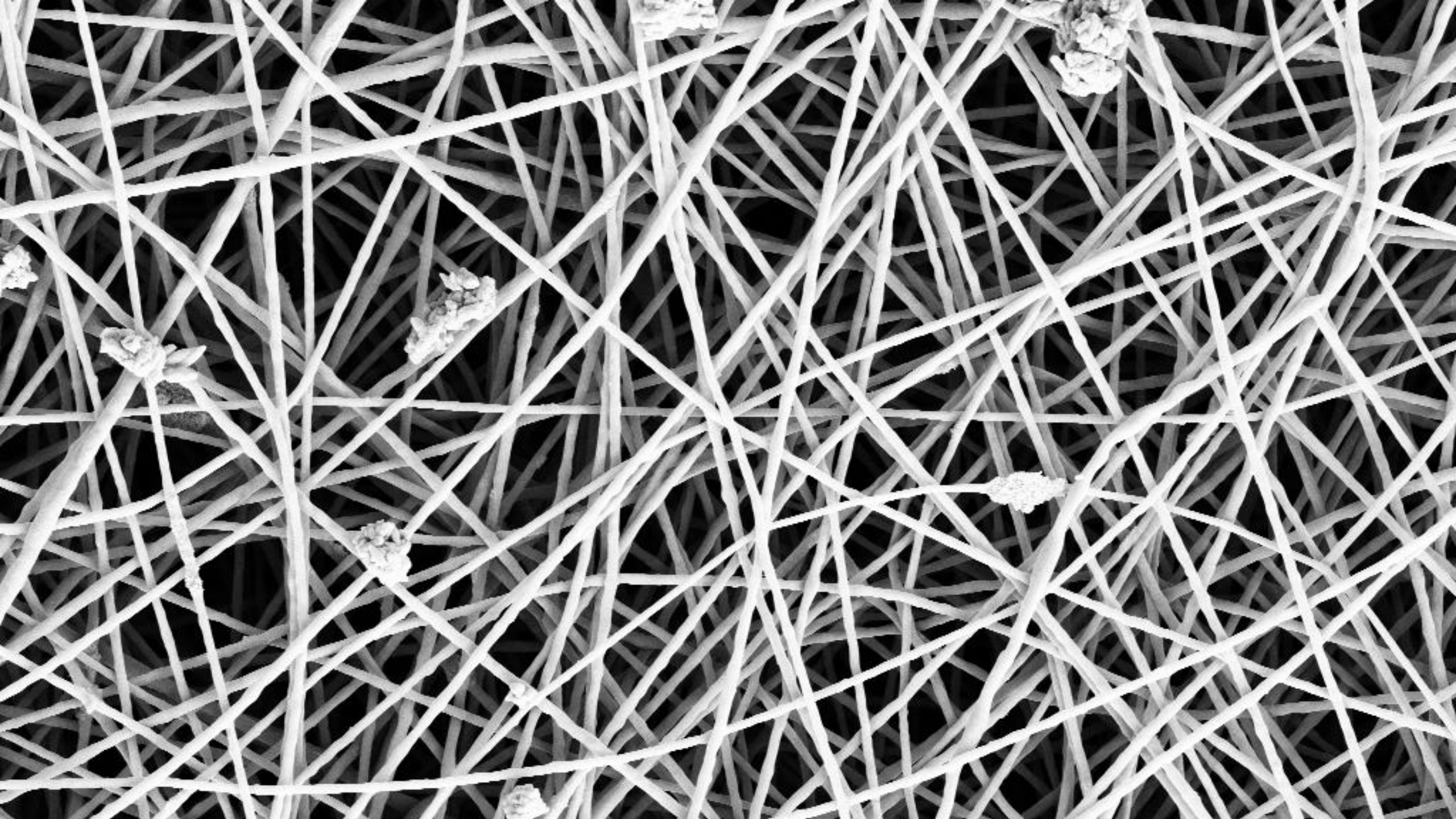
- **Learn** from $TR\backslash V$ the dictionary D
- **Learn** from V, the distribution $\hat{\phi}_0$ of normal features vectors $\boldsymbol{x}$.

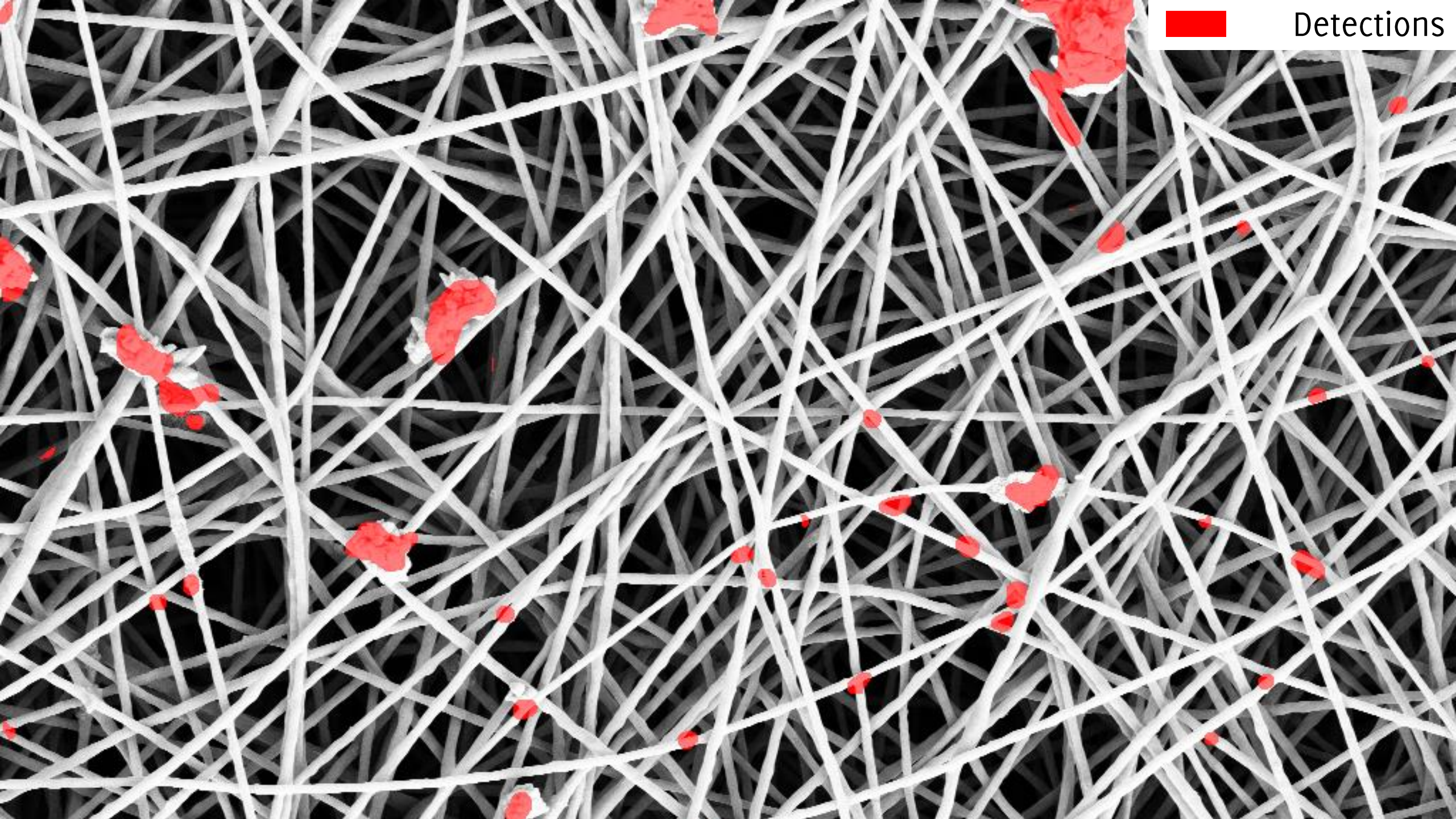
Testing:

- Compute feature vectors $\boldsymbol{x}$ via sparse coding
- Detect anomalies when $\hat{\phi}_0(\boldsymbol{x}) < \eta$

This solution is rather flexible and can be adapted when operating conditions changes (e.g. different zooming level)

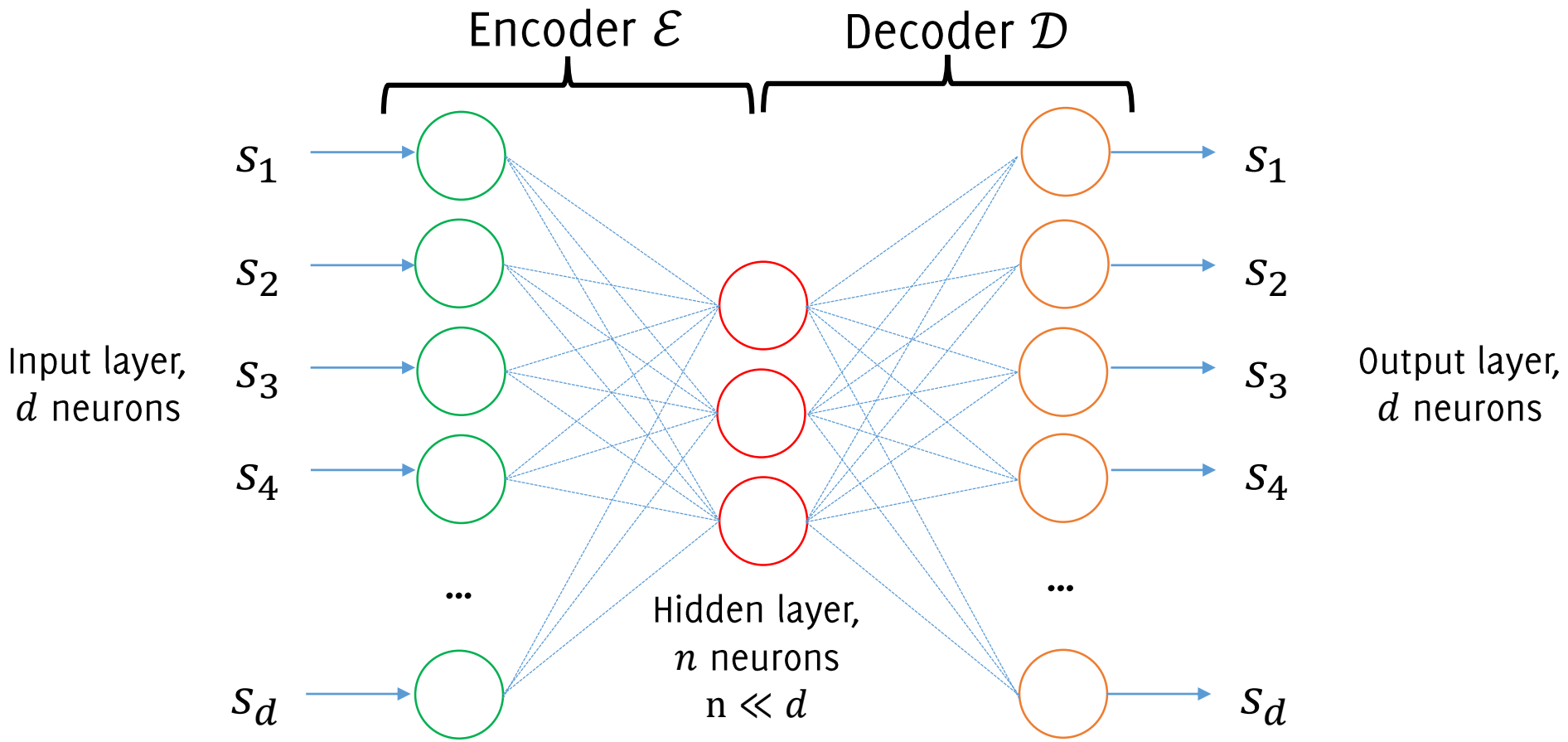
Carrera D., Boracchi G., Foi A., Wohlberg B. *"Scale-invariant Anomaly Detection With multiscale Group-sparse Models"* ICIP 2016




Detections

Feature-based Methods

Autoencoders can be also used in feature-based monitoring schemes, where the hidden representation of the input is the feature being monitored



Monitoring Feature Distribution

Detection by **feature monitoring** (AE notation)

Training (Monitoring Feature Distribution):

- Learn the autoencoder $\mathcal{D}(\mathcal{E}(\cdot))$ from the training set S
- Fit a density model $\hat{\phi}_0$ to the encoded features

$$\{\mathcal{E}(\boldsymbol{s}), \boldsymbol{s} \in V\}$$

  over a validation set $V \neq S$

- Define a suitable threshold γ for $\hat{\phi}_0(\boldsymbol{s})$

Testing (Monitoring Feature Distribution):

- Encode each incoming signal $\boldsymbol{s}$ through $\mathcal{E}$
- Detect anomalies if $\hat{\phi}_0(\mathcal{E}(\boldsymbol{s})) < \gamma$

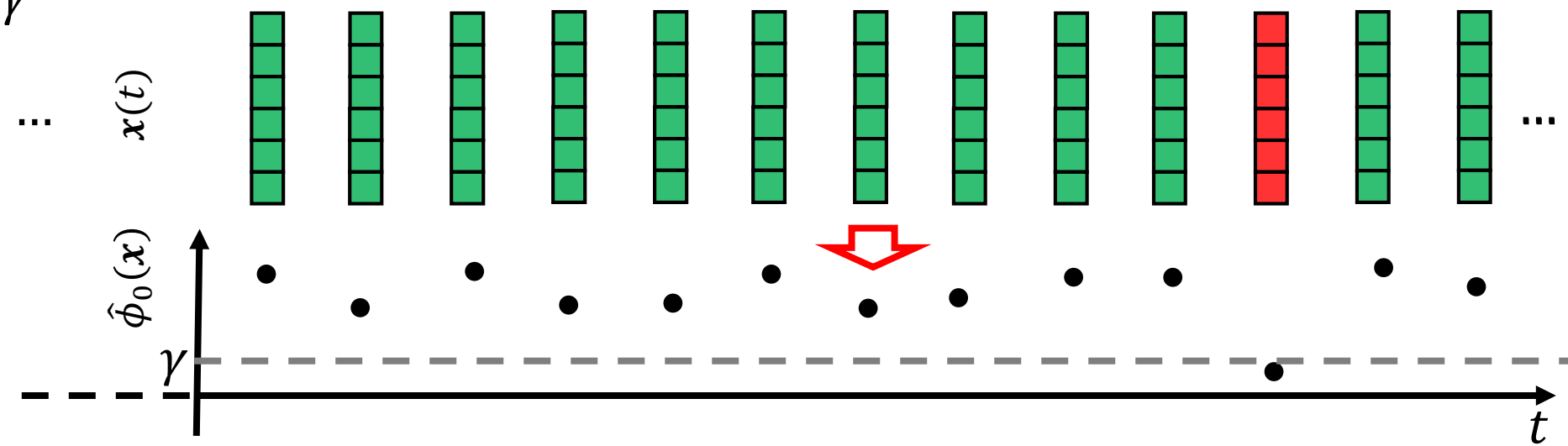
Monitoring Feature Distribution

Normal data are expected to yield $\mathcal{E}(\boldsymbol{s})$ that are **i.i.d. vectors (or features)** and that **follow an** unknown **distribution** ϕ_0.

Anomalous data do not, as they follow $\phi_1 \neq \phi_0$.

We are **back to our statistical framework** and we can

- learn $\hat{\phi}_0$ from a set features extracted from normal data
- detect anomalous data by computing $\boldsymbol{x} = \mathcal{E}(\boldsymbol{s})$ and then check whether $\hat{\phi}_0(\boldsymbol{x}) < \gamma$



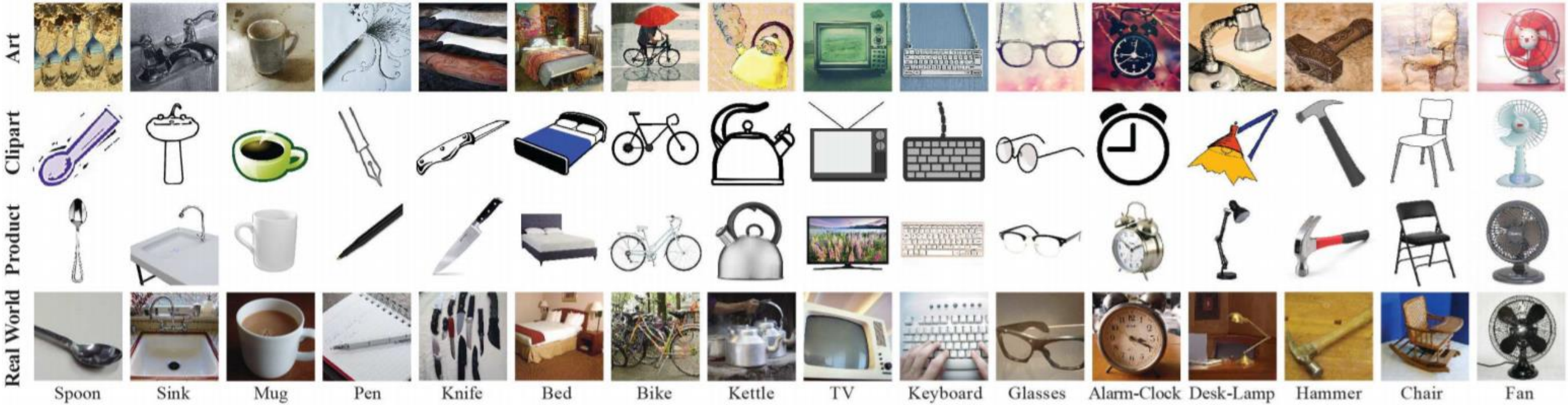
Domain Adaptation

Example of Domain Adaptation Questions

These are rather **common questions** when using an classifier for images

- Can we train classifiers with **Flickr photos**, as they have already been collected and annotated, and hope the classifiers still work well on **mobile camera images**?
- Image classifiers optimized on benchmark dataset often exhibit significant degradation in recognition accuracy when evaluated on another one

Domain Adaptation Datasets



Venkateswara, H., Eusebio, J., Chakraborty, S., & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. CVPR 2017

Problem Formulation

Consider a classification problem from an input space to a label space

$$\mathcal{X} \rightarrow \mathcal{Y}$$

and denote by

- A **domain** $\mathcal{D} = \{\mathcal{X}, \phi_{\boldsymbol{x}}\}$ such that $\boldsymbol{x} \sim \phi_{\boldsymbol{x}}$
- A **task** $\mathcal{T} = \{\mathcal{Y}, \phi(\cdot\,|\boldsymbol{x})\}$ which consists in associating the label to an input $\boldsymbol{x}$

S. J. Pan and Q. Yang, "A survey on transfer learning", TKDE 2010.

Problem Formulation

Definition: Transfer Learning

Given

a ***source domain*** $\mathcal{D}_S = \{\mathcal{X}_S, \phi_S\}$ *and learning task* $\mathcal{T}_S = \{\mathcal{Y}_S, \phi_S(\cdot\,|\boldsymbol{x})\}$,
and ***target domain*** $\mathcal{D}_T = \{\mathcal{X}_T, \phi_T\}$ *and learning task* $\mathcal{T}_T = \{\mathcal{Y}_T, \phi_T(\cdot\,|\boldsymbol{x})\}$,

Where $\mathcal{D}_S \neq \mathcal{D}_T$ *or* $\mathcal{T}_S \neq \mathcal{T}_T$ *(either or both)*

Transfer learning aims to ***improve the learning of the target predictive function*** $\boldsymbol{K}$ ***solving*** $\mathcal{T}_T$ ***in*** $\mathcal{D}_T$ ***using the knowledge in*** $\mathcal{D}_S$ ***and*** $\mathcal{T}_S$,

S. J. Pan and Q. Yang, "A survey on transfer learning", TKDE 2010.

Problem Formulation

Remarks:

$\mathcal{D}_S \neq \mathcal{D}_T$ implies that either

- $\mathcal{X}_S \neq \mathcal{X}_T$ (e.g. different input size / features)
- $\phi_{\boldsymbol{x}}^S \neq \phi_{\boldsymbol{x}}^T$ (e.g. different style of an image)

$\mathcal{T}_S \neq \mathcal{T}_T$ implies that either

- $\mathcal{Y}_S \neq \mathcal{Y}_T$ (e.g. new labels in target)
- $\phi^S(y|\boldsymbol{x}) \neq \phi^T(y|\boldsymbol{x})$ (e.g. different class imbalance between S and T)

Obviously, $\mathcal{D}_S = \mathcal{D}_T$ and $\mathcal{T}_S = \mathcal{T}_T$ corresponds to traditional machine learning settings

Example: $\mathcal{X}_S \neq \mathcal{X}_T$ same task, different domains

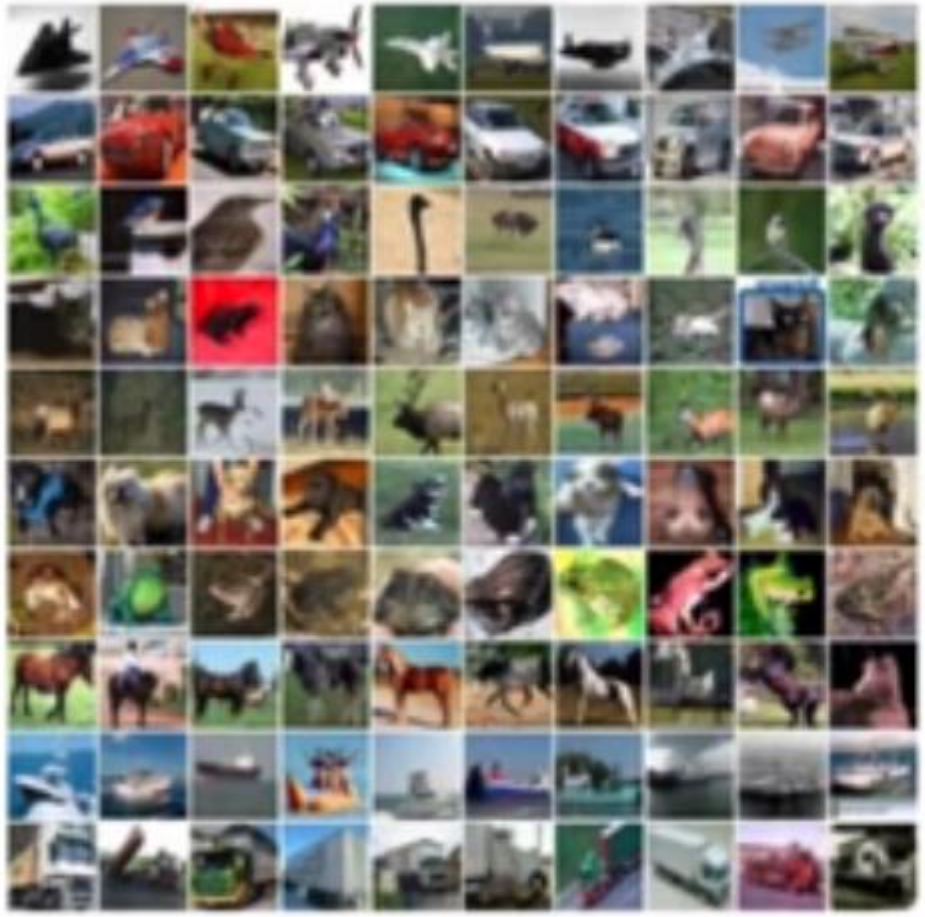
- Classification of documents in different languages
- Images of different sizes
- Colour vs grayscale images

Example: $\phi_x^S \neq \phi_x^T$ and $\mathcal{X}_S = \mathcal{X}_T$

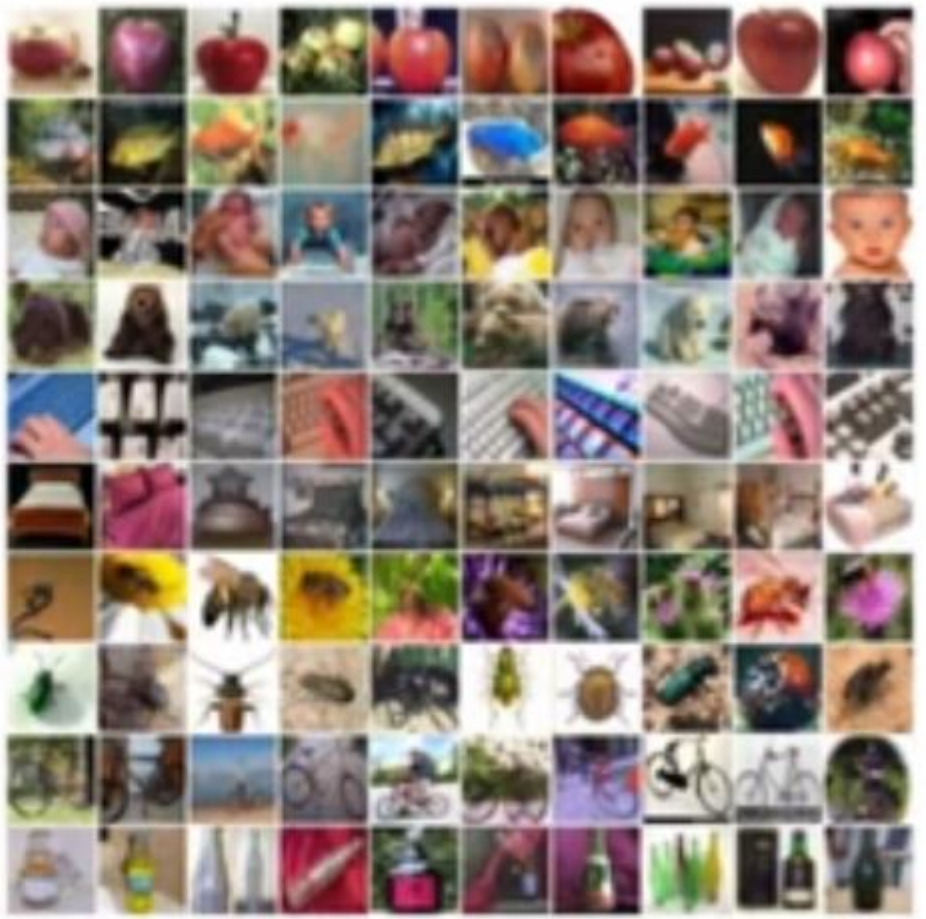
Images have the same sizes, same labels (and tasks) but different distributions

Venkateswara, H., Eusebio, J., Chakraborty, S., & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. CVPR 2017

Example: $\mathcal{Y}_S \neq \mathcal{Y}_T$ different task, same domains



(b) CIFAR-10



(c) CIFAR-100

Example: $\phi^S(y|\boldsymbol{x}) \neq \phi^T(y|\boldsymbol{x})$

Distribution changes are ubiquitous!

In the case of text classification, words might have a different meaning depending on the domain

Consider

$\mathcal{D}_S$: articles from newspapers

$\mathcal{D}_T$: articles from a computer magazine

The word «monitor» in the two context is very likely to have different meanings, leading to different classes of documents in each domain

Standard Settings

Consider a classification problem where we are provided with

$$TR = \{(\boldsymbol{x}_0, y_0), \dots, (\boldsymbol{x}_n, y_n) \in \mathcal{X}_S \times \mathcal{Y}_S\}$$

That are from the source domain $\mathcal{D}_S$ and representative of $\mathcal{T}_S$.

You are asked to prepare a classifier K that is able to operate in the target domain $\mathcal{D}_T$ to address the task $\mathcal{T}_T$

Usually there are additional constraints over the target domain (task), such that little or even no supervised information is provided.

S. J. Pan and Q. Yang, "A survey on transfer learning", TKDE 2010.

Supervised Case

The fully supervised case:

- we have access to

$$TR_S = \{(\boldsymbol{x}_0, y_0), \ldots, (\boldsymbol{x}_n, y_n) \in \mathcal{X}_S \times \mathcal{Y}_S\}$$

a large, annotated corpus of data from the **source domain $\mathcal{D}_S$ and representative of $\mathcal{T}_S$.**

- we spend a little money to annotate a small corpus in the target domain **$\mathcal{D}_T$ and representative of $\mathcal{T}_T$**

$$TR_T = \{(\boldsymbol{x}_0, y_0), \ldots, (\boldsymbol{x}_m, y_m) \in \mathcal{X}_T \times \mathcal{Y}_T\}$$

Thus $m \ll n$.

Goal: learn a classifier K that is very accurate in $\mathcal{D}_T$ to solve the task $\mathcal{T}_T$

Unsupervised Case

The fully unsupervised case:

- we have access to

$$TR_S = \{(\boldsymbol{x}_0, y_0), \dots, (\boldsymbol{x}_n, y_n) \in \mathcal{X}_S \times \mathcal{Y}_S\}$$

a large, annotated corpus of data from the **source domain $\mathcal{D}_S$ and representative of $\mathcal{T}_S$.**

- we have no annotations over $\mathcal{D}_T$

$$X_T = \{\boldsymbol{x_0}, \dots, \boldsymbol{x_m} \in \mathcal{X}_T\}$$

And m might be even larger than n.

Goal: learn a classifier K that is very accurate in $\mathcal{D}_T$ to solve the task $\mathcal{T}_T$

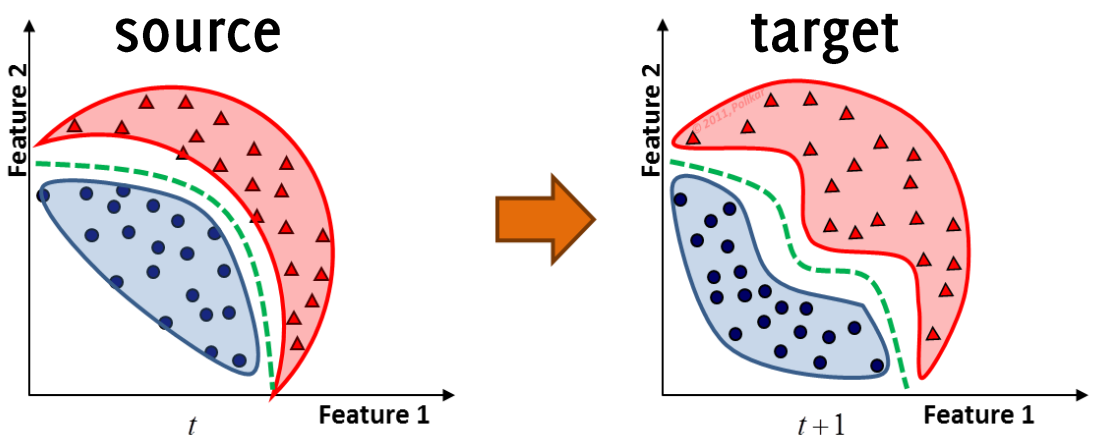
Does this remind you something?

Connection with Concept Drift

It's the **same problem to be addressed in Datastreams affected by Concept Drift, but without**

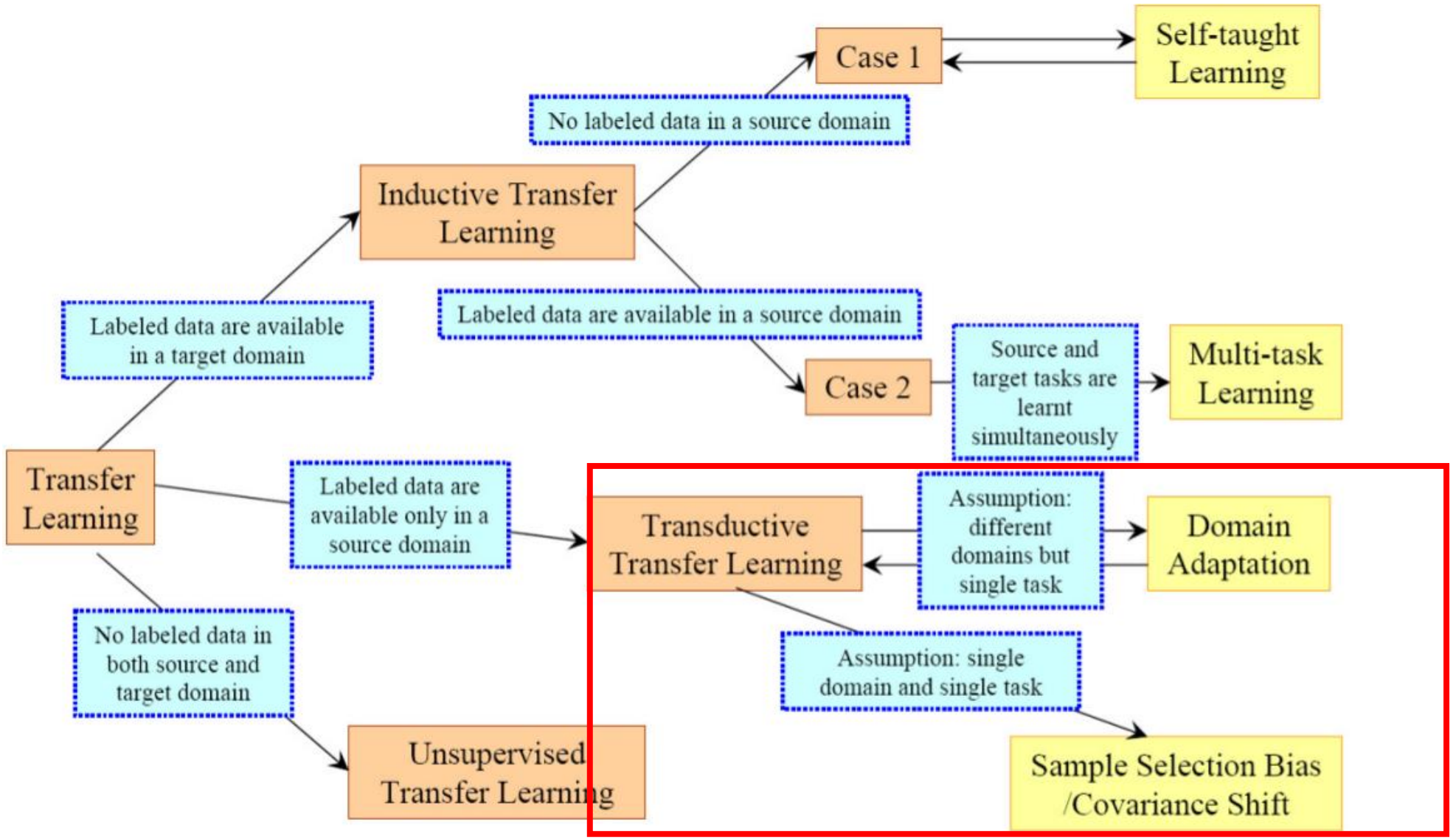
- Temporal dimension
- Need to detect changes

You are given a training set that is (stationary) from the source domain and you have to operate in a different (stationary) target domain



Major Approaches to Domain Adaptation

Transfer Learning Taxhonomy



S. J. Pan and Q. Yang, "A survey on transfer learning", TKDE 2010.

Reweighting

Correct a sample bias by reweighting source labeled data.

The rationale:

source instances close to target instances are more important

Motivations:

- Domains share the same support (i.e. bag of words)
- Distribution shift is caused by sampling bias/shift between marginals

Idea:

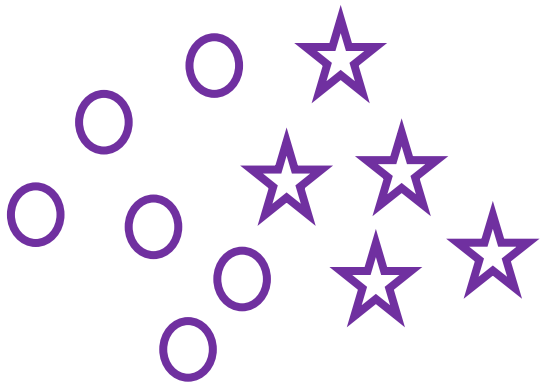
- Reweight or select instances to reduce the discrepancy between source and target domains

Supervised Reweighting

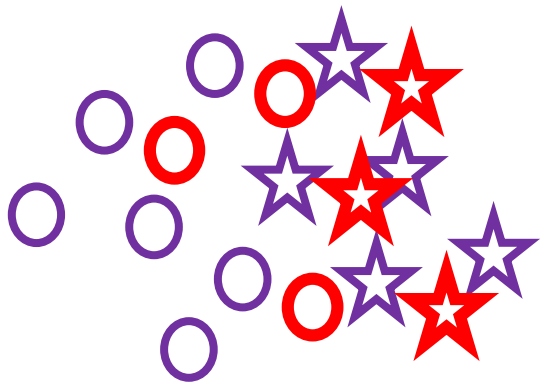
Reweight or select instances to reduce the discrepancy between source and target domains

Supervised case: (where $\mathcal{Y}_S = \mathcal{Y}_T$ and $\mathcal{X}_S = \mathcal{X}_T$), it is possible to train a classifier over $TR_S \cup TR_T$ by weighting more the loss over instances from TR_T (or by resampling TR_T)

TR_S



$TR_S \cup TR_T$



Unsupervised Reweighting over TR_S

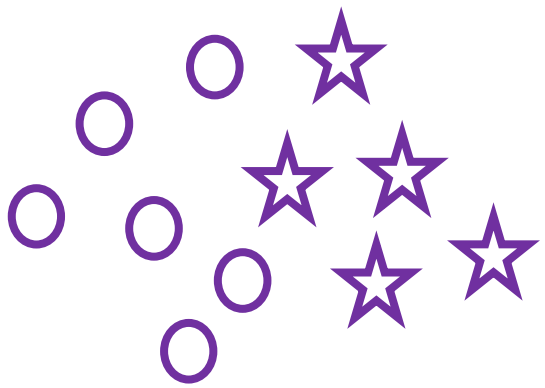
Estimate $\hat{\phi}_{S,\boldsymbol{x}}$ and $\hat{\phi}_{T,\boldsymbol{x}}$ from $\mathcal{X}_S$ and $\mathcal{X}_T$ (no labels needed) and compute

$$w_i = \frac{\hat{\phi}_{T,\boldsymbol{x}}(x_i)}{\hat{\phi}_{S,\boldsymbol{x}}(x_i)}$$

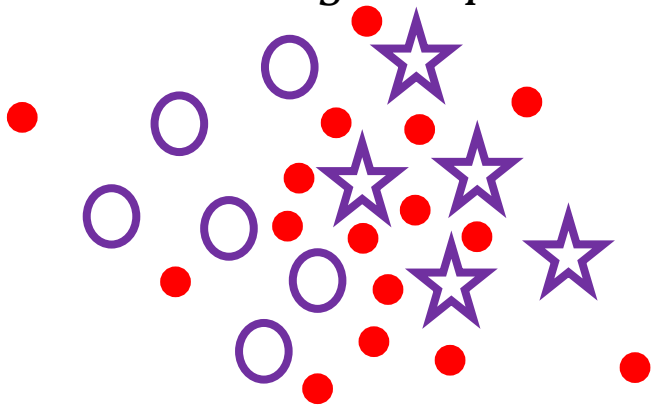
Train a classifier to minimize the weighted loss

$$\mathcal{L}(TR_S) = \sum_{x_i \in TR_S} w_i \, I[K(\boldsymbol{x_i}) \neq y_i]$$

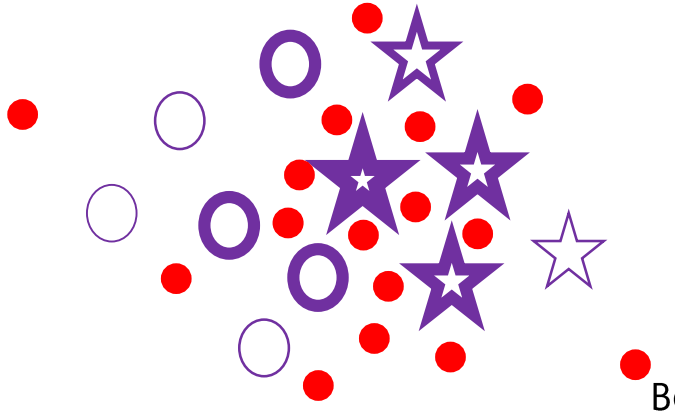
TR_S



$TR_S \cup X_T$



$TR_S \cup X_T$ and reweighting



Feature-based Methods

Find a common space where source and target are close (projection, new features, etc)

Change the feature representation $\mathcal{X}$ to better represent shared characteristics between the two domains

- some features are domain-specific,
- others are generalizable
- or there exist mappings from the original space

Idea:

- Make source and target domains explicitly similar
- Learn a new feature space by embedding or projection

Supervised, Feature-based Methods

Train a classifier K_S over TR_S

Use the output of K_S as an additional feature to train the classifier K_T over augmented features

$$[x_i;\ K_S(x_i)]$$

Or, train a joint classifier over an augmented training set TR_A where inputs are defined as:

$$\boldsymbol{x_i} \rightarrow [\boldsymbol{x_i}; \boldsymbol{x_i}; \mathbf{0}] \text{ when } \boldsymbol{x_i} \text{ from } TR_S$$
$$\boldsymbol{x_i} \rightarrow [\boldsymbol{x_i}; \mathbf{0}; \boldsymbol{x_i}] \text{ when } \boldsymbol{x_i} \text{ from } TR_T$$

During operations augment everything as in TR_T

Daumé III, H. Frustratingly easy domain adaptation. ACL 2007

Domain Adaptation by CORrelation ALignmetn

CORAL minimizes domain shift by aligning the second-order statistics of of the distributions of features from $\mathcal{X}_S$ and $\mathcal{X}_T$

- Does not require any target label
- Very simple to implement

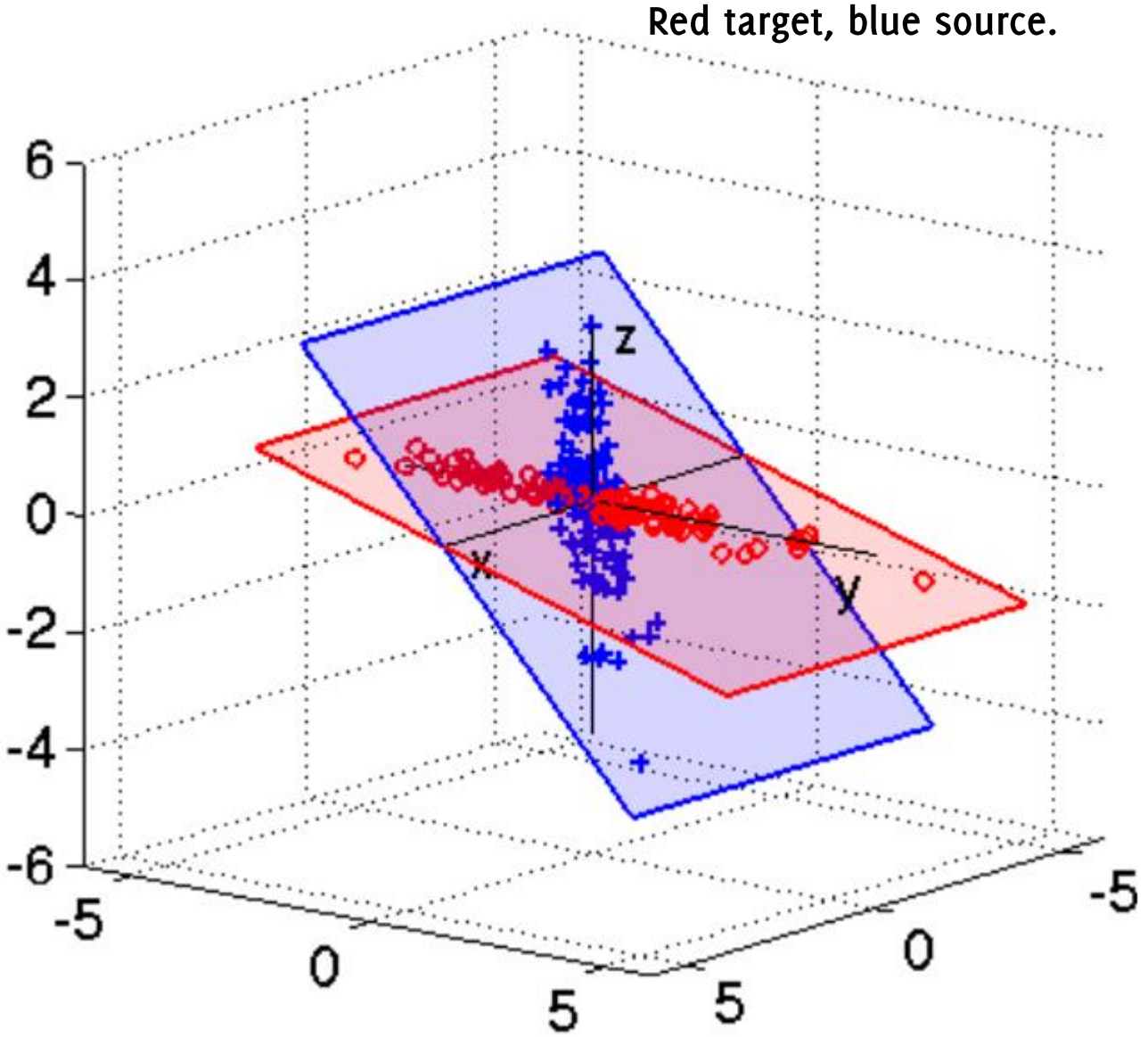
Algorithm 1 CORAL for Unsupervised Domain Adaptation

Input: Source Data D_S, Target Data D_T
Output: Adjusted Source Data D_s^*
$C_S = cov(D_S) + eye(size(D_S, 2))$
$C_T = cov(D_T) + eye(size(D_T, 2))$
$D_S = D_S * C_S^{\frac{-1}{2}}$ % whitening source
$D_S^* = D_S * C_T^{\frac{1}{2}}$ % re-coloring with target covariance

Sun, Baochen, Jiashi Feng, and Kate Saenko. "Correlation alignment for unsupervised domain adaptation." *Domain Adaptation in Computer Vision Applications*. 2017

Domain Adaptation by CORrelation ALignmetn

Illustration of a two dimensional feature space for the classifier. No label information is provided (thus displayed)

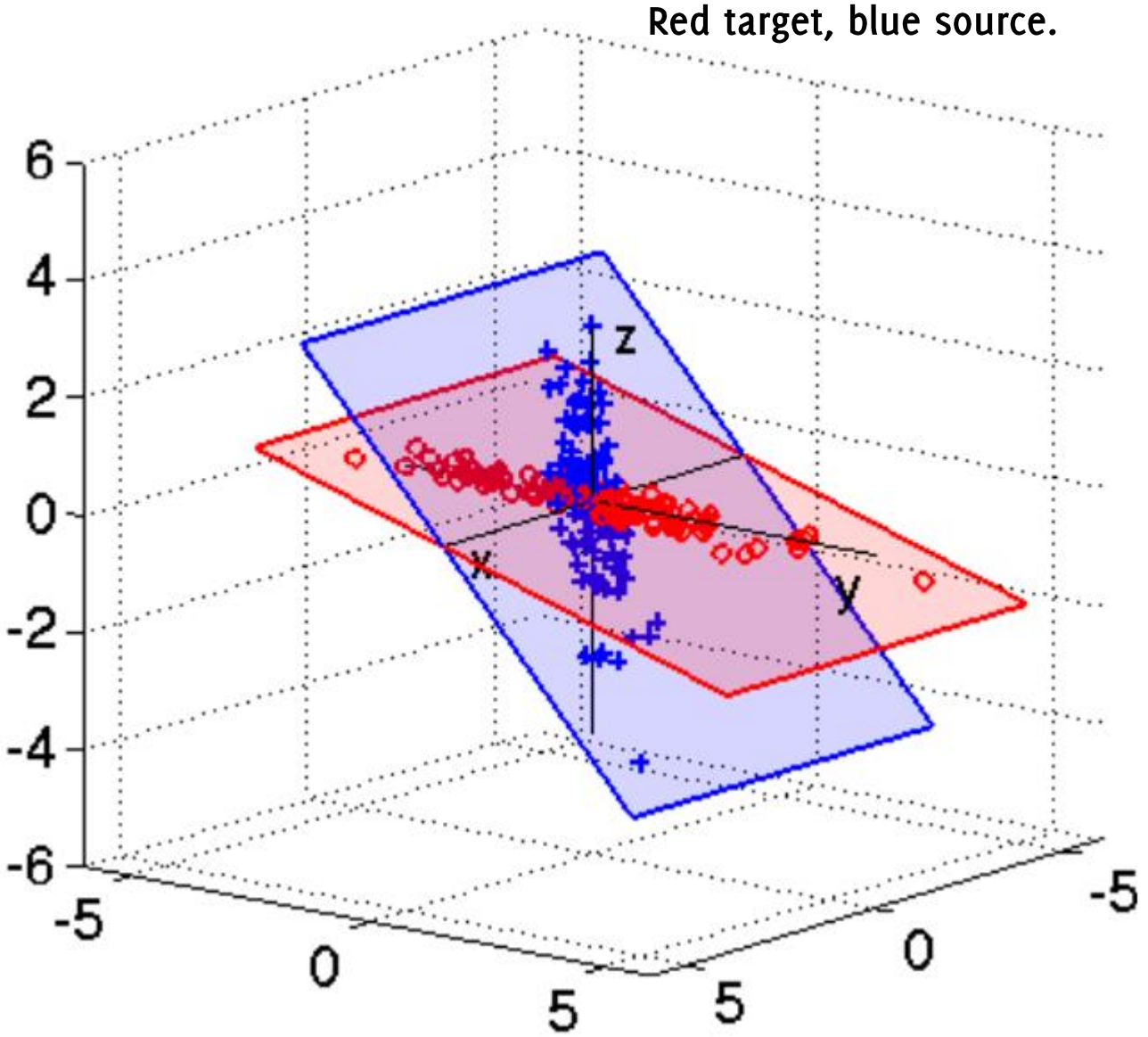


Sun, Baochen, Jiashi Feng, and Kate Saenko. "Correlation alignment for unsupervised domain adaptation." *Domain Adaptation in Computer Vision Applications*. 2017

Domain Adaptation by CORrelation ALignmetn

The original source and target domains have different distribution covariances, **despite the features standardization** (zero mean and unit standard deviation)

Compute sample covariances Σ_S and Σ_T



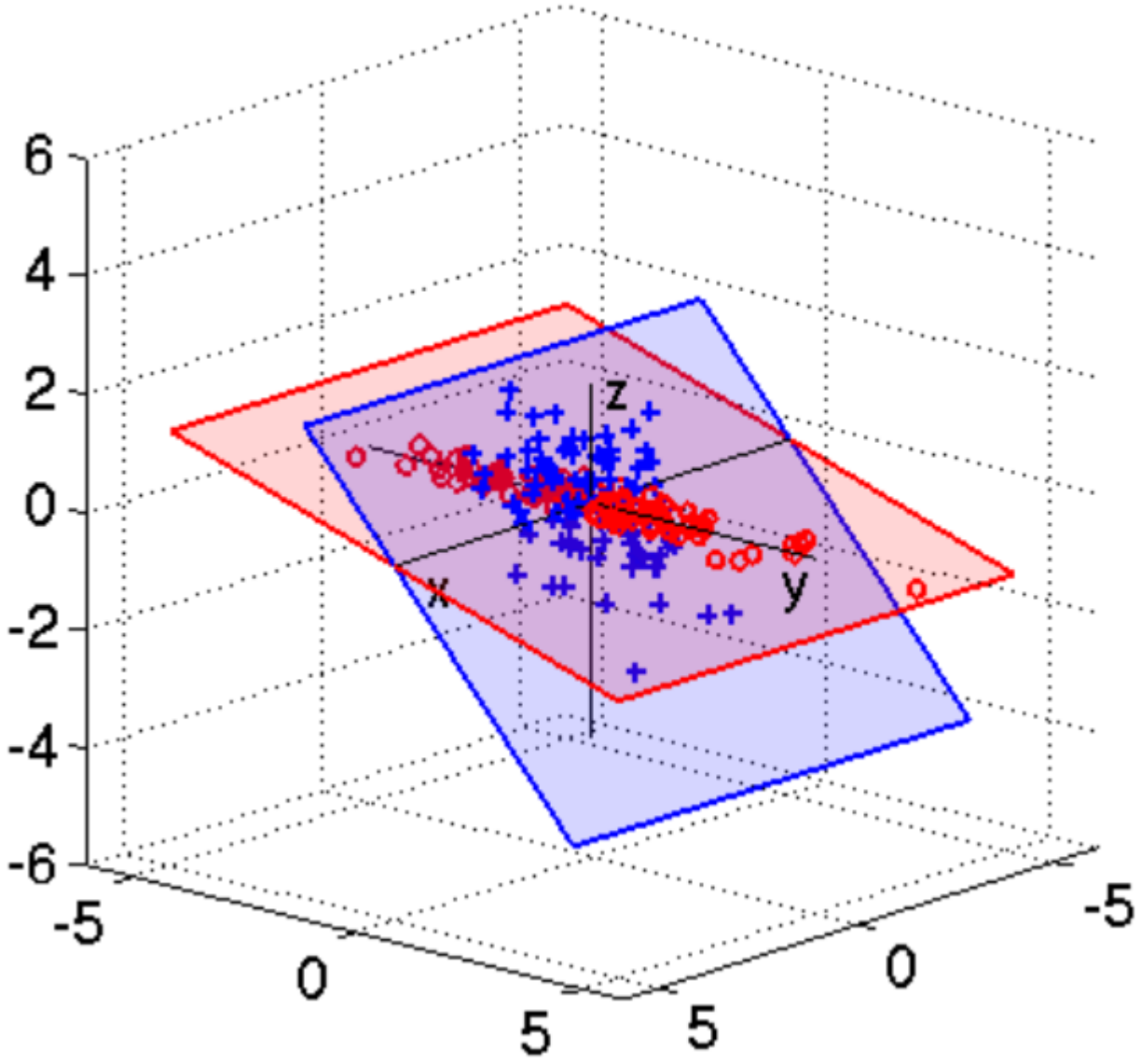
Sun, Baochen, Jiashi Feng, and Kate Saenko. "Correlation alignment for unsupervised domain adaptation." *Domain Adaptation in Computer Vision Applications*. 2017

Domain Adaptation by CORrelation ALignmetn

Whitening source data

$$\tilde{X}_S = \Sigma_S^{-\frac{1}{2}} X_S$$

Source decorrelation, i.e. removing the feature correlations of the source domain



Sun, Baochen, Jiashi Feng, and Kate Saenko. "Correlation alignment for unsupervised domain adaptation." *Domain Adaptation in Computer Vision Applications*. 2017

Domain Adaptation by CORrelation ALignmetn

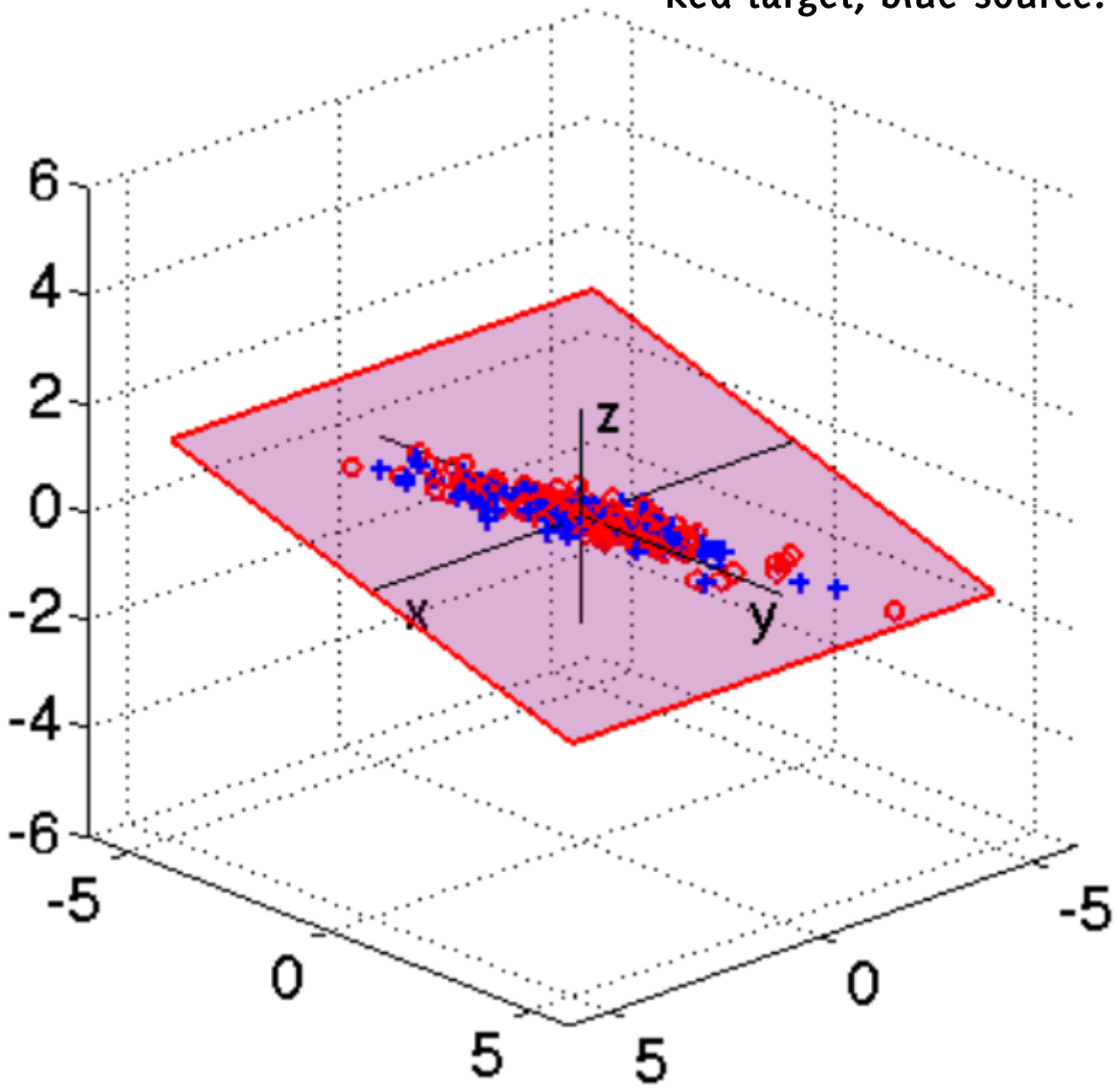
Colouring Target

$$X_S^* = \Sigma_T^{-\frac{1}{2}} \tilde{X}_S$$

Target re-correlation, adding the correlation of the target domain to the source features.

After this step, the source and target distributions are well aligned and the classifier trained on the adjusted source domain is expected to work well in the target domain.

Red target, blue source.



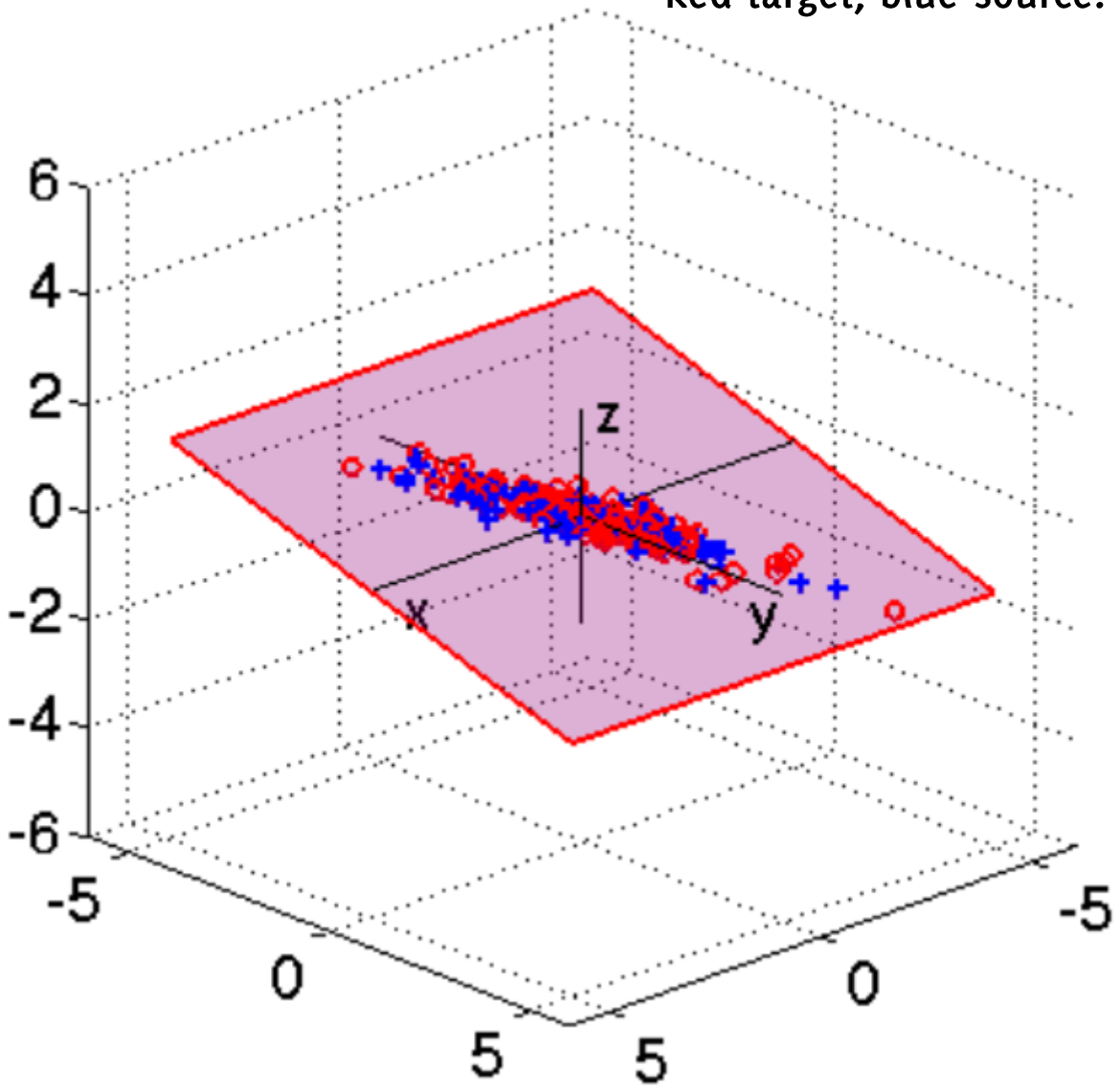
Sun, Baochen, Jiashi Feng, and Kate Saenko. "Correlation alignment for unsupervised domain adaptation." *Domain Adaptation in Computer Vision Applications*. 2017

Domain Adaptation by CORrelation ALignmetn

Train the classifier on X_S^* using source labels.

The source and target distributions are well aligned the classifier trained on the adjusted source domain is expected to work well in the target domain.

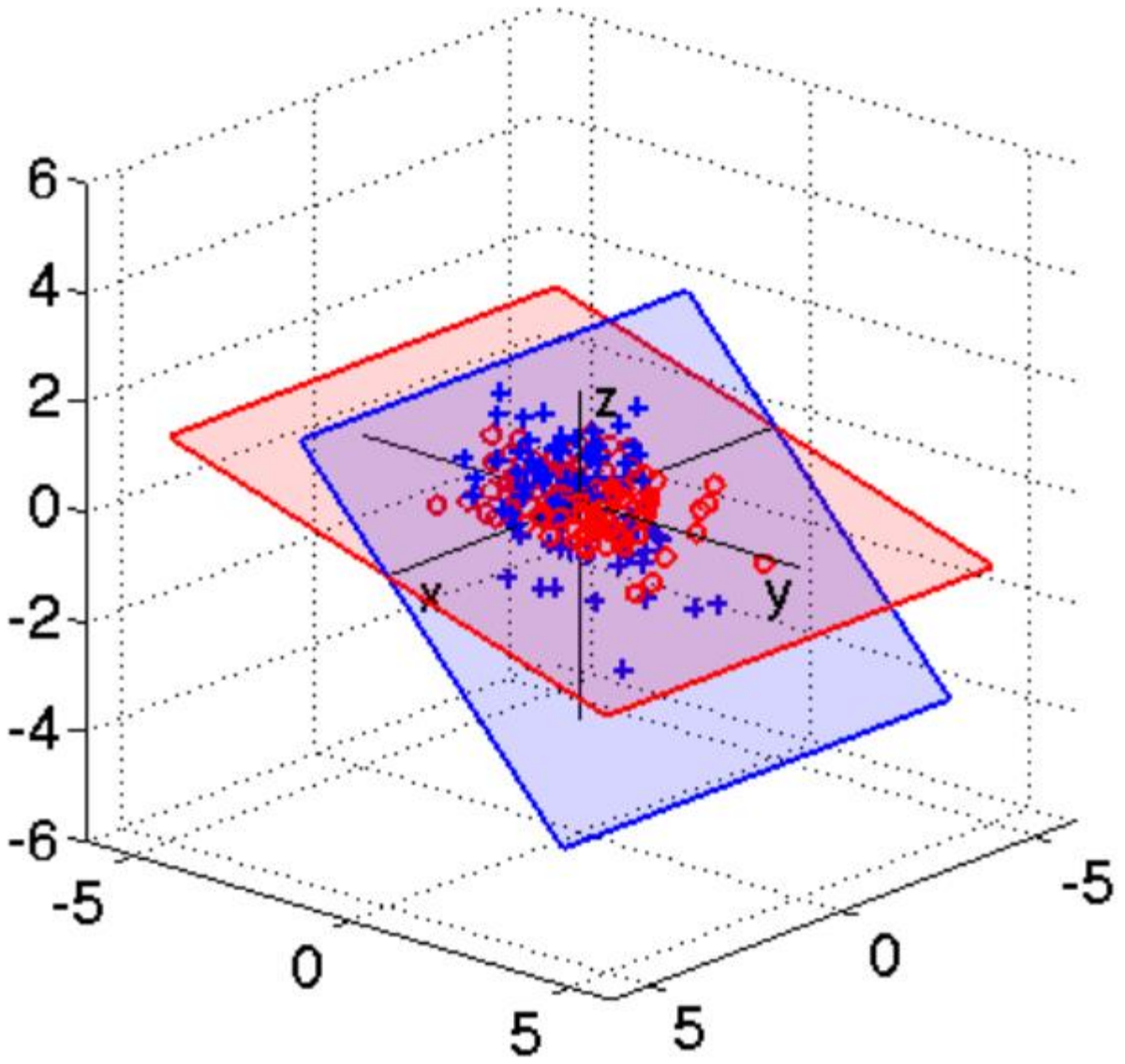
Red target, blue source.



Sun, Baochen, Jiashi Feng, and Kate Saenko. "Correlation alignment for unsupervised domain adaptation." *Domain Adaptation in Computer Vision Applications*. 2017

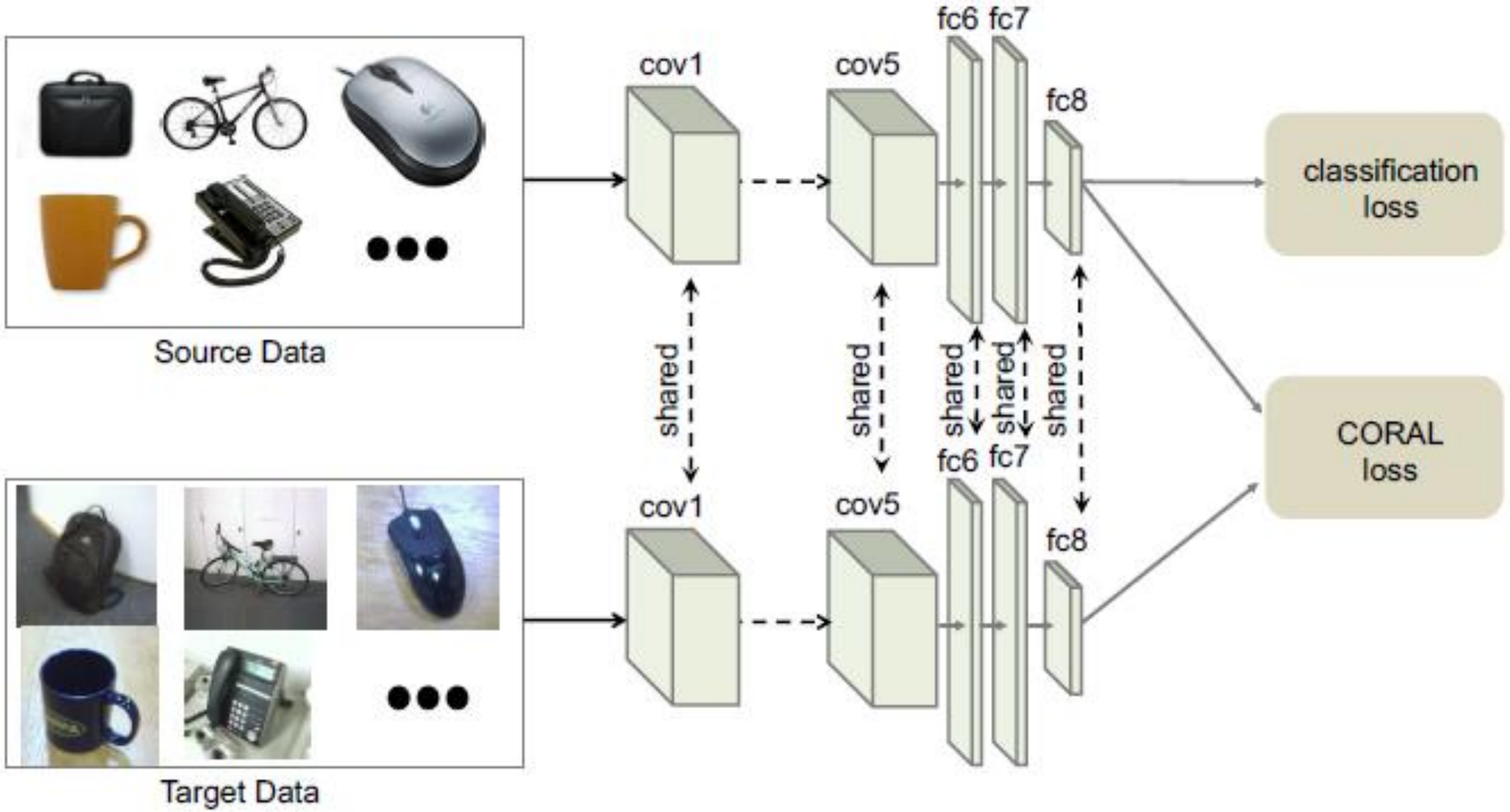
Domain Adaptation by CORrelation ALignmetn

One might instead attempt to align the distributions by whitening both source and target. However, this will fail since the source and target data are likely to lie on different subspaces due to domain shift



Sun, Baochen, Jiashi Feng, and Kate Saenko. "Correlation alignment for unsupervised domain adaptation." *Domain Adaptation in Computer Vision Applications*. 2017

Deep Coral



The final deep features need to be both discriminative enough to train a strong classifier and invariant to the difference between source and target domains

Sun, Baochen, Jiashi Feng, and Kate Saenko. "Correlation alignment for unsupervised domain adaptation." *Domain Adaptation in Computer Vision Applications*. 2017

Deep Coral

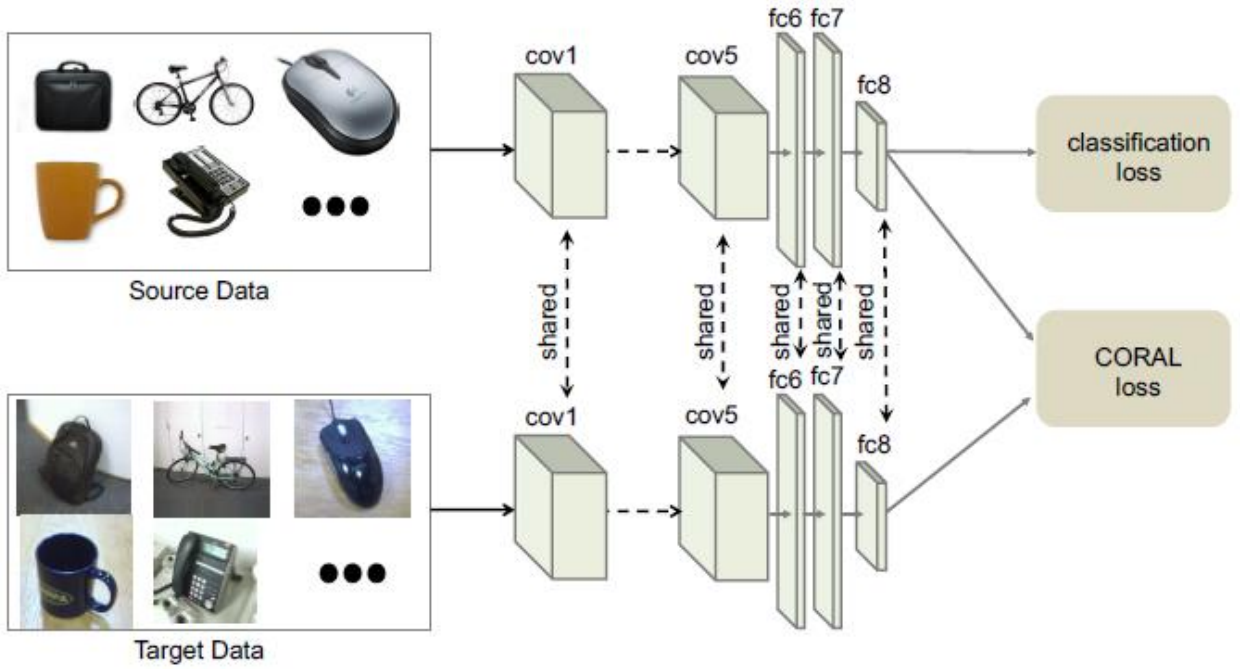
Include an additioal loss when training deep learning models.

$$\mathcal{L} = \mathcal{L}_{CLASS} + \sum_i \lambda_i \mathcal{L}_{CORAL_i}$$

Being the loss onn a single feature layer

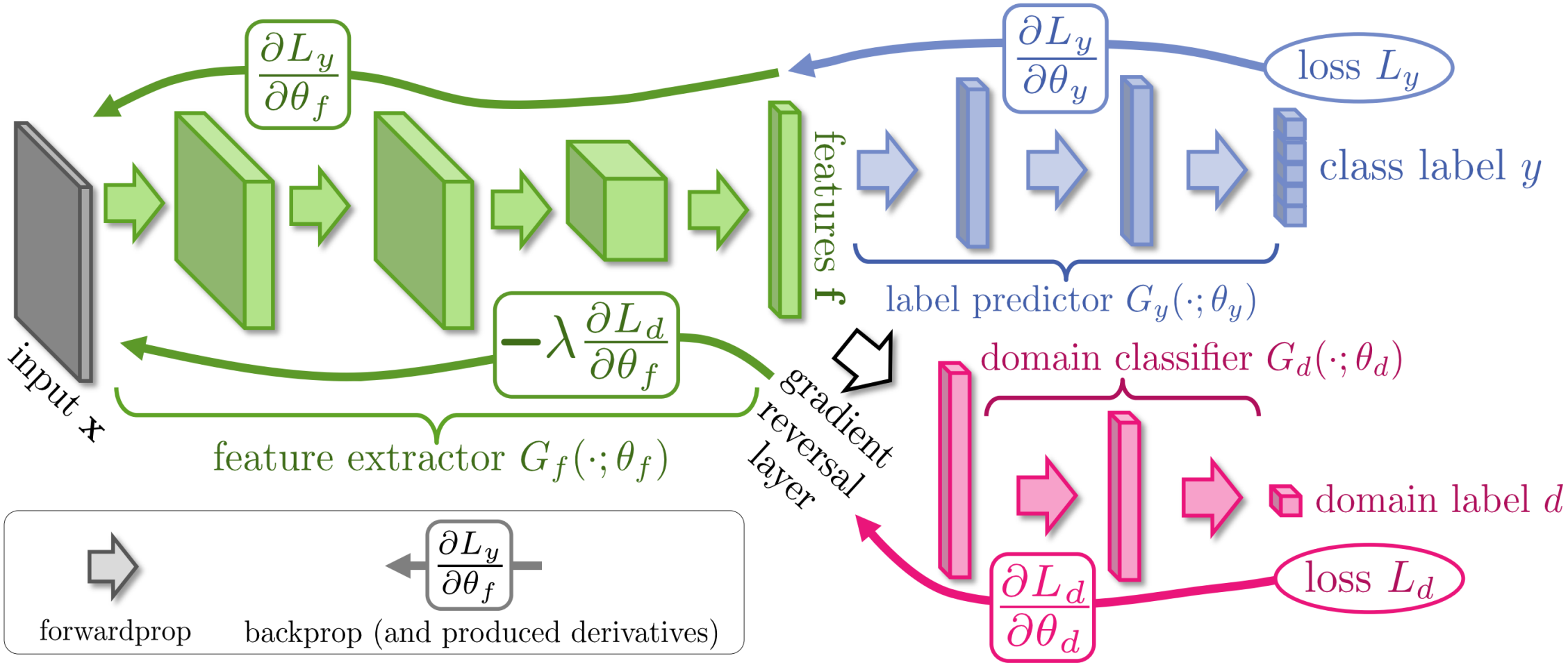
$$\mathcal{L}_{CORAL} = \frac{1}{4d^2} \|\Sigma_S - \Sigma_T\|_F^2$$

This is meant to bring source and target data to the same distribution



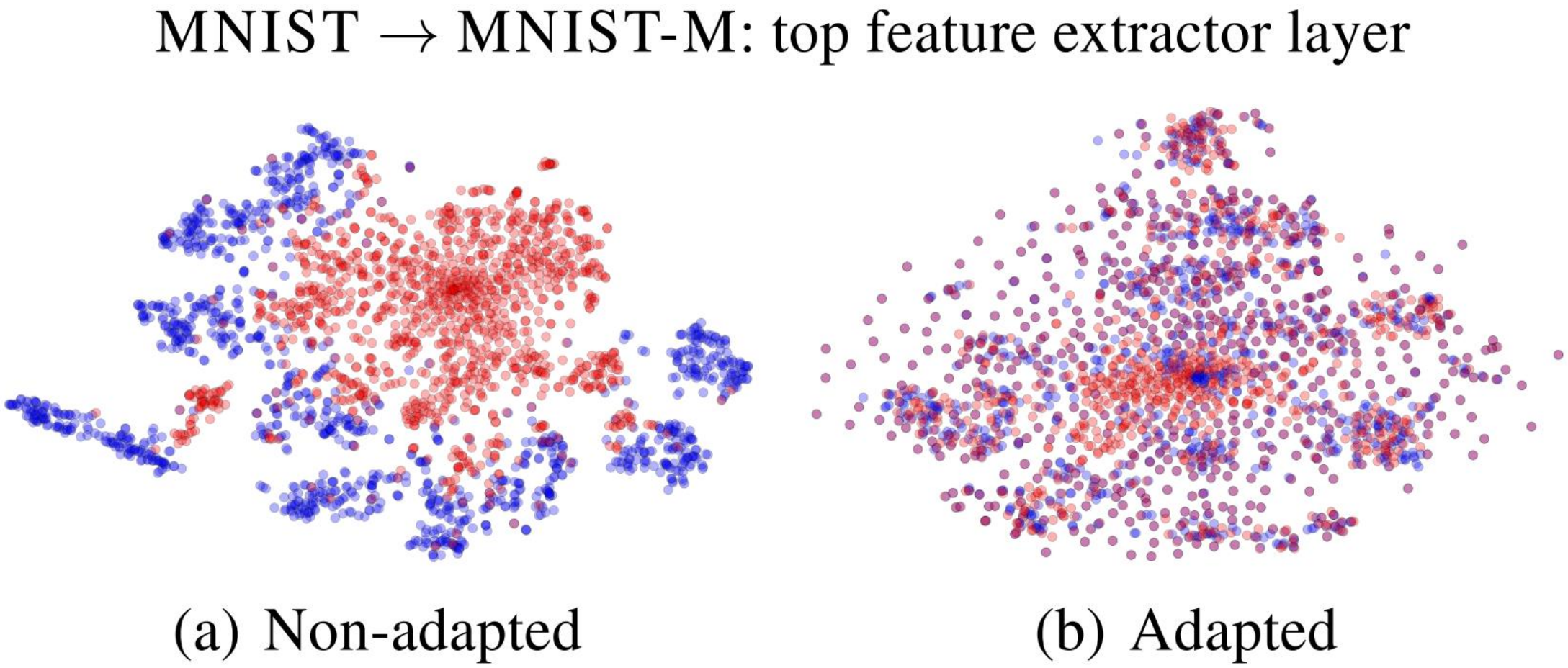
Sun, Baochen, Jiashi Feng, and Kate Saenko. "Correlation alignment for unsupervised domain adaptation." *Domain Adaptation in Computer Vision Applications*. 2017

Unsupervised Domain Adaptation



Ganin, Y., & Lempitsky, V. Unsupervised domain adaptation by backpropagation. *ICML 2015*

Unsupervised Domain Adaptation



Blue is the source, red is the target

Ganin, Y., & Lempitsky, V. Unsupervised domain adaptation by backpropagation. *ICML 2015*

More on Change Detection

And in particular on high-dimensional data-streams

G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016,

Desiderata, Challenge and Goal

Desiderata

i. The model $\hat{\phi}_0$ describing ϕ_0 has to be:
   - general and simple
   - learnable from a training set

ii. The statistic $\mathcal{T}$ used to test incoming data has to:
   - provide a controlled response under ϕ_0
   - provide a different response under ϕ_1

iii. A decision rule that monitors $\mathcal{T}$ has to:
   - promptly detect changes and
   - control FPR (type I error in hypothesis testing) or ARL (averge run length in sequential monitoring)

Research Challenges

Most of the research has been devoted to univariate monitoring schemes

- These were the first case studies in SPC
- Extension to monitoring classification / regression error are straightforward

Challenges for truly multivariate ($d > 1$) monitoring scheme, $d \gg 1$:

- Parametric models $\hat{\phi}_0$ properly matching ϕ_0 are difficult to find
- Non parametric models often require:
  - prohibitively large training sets
  - prohibitively long computing times

Research Challenges

Build a model $\hat{\phi}_0$ and a truly multivariate monitoring scheme that:

- allows change detection in multivariate, possibly high dimensional data
- guarantees a control over the false positives
- it does not require too many training data
- it is very efficient to test

Do I really need $\hat{\phi}^0$?

Of course not….

But it might come handy

Do I really need $\hat{\phi}^0$?

There are a few **non-parametric statistcs** that do need to fit $\hat{\phi}_0$

- Mann Whitney
- Mood
- Lepage
- Kolmogorov-Smirnov
- Cramer Von Mises

Thresholds are either:

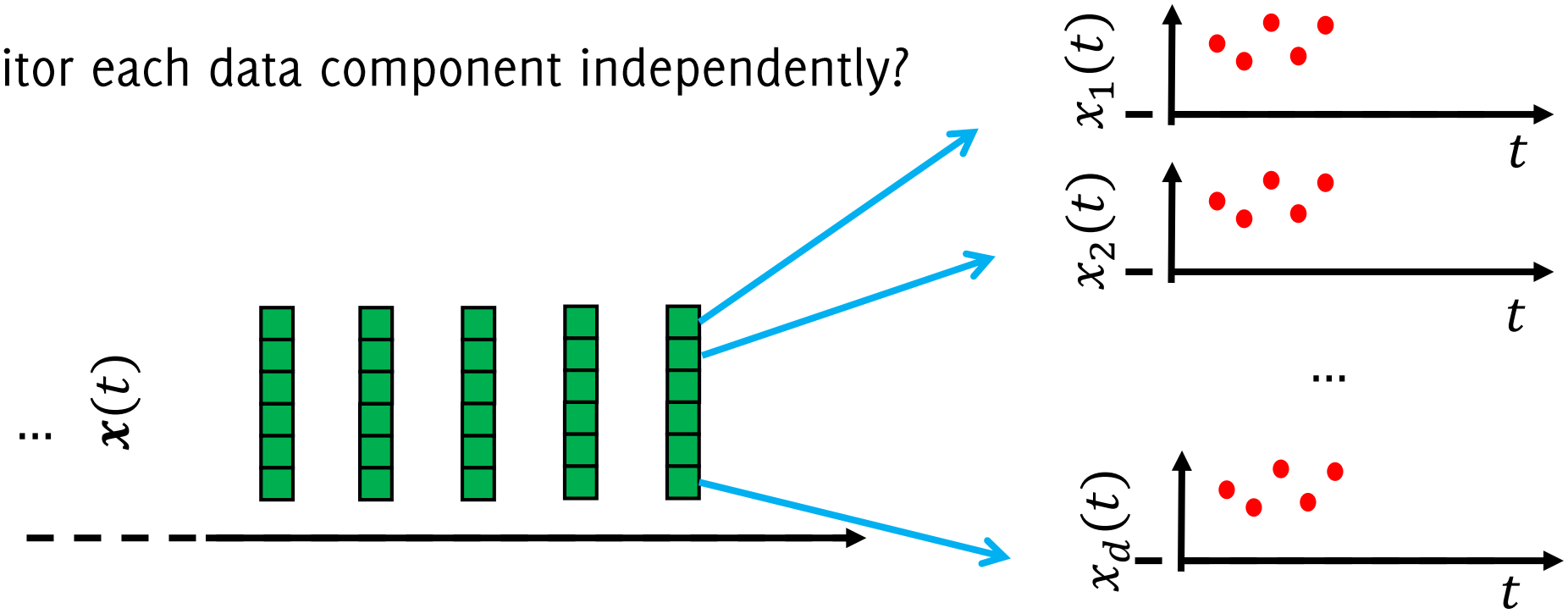
- provided by asymptotic approximation of the statistic
- easily computed from numerical simulations

Unfortunately, most nonparametric statistics relies on sorting and **cannot be extended to multivariate data**

Ross, G. J. and Adams, N. M. "Two nonparametric control charts for detecting arbitrary distribution changes". Journal of Quality Technology, 2012.
Ross, G. J., Tasoulis, D. K., and Adams, N. M. "Nonparametric monitoring of data streams for changes in location and scale". Technometrics, 2011.

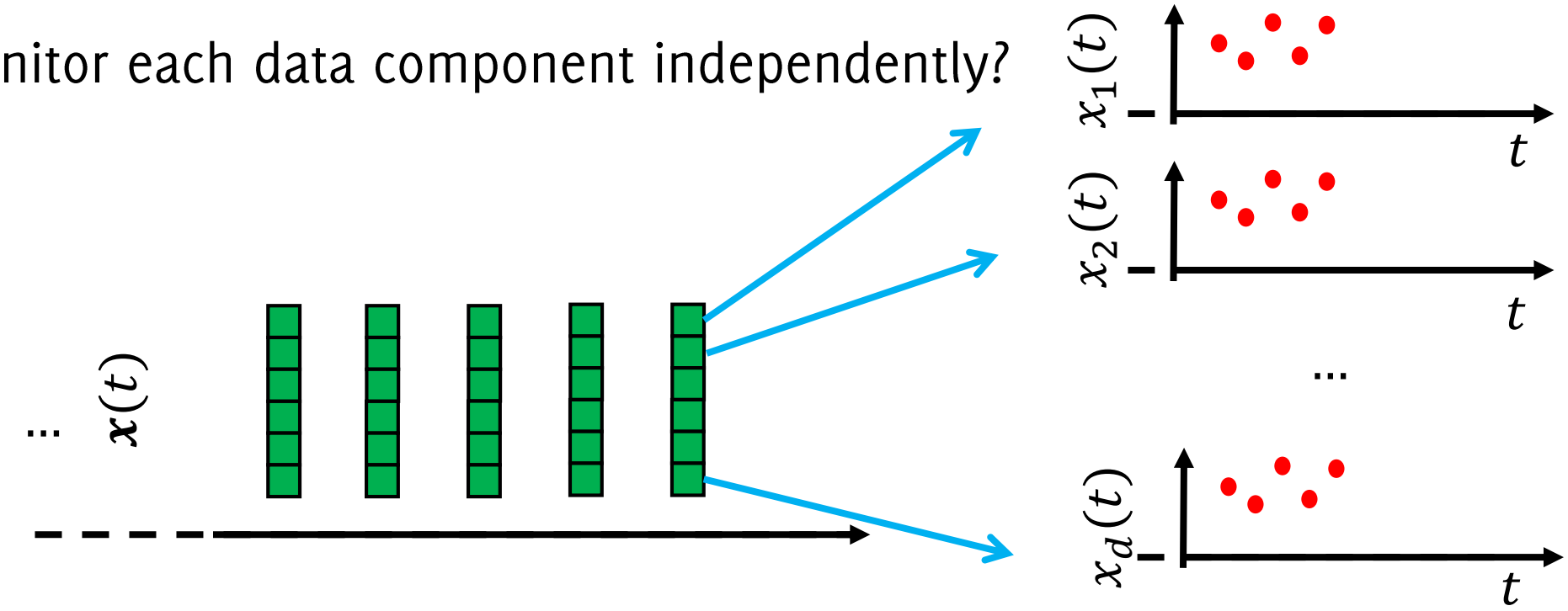
Can’t I go back to a few univariate problems?

Can’t I monitor each data component independently?

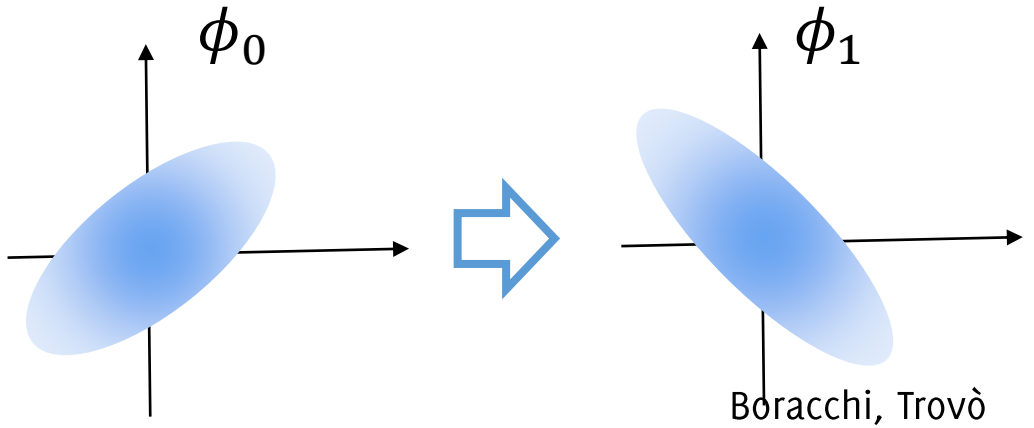


Can’t I go back to a few univariate problems?

Can’t I monitor each data component independently?



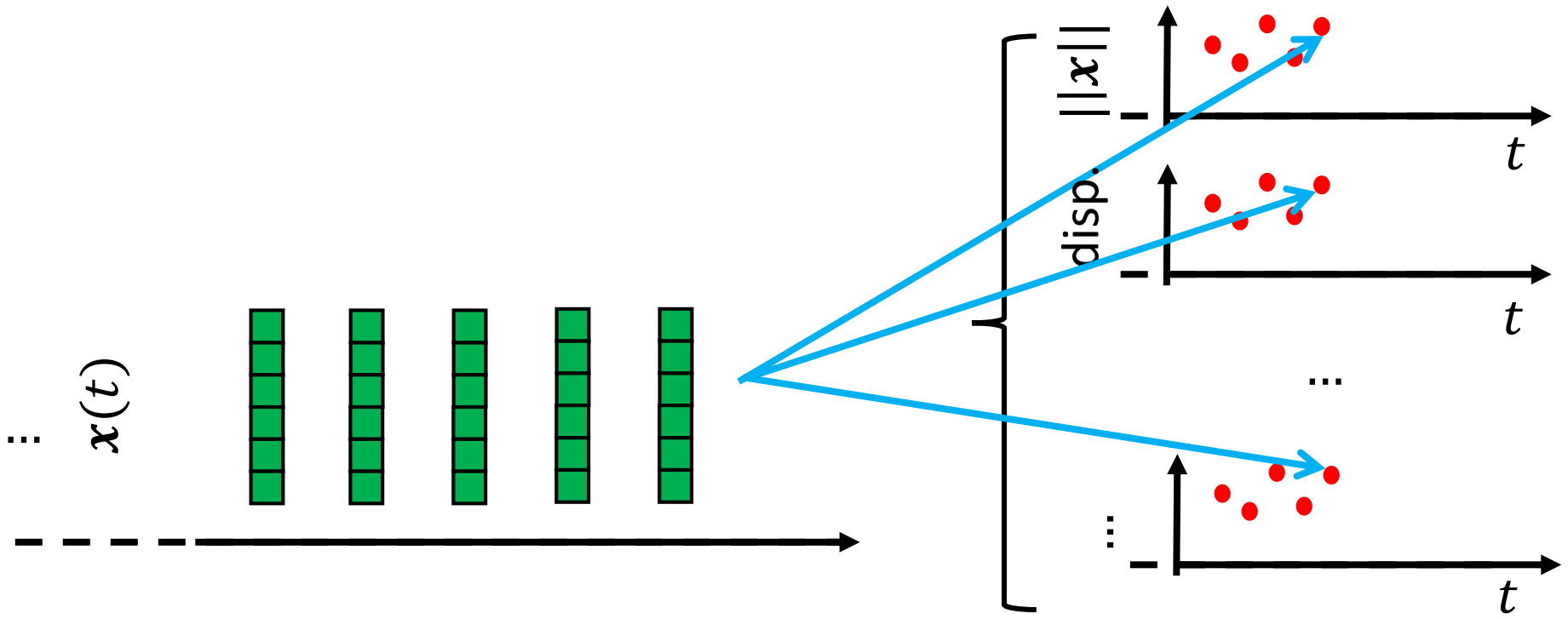
This is **not a truly multivariate** monitoring scheme.
for instance You would not be able to detect
changes affecting correlation



Boracchi, Trovò

Can’t I go back to a few univariate problems?

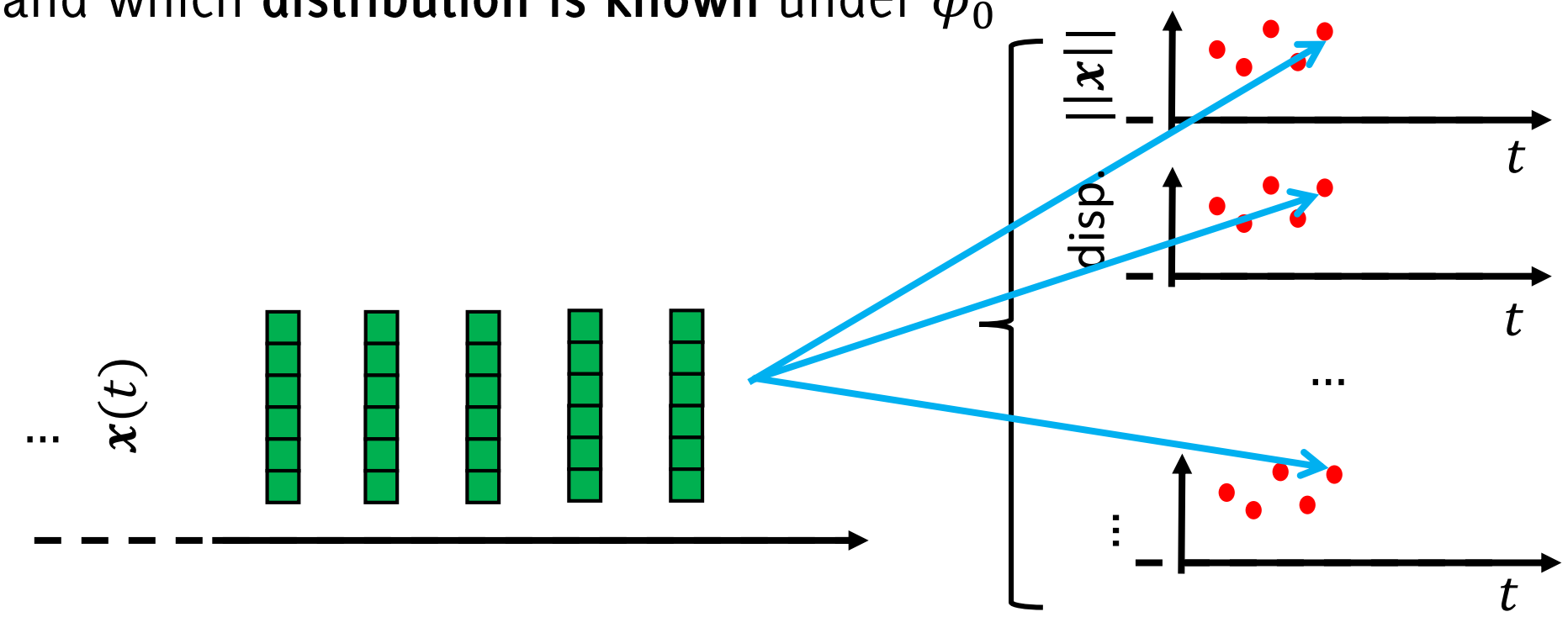
Can’t I **extract a few features / indicators** that are **expected to change** when $\phi_0 \rightarrow \phi_1$ and which **distribution is known** under ϕ_0



Alippi, C., Boracchi, G., Roveri, M. *“Change detection tests using the ICI rule”* IJCNN 2010 (pp. 1-7).

Can’t I go back to a few univariate problems?

Can’t I **extract a few features / indicators** that are **expected to change** when $\phi_0 \rightarrow \phi_1$ and which **distribution is known** under ϕ_0



Not truly multivariate: **only changes affecting features are detectable.**

Alippi, C., Boracchi, G., Roveri, M. *“Change detection tests using the ICI rule”* IJCNN 2010 (pp. 1-7).

To warp-up: limitations

When d increases,

- Defining models $\hat{\phi}_0$ that match ϕ_0 becomes more difficult
- Non parametric models often require
  - prohibitively large training sets
  - prohibitively long computing times
- Most nonparametric statistics are based on ranking and do not apply when $d > 1$
- «component-wise monitoring» is too coarse a solution
- «feature-extraction and monitoring» does not lead to a truly multivariate monitoring
- Overall, it is not easy to control FPR / ARL

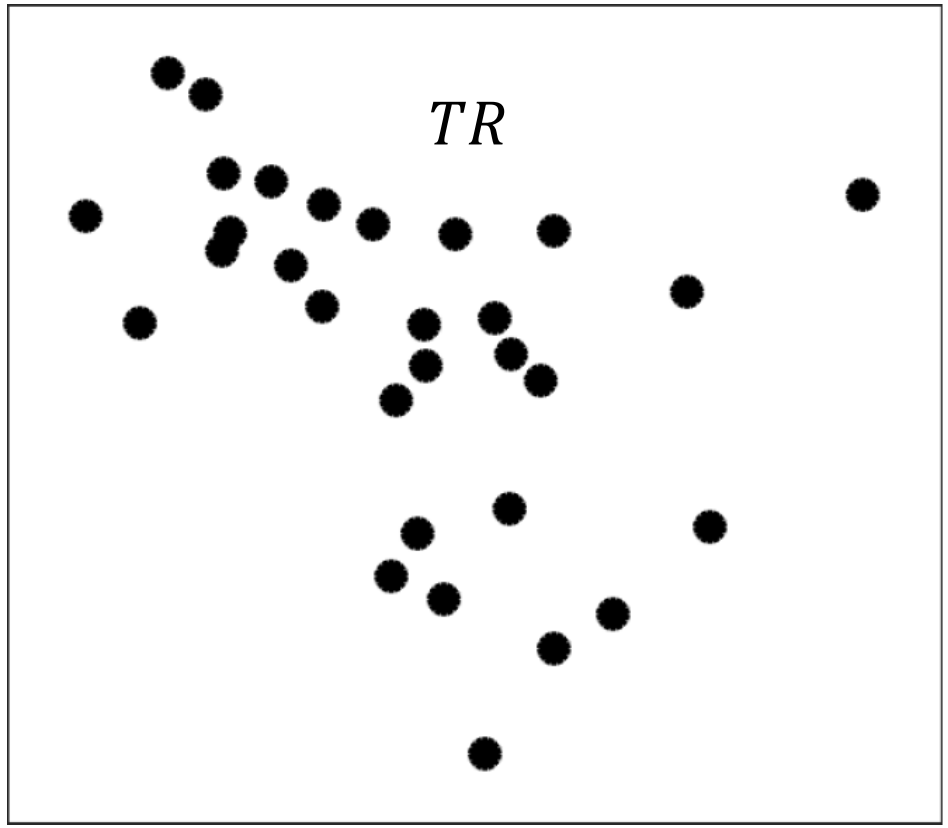
QuantTrees

A partitioning scheme specifically designed for change detection

G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

QuantTrees: Histograms for change detection

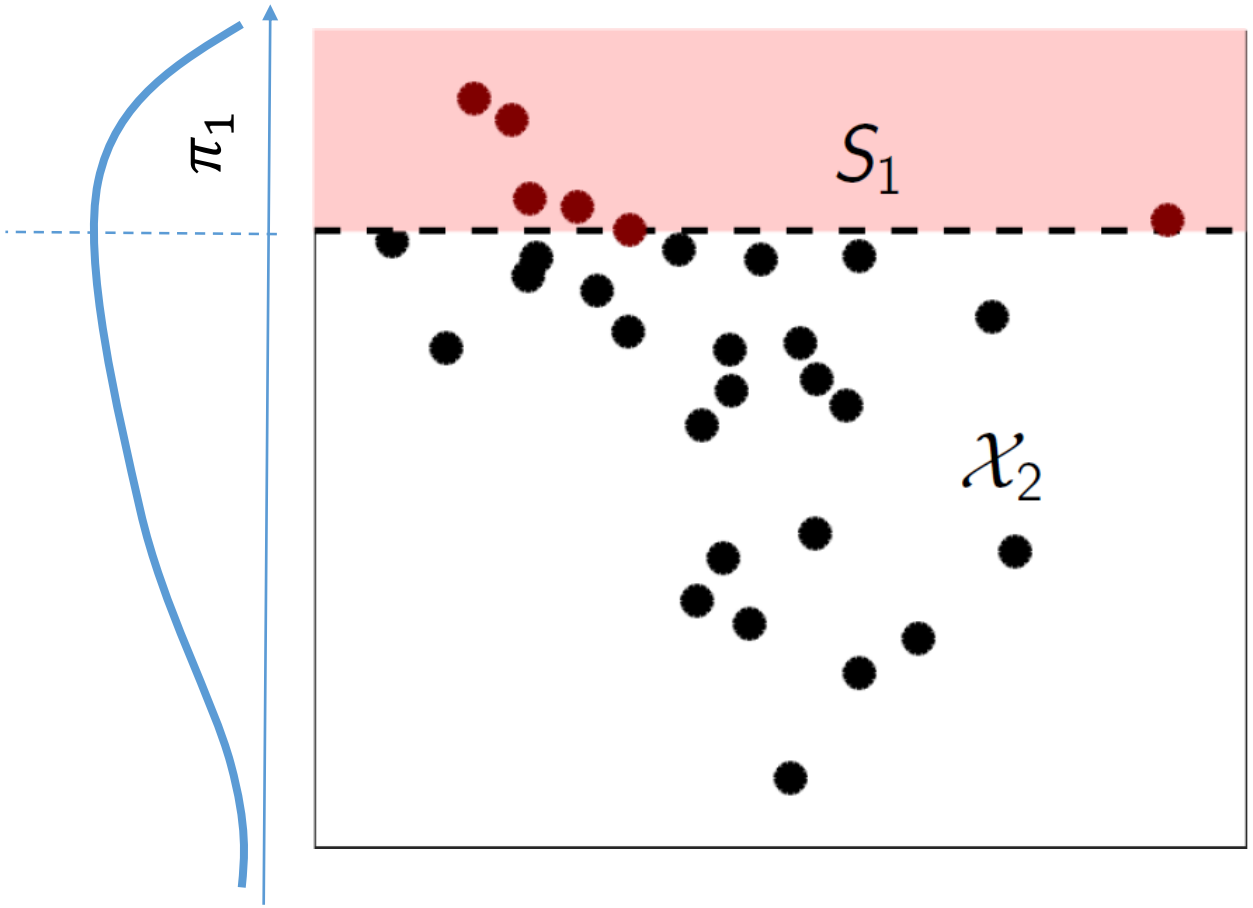
Assume you are given a set of target probabilities $\{\pi_i\}_{i=1,..,K}$ and a training set TR

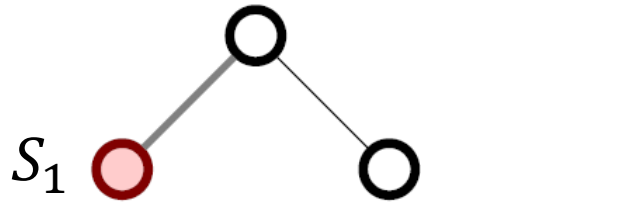


G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

QuantTrees: Histograms for change detection

Choose a dimension j at random, define the S_1 as the set containing the $1 - \pi_1$ quantile of the marginal distribution of training samples along j



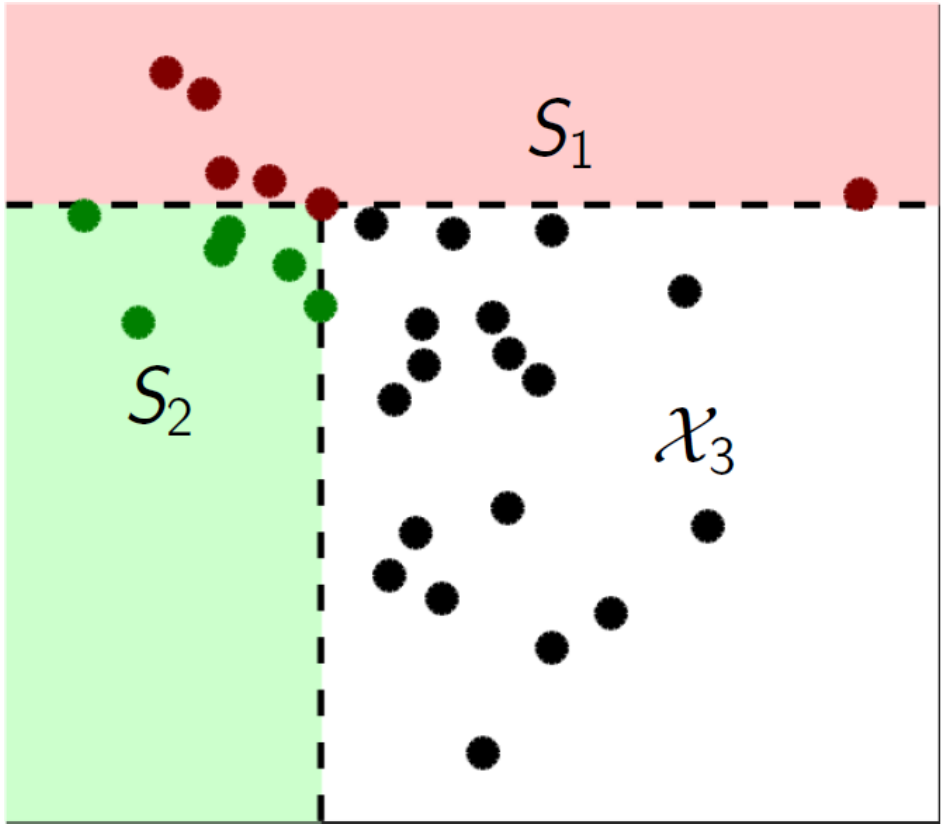


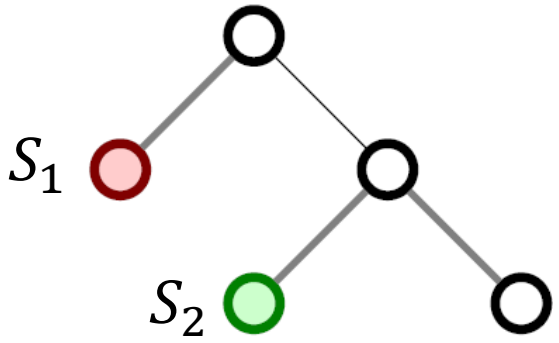
Call $\mathcal{X}_2$ the remaining samples

G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

QuantTrees: Histograms for change detection

The procedure is iterated on the training samples that have not been included in a bin.

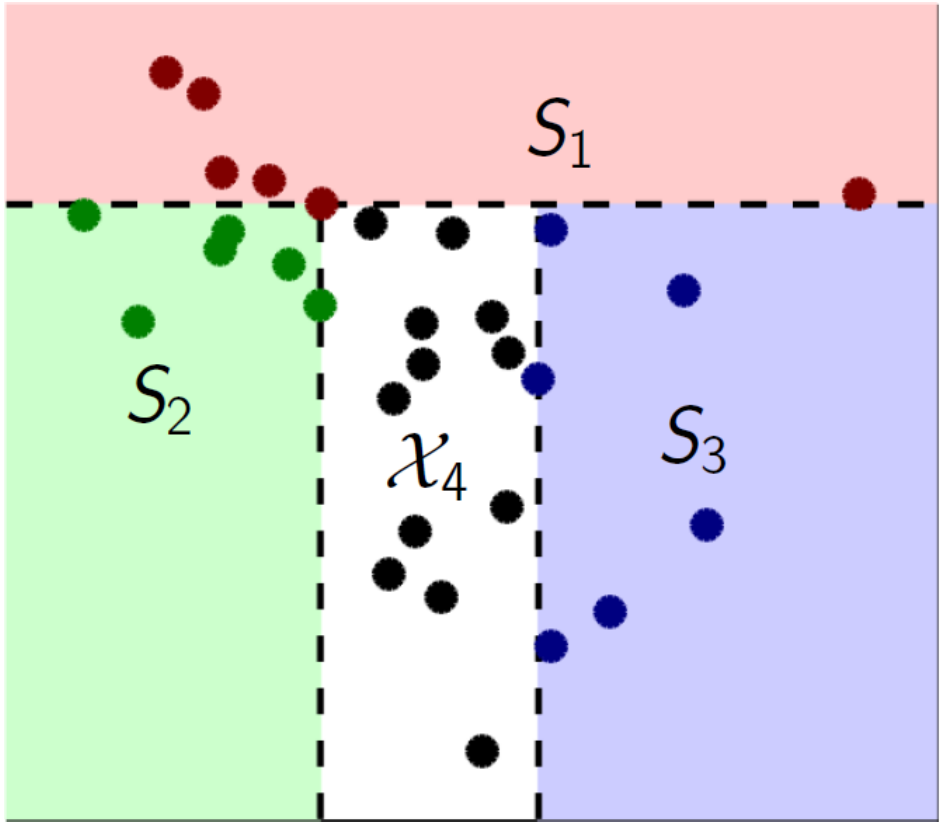


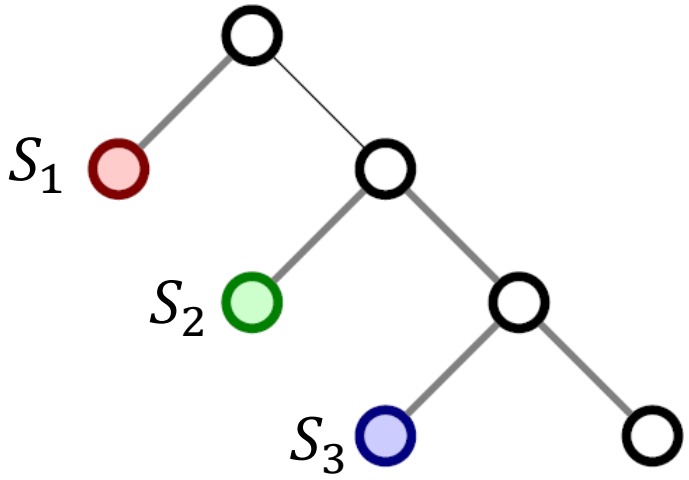


G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

QuantTrees: Histograms for change detection

The procedure is iterated on the training samples that have not been included in a bin.

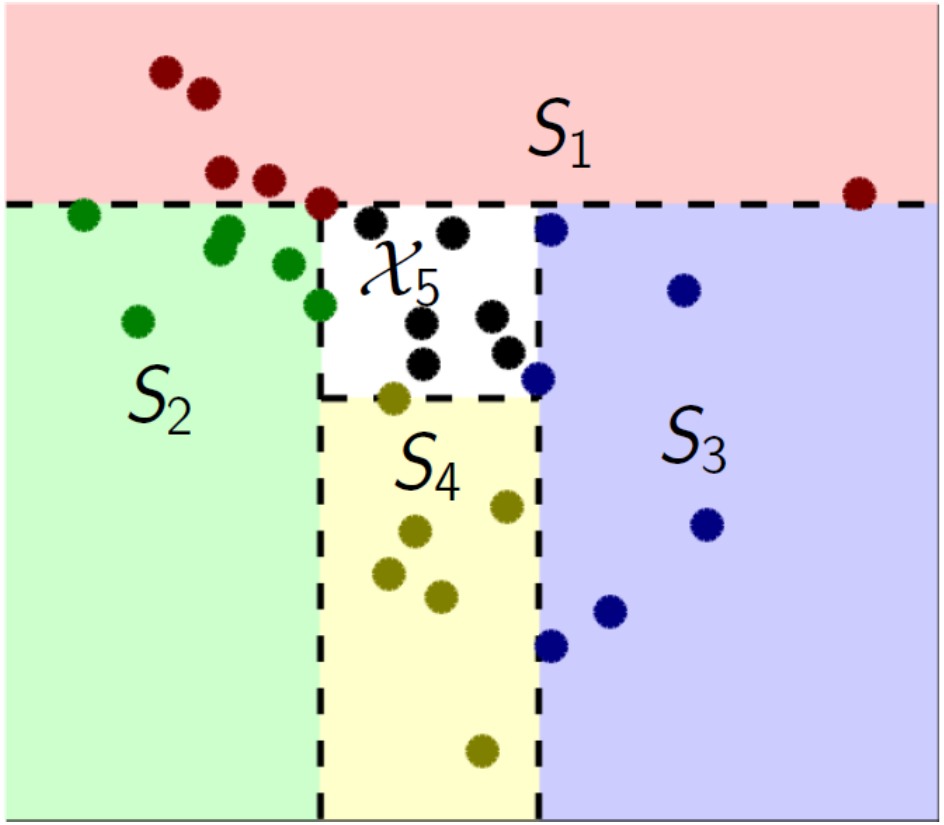


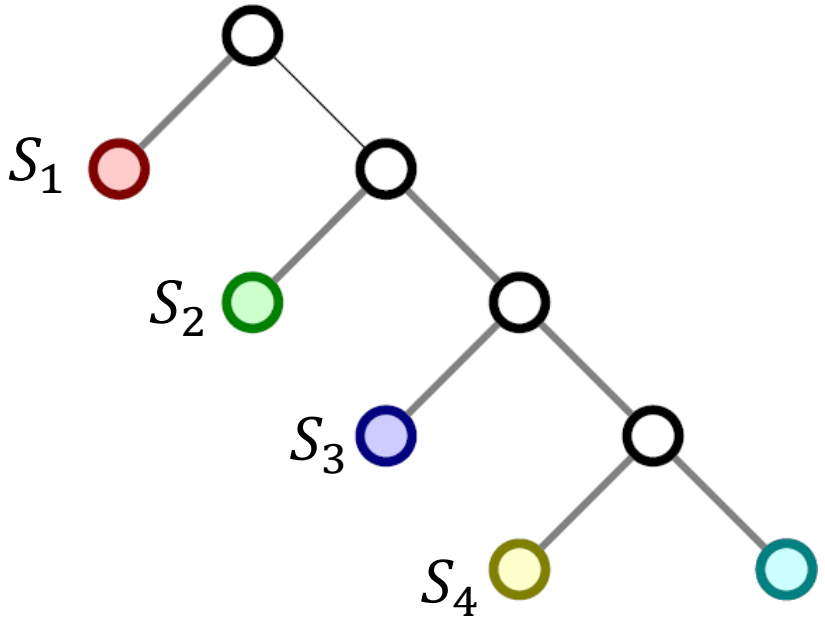


G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

QuantTrees: Histograms for change detection

The procedure is iterated on the training samples that have not been included in a bin.

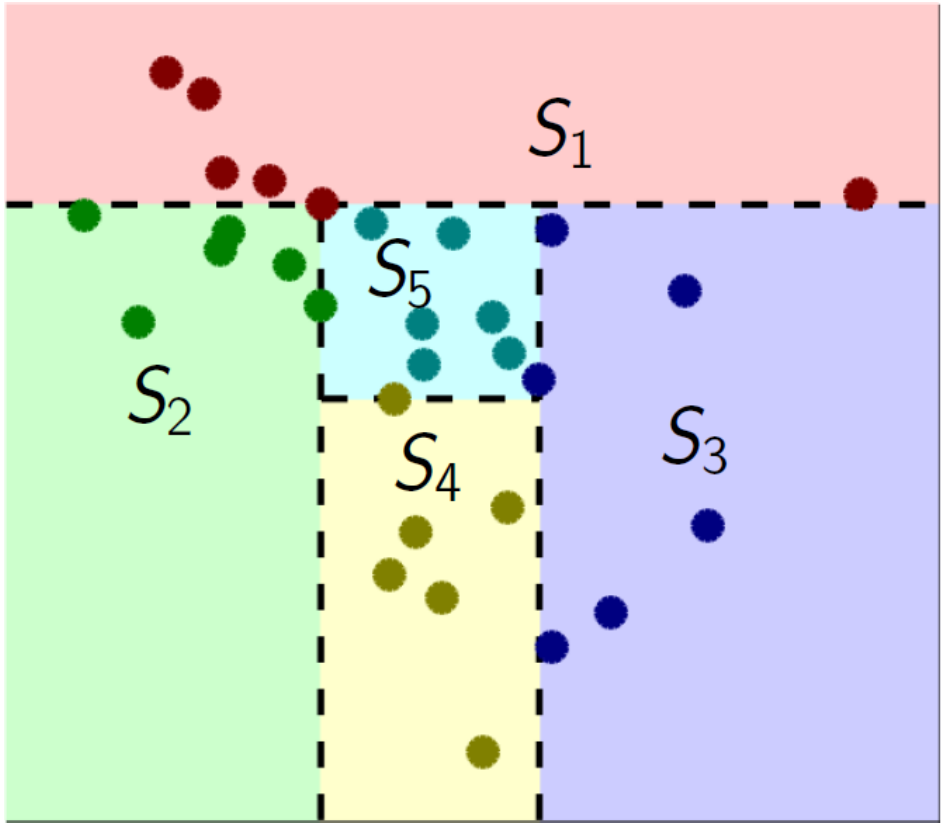


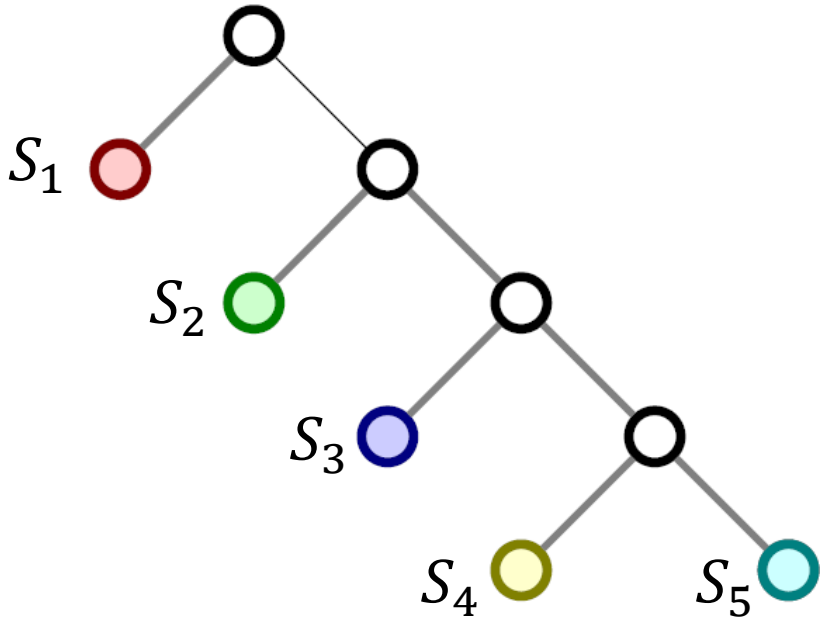


G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

QuantTrees: Histograms for change detection

The procedure is iterated on the training samples that have not been included in a bin.





G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

QuantTrees: Histograms for change detection

QuantTree **iteratively divides** the input space by **binary splits along a single covariate**, where the cutting points are defined by the **quantiles of the marginal distributions**

Algorithm 1 QuantTree

Input: Training set TR containing N stationary points in $\mathcal{X}$; number of bins K; target probabilities $\{\pi_k\}_k$.

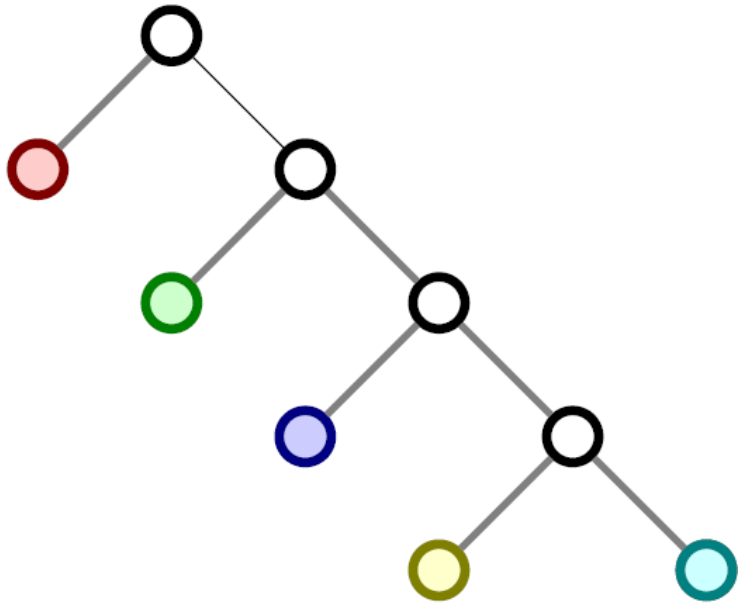
Output: The histogram $h = \{(S_k, \widehat{\pi}_k)\}_k$.

Set $N_0 = N$, $L_0 = 0$.
for $k = 1, \ldots, K$ **do**
    Set $N_k = N_{k-1} - L_{k-1}$, $\mathcal{X}_k = \mathcal{X} \setminus \bigcup_{j<k} S_j$, and $L_k = \text{round}(\pi^k N)$.
    Choose a random component $i \in \{1, \ldots, d\}$.
    Define $z_n = [\mathbf{x}_n]_i$ for each $\mathbf{x}_n \in \mathcal{X}_k$.
    Sort $\{z_n\}$: $z_{(1)} \leq z_{(2)} \leq \ldots z_{(N_k)}$.
    Draw $\gamma \in \{0, 1\}$ from a Bernoulli(0.5).
    if $\gamma = 0$ **then**
        Define $S_k = \{\mathbf{x} \in \mathcal{X}_k \quad [\mathbf{x}]_i \leq z_{(L_k)}\}$.
    else
        Define $S_k = \{\mathbf{x} \in \mathcal{X}_k \quad [\mathbf{x}]_i \geq z_{(N_k - L_k + 1)}\}$.
    end if
    Set $\widehat{\pi}_k = L_k / N$.
end for

G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

QuantTrees: Histograms for change detection

QuantTree **iteratively divides** the input space by **binary splits along a single covariate**, where the cutting points are defined by the **quantiles of the marginal distributions**

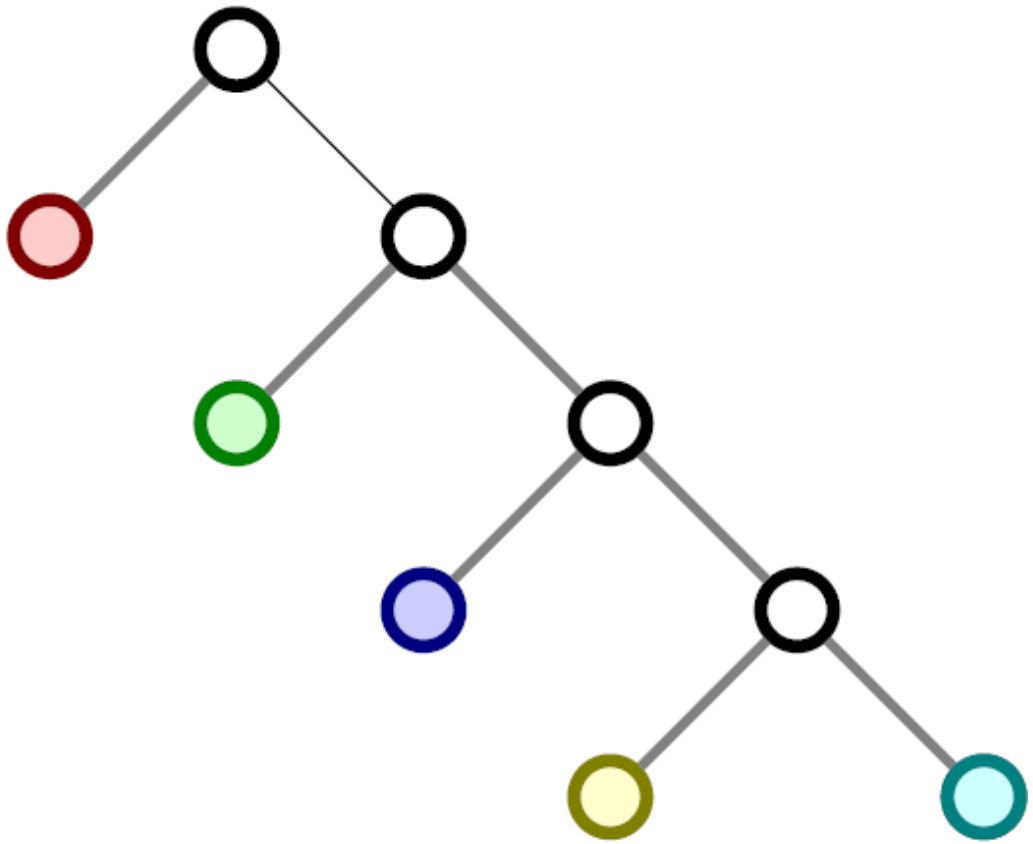


The QuantTree construction is randomized by the random selection of the component for each split and whether to take the π_i or $1 - \pi_i$ quantile

G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

QuantTrees: Histograms for change detection

The resulting histogram corresponds to a tree, which is highly unbalanced (thus not efficient to be scanned) that is very advantageous for change-detection purposes.



G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

QuantTrees: Histograms for change detection

Theorem (ICML18)

Let $T_h(\cdot)$ be a statistic defined over the bin probabilities of an histogram h computed by QuantTree.

When $W \sim \phi_0$, the distribution of $T_h(W)$ depends only on:

- the number of training samples N,
- the size of window W,
- the expected probabilities in each bin $\{\pi_i\}_{i=1,\dots,K}$

G. Boracchi, D. Carrera, C. Cervellera, D. Macciò *"QuantTree: Histograms for Change Detection in Multivariate Data Streams"* ICML 2018

Implications

In histograms constructed by QuantTrees, the bin probabilities do not depend on ϕ_0, nor data dimension d.

Thus, **thresholds** of tests statistics **can be numerically computed from univariate data** that have been synthetically generated, yet guaranteeing a controlled false positive rate.

	$d > 1$	$d = 1$
Training	$O(KN \log N)$	$O(N \log N)$
Test	$O(K)$	$O(\log K)$

Example of Thresholds Computed for QuantTrees (from 1D Montecarlo simulation)

α	Pearson		Total Variation			
	$K = 32$	$K = 128$	$K = 32$	$K = 128$	N	ν
0.001	64	192	25	43	4096	64
	62.75	187	52	85	16384	256
0.01	54	172	23	42	4096	64
	53.25	171	47	81	16384	256
0.05	46	156	21	41	4096	64
	45.75	157	44	78	16384	256

QuantTrees

Provide a model $\hat{\phi}_0$ and a truly multivariate monitoring scheme that:

- allows change detection in multivariate, possibly high dimensional data
- guarantees a control over the false positives
- it does not require too many training data
- it is very efficient to test

Sequential Monitoring by QuantTrees

So far we have been addressing an hypothesis testing framework

- The same properties of the statistics guarantees nonparametric monitoring also in other schemes
- However controlling ARL is more complicated -> we need QT-EWMA

Proposed Solution

Algorithm 1: QT-EWMA

input : datastream $x_1, x_2, \ldots$, target $\{\pi_j\}_{j=1}^K$, thresholds $\{h_t\}_t$, TR
output : detection flag `ChangeDetected`, detection time t^*

`ChangeDetected` $\leftarrow$ False, $t^* \leftarrow \infty$;
estimate QT histogram $\{(S_j, \pi_j)\}_{j=1}^K$ from TR and define $\{\hat{\pi}_j\}_{j=1}^K$ as in (4);
$Z_{j,0} \leftarrow \hat{\pi}_j \ \forall j = 1, \ldots, K$;
for $t = 1, \ldots$ **do**
  $y_{j,t} \leftarrow \mathbb{1}(x_t \in S_j)$;
  $Z_{j,t} \leftarrow (1-\lambda) Z_{j,t-1} + \lambda y_{j,t}, \quad j = 1 \ldots, K$;
  $T_t \leftarrow \sum_{j=1}^K (Z_{j,t} - \hat{\pi}_j)^2 / \hat{\pi}_j$;
  if $T_t > h_t$ **then**
    `ChangeDetected` $\leftarrow$ True, $t^* \leftarrow t$;
    break;
  end
end
return `ChangeDetected`, t^*

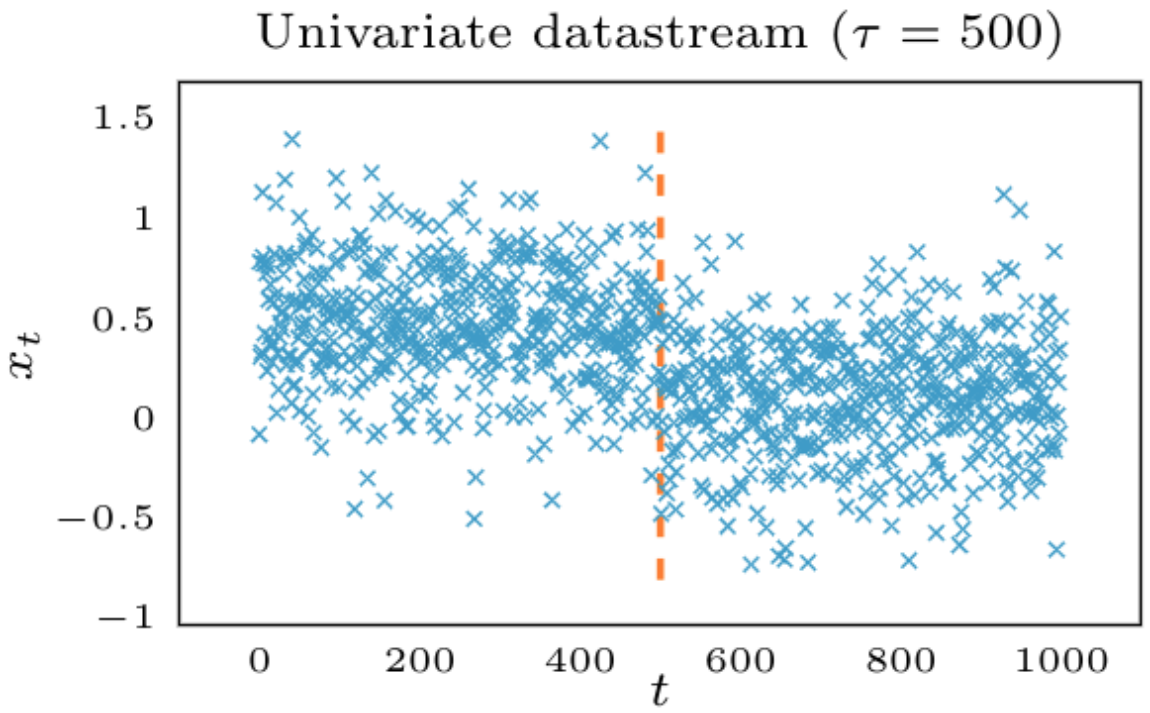
find in which bin each sample falls

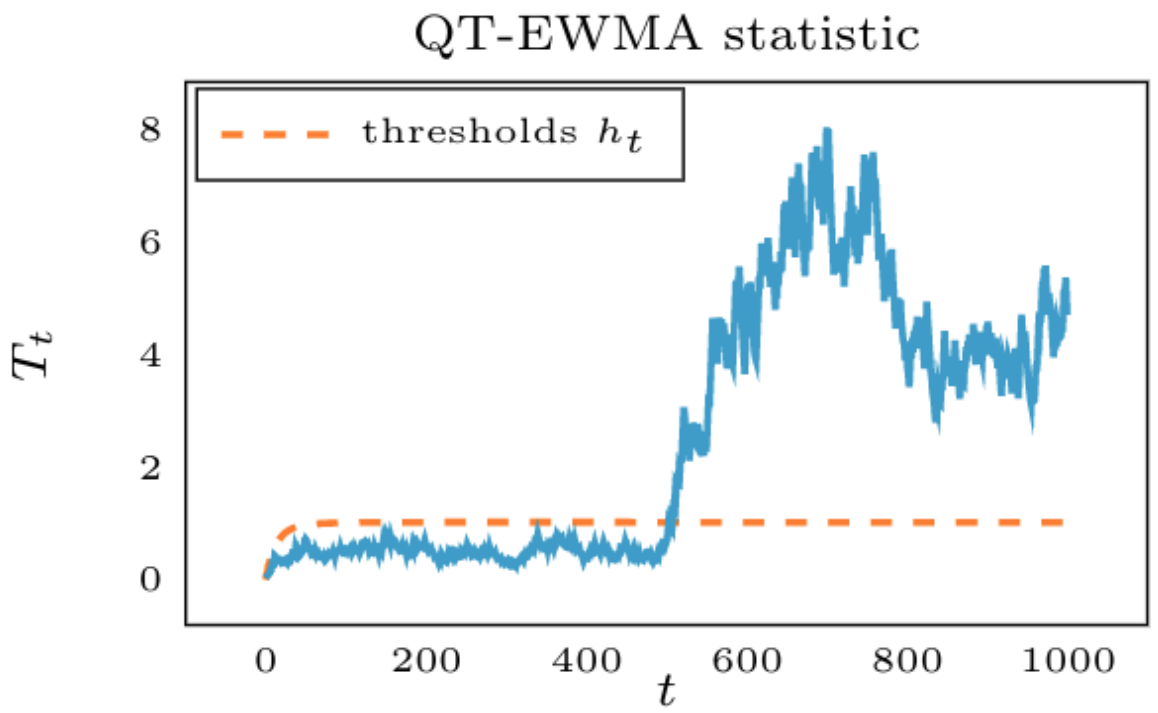
monitor the empirical bin probabilities by EWMA statistics $Z_{j,t}$

measure the deviation from the expected probabilities by T_t

detect a change when T_t exceeds a threshold h_t

Example





The **deviation** of the **bin probabilities** from their expected values measured by T_t **increases** after a **distribution change**

Thresholds Controlling the ARL_0

The theoretical properties of QuantTree **guarantee** that our statistics are **independent** from the data distribution

We design an efficient **Monte Carlo scheme** to define thresholds that maintain a target ARL_0 by setting a **constant false alarm probability**

$$\mathbb{P}(T_t > h_t | T_k \leq h_t \forall k < t) = \alpha = \frac{1}{\mathrm{ARL}_0}$$

T. M. Margavio et al. “Alarm Rates for Quality Control Charts”, Statistics & Probability Letters, 1995

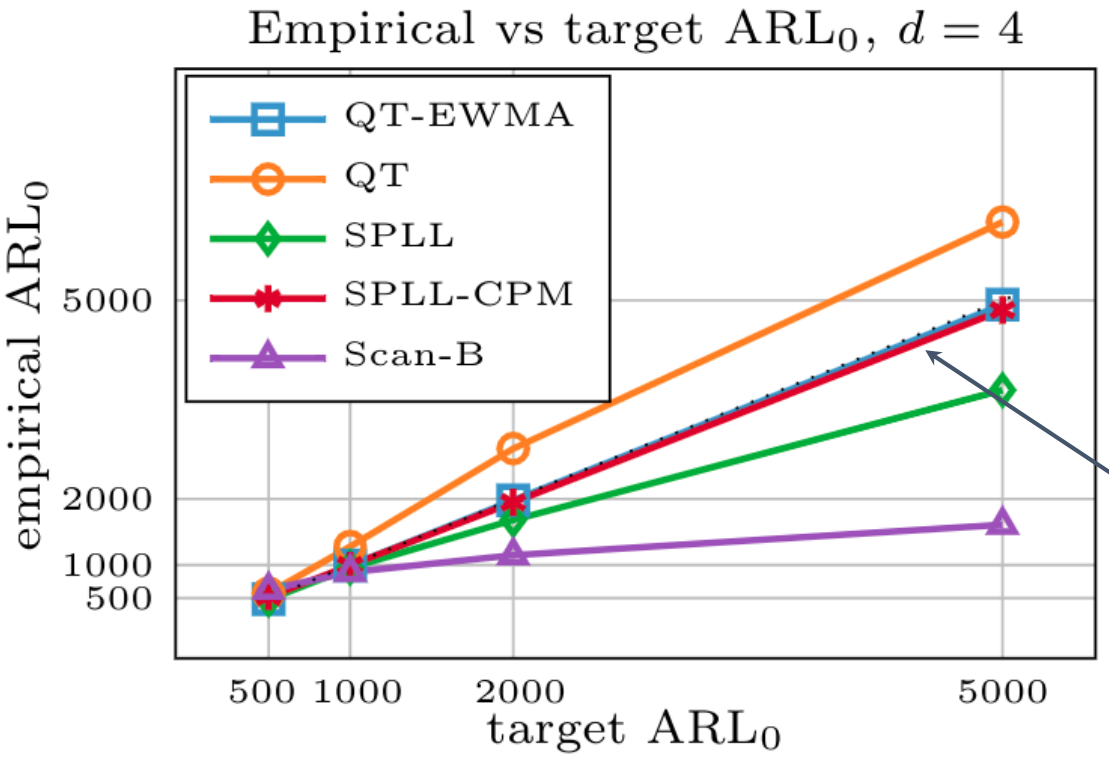
Control of False Alarm Rates

When the **false alarms probability is constant**, we can compute the probability of having a false alarm before a given time by the **Geometric sum**:

$$\mathbb{P}(t^* \leq t) = \sum_{k=1}^{t} \alpha(1-\alpha)^{k-1} = 1 - (1-\alpha)^t$$

Thus, QT-EWMA can also **control false alarm rates**

Experiments: synthetic Gaussian data



We set different ARL_0 values and measure the **empirical ARL_0** of QT-EWMA and the other considered methods

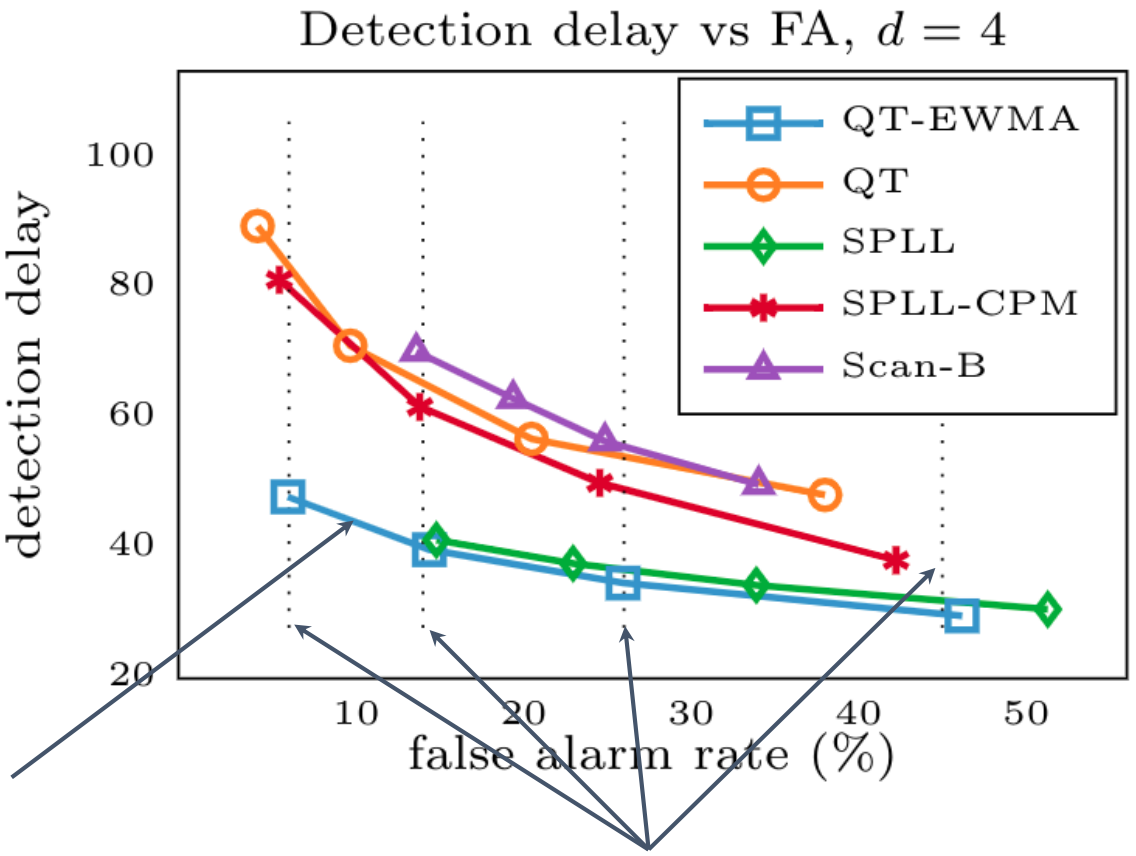
Goal: maintain the target ARL_0 (i.e. approach the diagonal)

[SPLL] L. Kuncheva “Change Detection in Streaming Multivariate Data Using Likelihood Detectors”, IEEE Transactions on Knowledge and Data Engineering, 2011

[Scan-B] S. Li et al. “M-Statistic for Kernel Change-Point Detection”, Advances in Neural Information Processing Systems, 2015

Experiments: synthetic Gaussian data

We set different ARL_0 values and observe the **trade-off** between **detection delay** and **false alarm rate**



Goal 1: minimize the detection delay

Goal 2: maintain the target false alarm rates depending on the target ARL_0

Experiments

Experiments

Considered Methods

- Pearson's Dist. Free
- TV Dist. Free
- Pearson Asymptotic
- TV Bootstrap

Uniform histograms by QuantTree $\pi_i = 1/K$

- Voronoi
- Density Tree

Non Uniform histograms

- Parametric

Gaussians Fitted on TR

Experiments

Considered Methods

- Pearson's Dist. Free
- TV Dist. Free

Thresholds estimated by $W \sim U(0,1)$

- Pearson Asymptotic

- TV Bootstrap
- Voronoi
- Density Tree

Thresholds estimate through bootstrap

- Parametric

T-test on the log-likelihood

The data

Synthetic data:

- Gaussians ϕ_0 is randomly defined having dimension $d = \{2, 8, 32, 64\}$
- Changes $\phi_0 \rightarrow \phi_1$ are such that

$$\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v}) \text{ and } \mathrm{sKL}(\phi_0, \phi_1) = 1$$

This was done to preserve the same distance between pre- and post-change distribution in any dimension. We used CCM-framework

Real-world data:

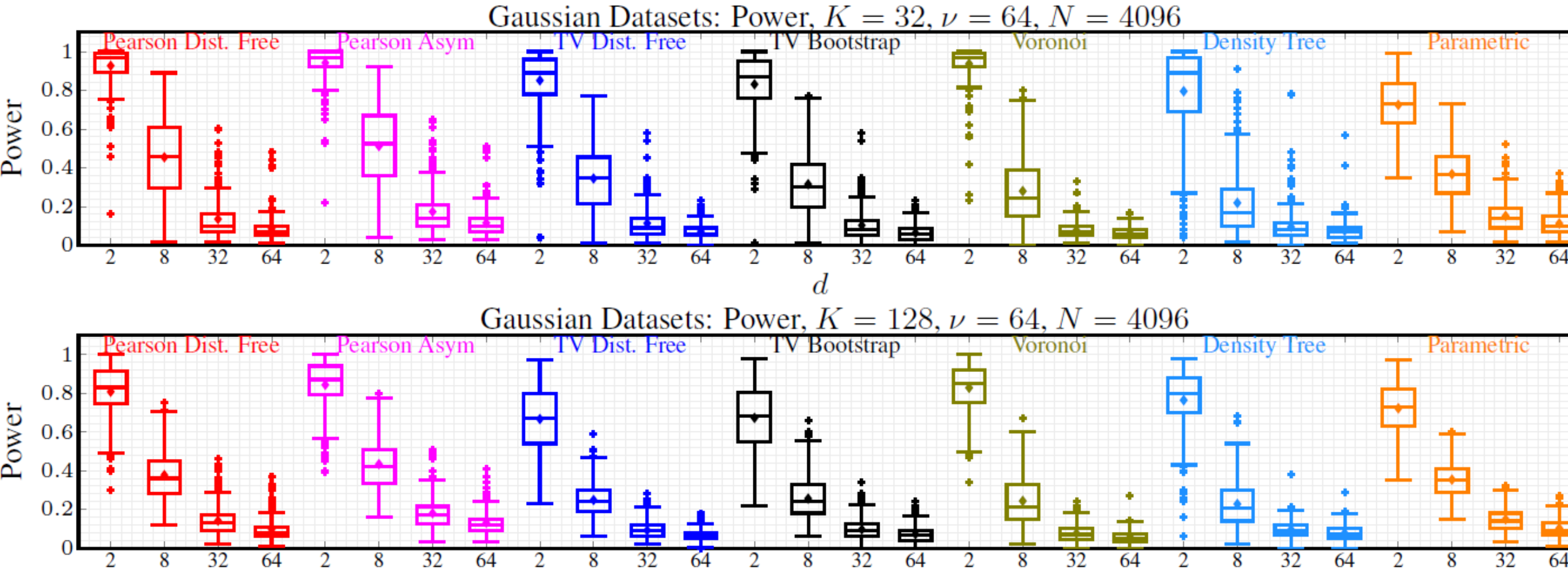
- "particle" (d = 50), "protein" (d = 9), "sensorless" (d = 48), "credit" (d = 29). These have been standardized.
- Changes introduced by a random shift (no control on the magnitude)

Test data:

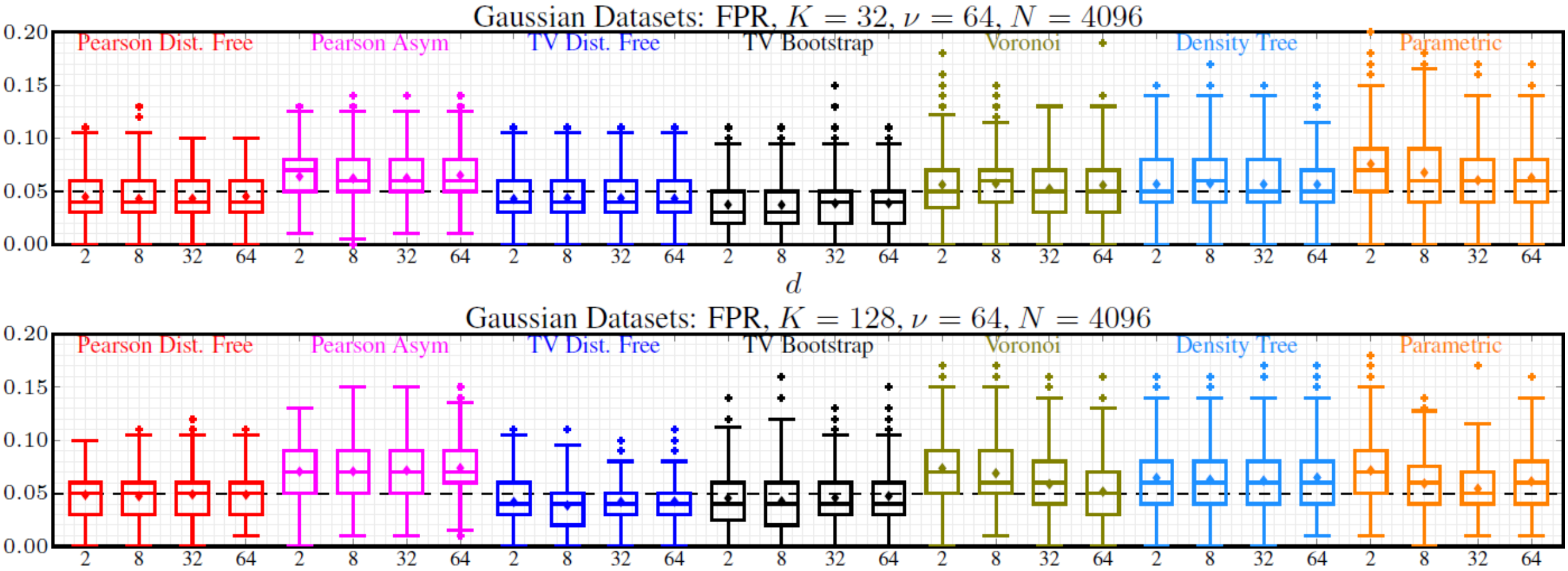
100 windows W_0 and W_1 of 64 (256) samples, generated by ϕ_0 and ϕ_1 to compute power and FPR

Carrera, D. and Boracchi, G. *"Generating high-dimensional datastreams for change detection"*. Big Data Research, 2017.

Experiments on Gaussian Dataset: The POWER

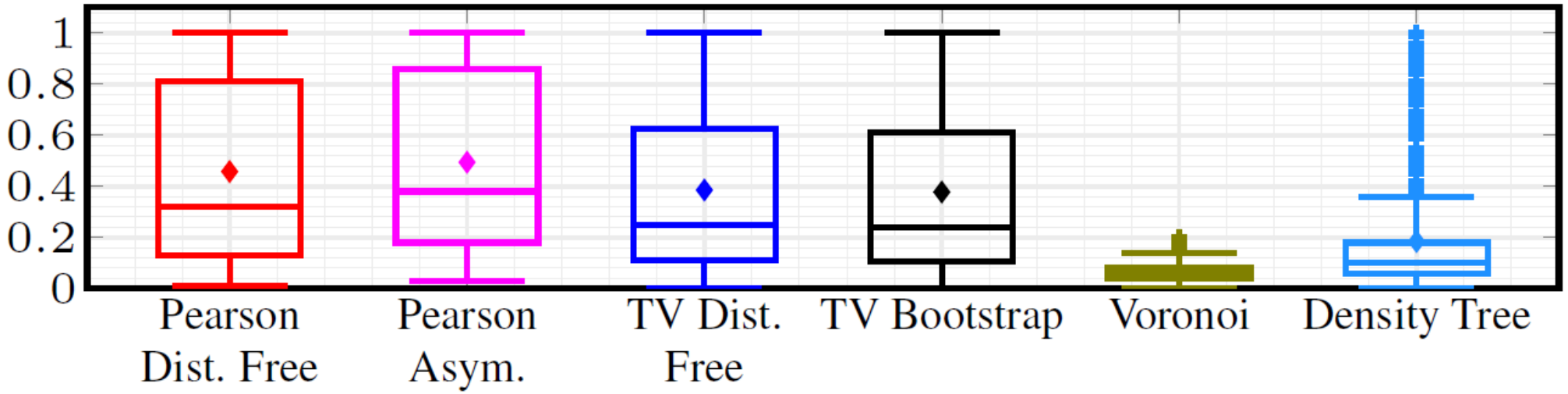


Experiments on Gaussian Dataset: The FPR (target value 0.05)

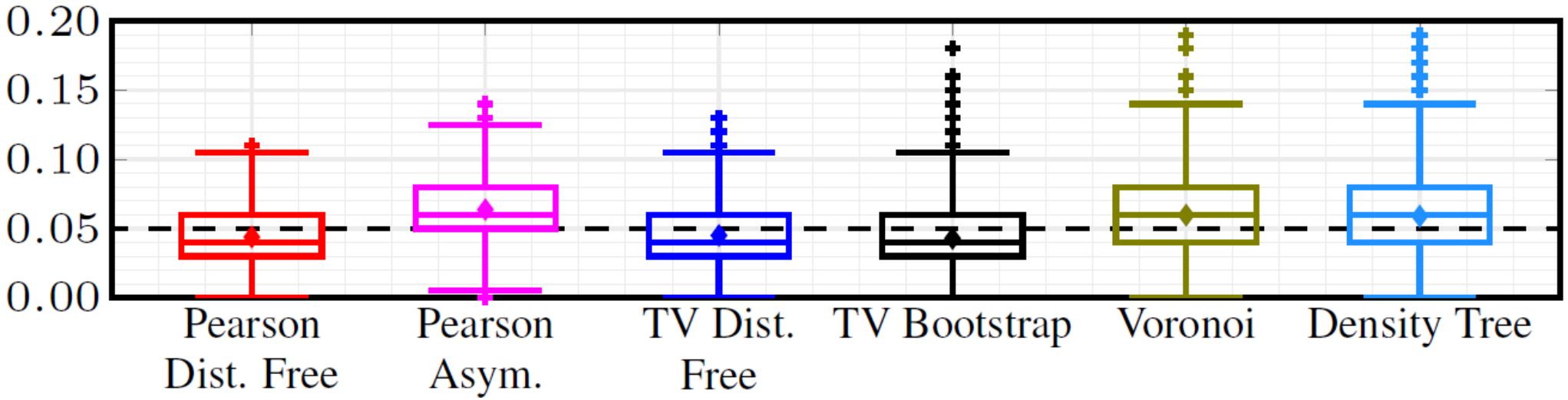


Experiments on real-World Datasets

Real World Datasets: Power, $K = 32$, $\nu = 64$, $N = 4096$



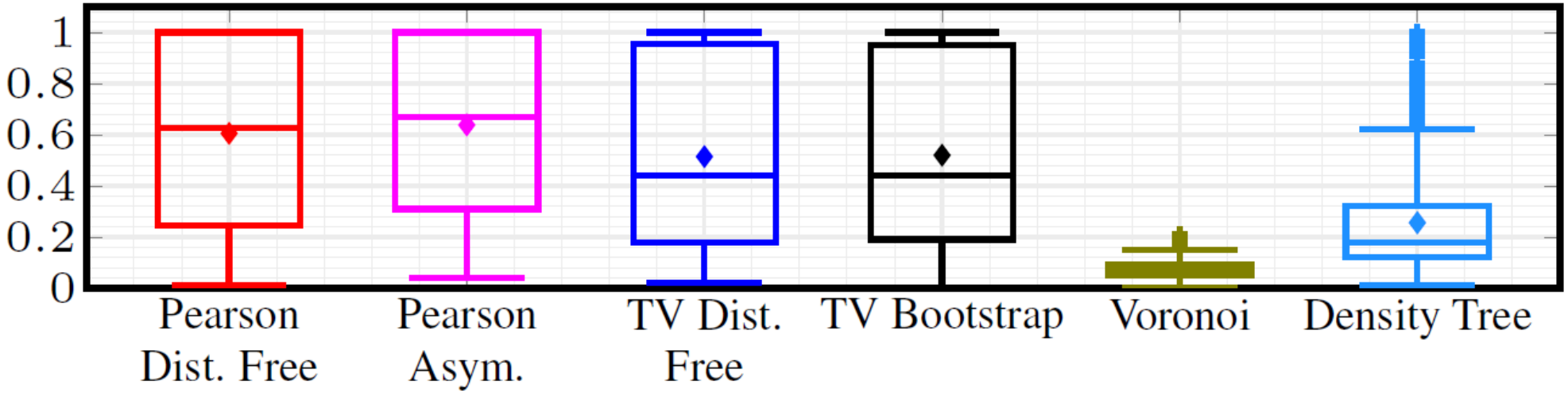
Real World Datasets: FPR, $K = 32$, $\nu = 64$, $N = 4096$



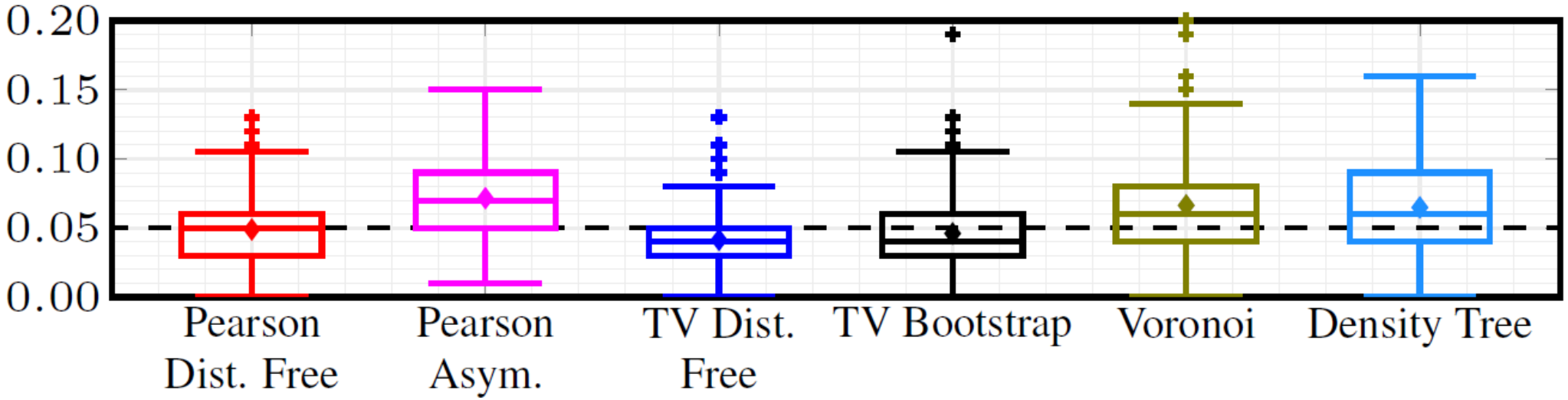
Boracchi, Trovò

Experiments on real-World Datasets

Real World Datasets: Power, $K = 128$, $\nu = 64$, $N = 4096$



Real World Datasets: FPR, $K = 128$, $\nu = 64$, $N = 4096$



A few comments (FPR)

QuantTrees provide **very accurate thresholds**, yielding FPR below the target value

- **TV** is below the reference value since it assume **less distinct values**
- Increasing K attenuates this problem

Alternative solutions do not control FPR effectively,

- Asymptotic approximation for Pearson is not accurate
- Log-likelihood is not Gaussian distributed

Results are consistent between synthetic and real data

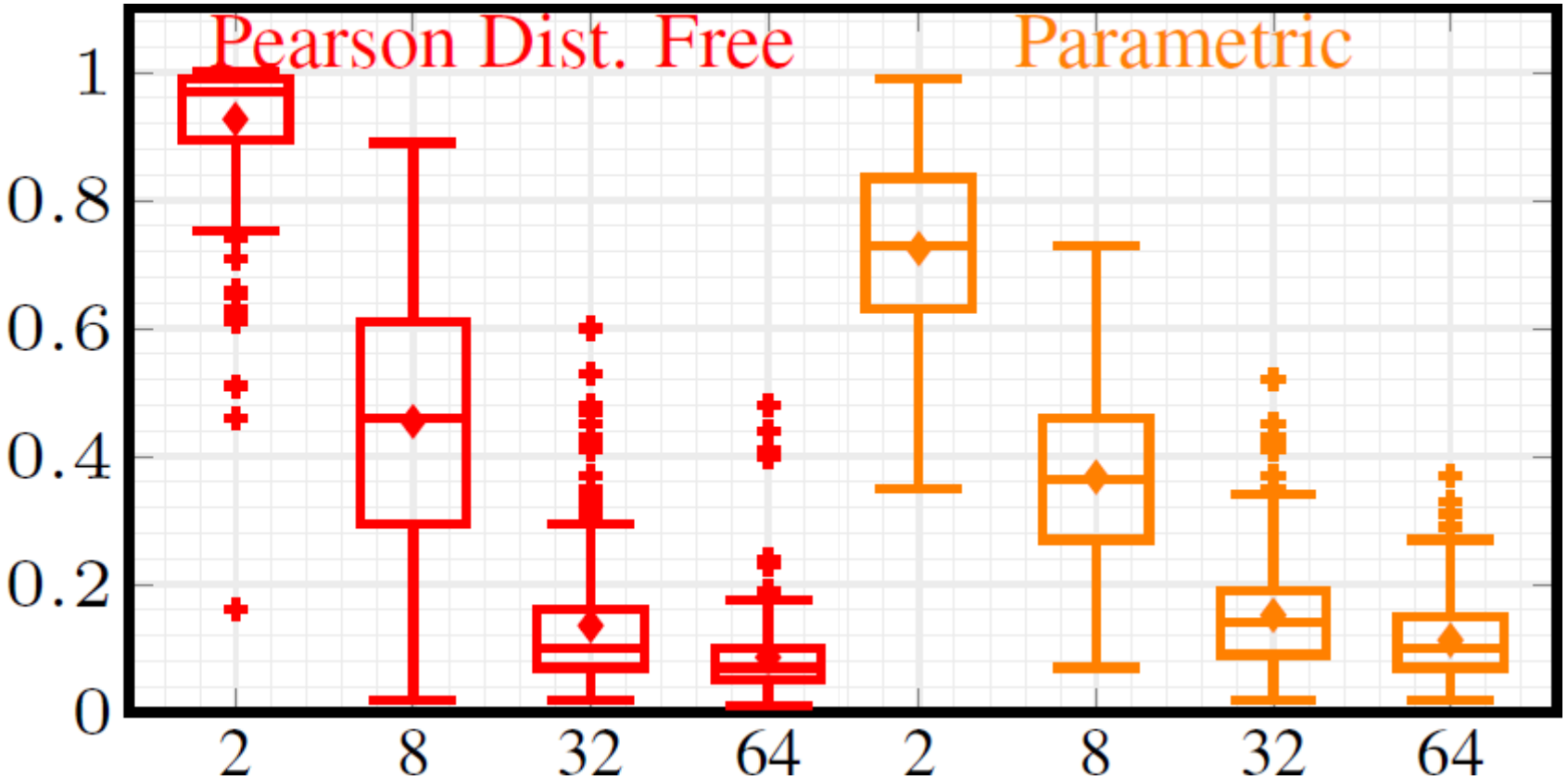
A few comments (POWER)

- Effective detection in high dimensional data with relatively few bins
- Uniform histograms from QuantTrees are more effective than other histograms

A few comments (POWER)

- Effective detection in high dimensional data with relatively few bins
- Uniform histograms from QuantTrees are more effective than other histograms

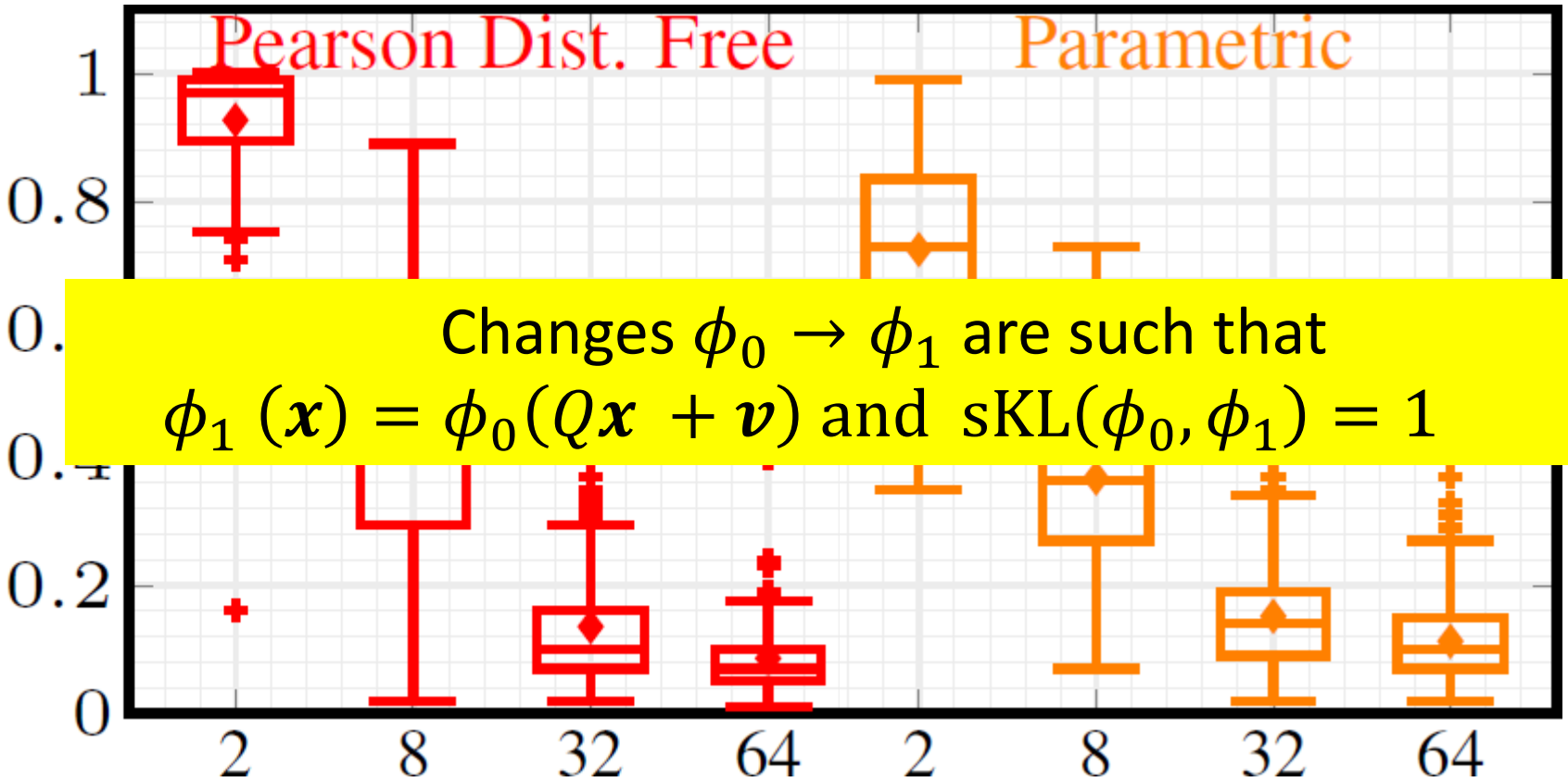
What about this problem?



A few comments (POWER)

- Effective detection in high dimensional data with relatively few bins
- Uniform histograms from QuantTrees are more effective than other histograms

What about this problem?



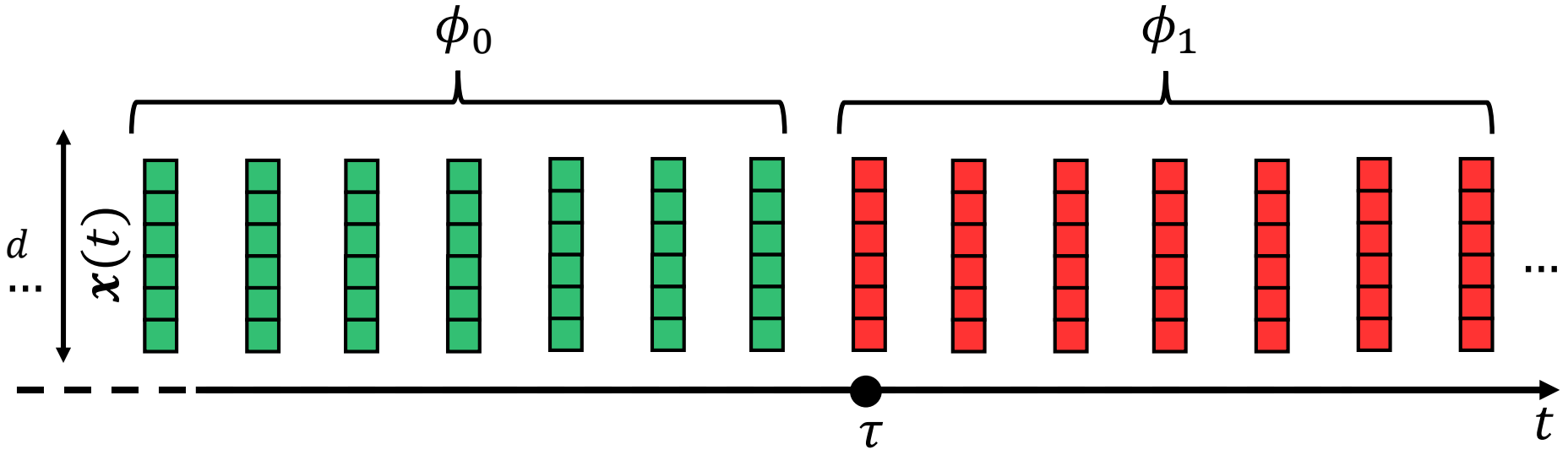
Changes $\phi_0 \rightarrow \phi_1$ are such that
$\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ and $\mathrm{sKL}(\phi_0, \phi_1) = 1$

Detectability loss in high-dimensional data

How data dimension affects monitoring the Log-likelihood

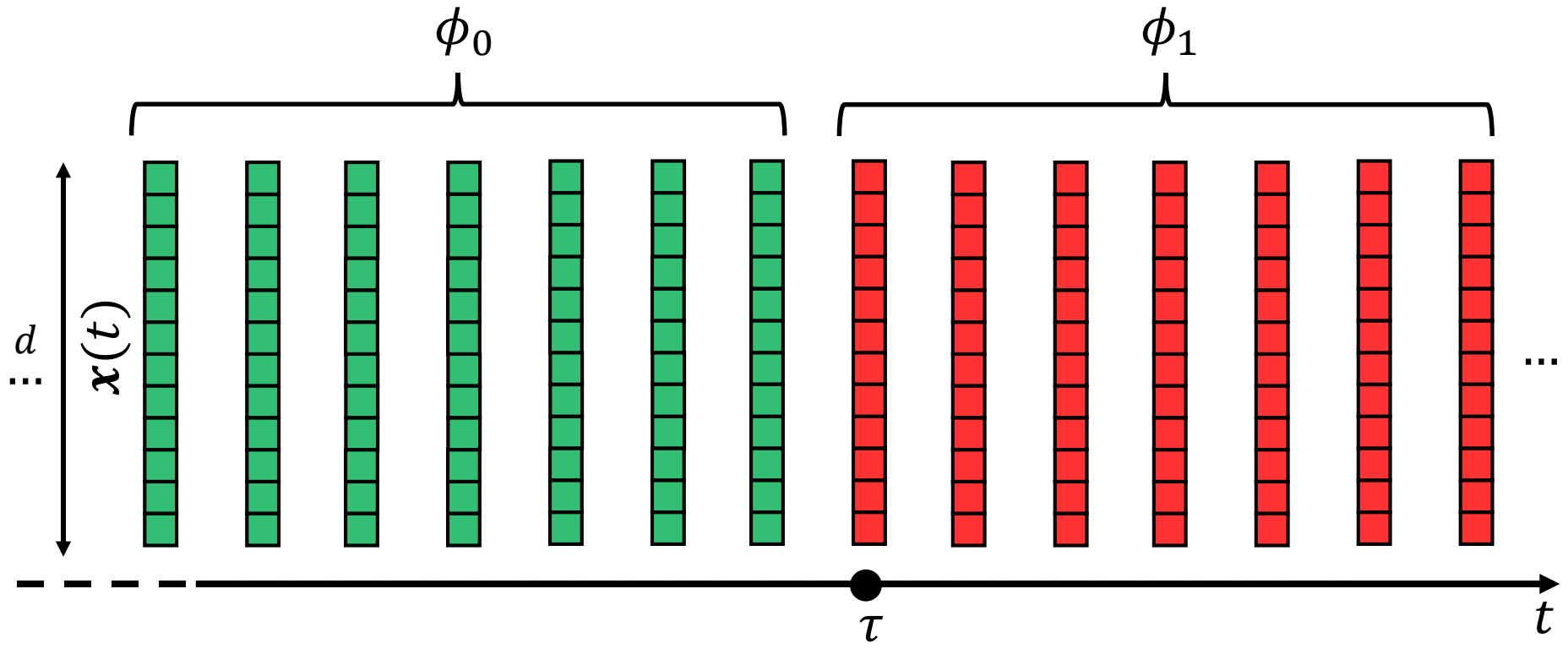
Our Goal

*Study how the **data dimension** d **influences** the **change detectability**, i.e., how difficult is to solve change/anomaly detection problems*



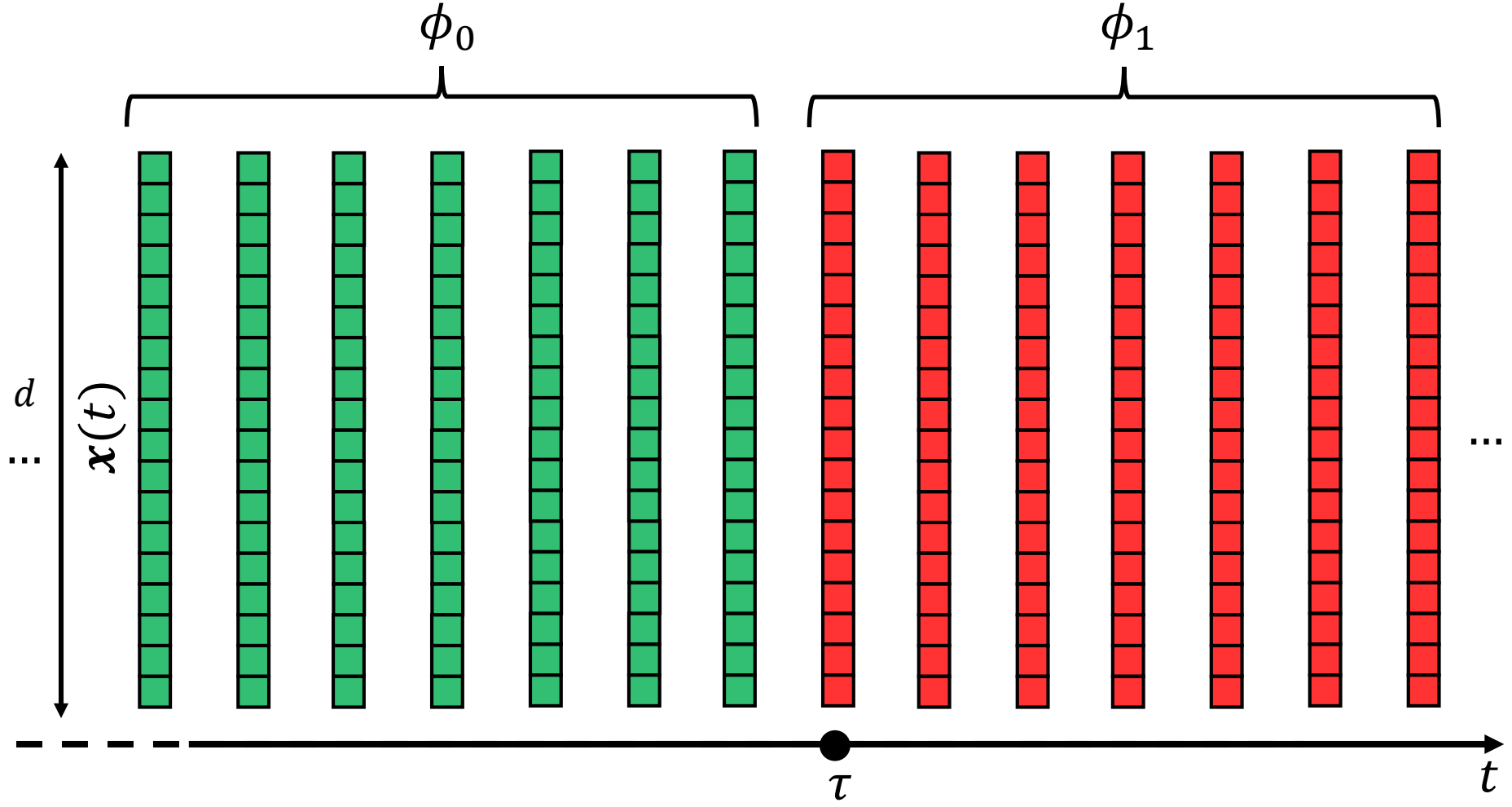
Our Goal

*Study how the **data dimension** d **influences** the **change detectability**, i.e., how difficult is to solve change/anomaly detection problems*



Our Goal

*Study how the **data dimension** d **influences** the **change detectability**, i.e., how difficult is to solve change/anomaly detection problems*



Our Approach

To study the impact of the **sole data dimension** d in **change-detection problems**:

1. Consider a **change-detection approach**
2. Define a measure of **change detectability** that well correlates with traditional performance measures
3. Define a measure of **change magnitude** that refers only to differences between ϕ_0 and ϕ_1

Our Result

We show there is a **detectability loss** problem, i.e. that change **detectability** steadily **decreases** when d increases.

Detectability loss is shown by:

- Analytical derivations: when ϕ_0 and ϕ_1 are **Gaussians**
- Empirical analysis on real data: measuring the **power of hypothesis tests**

Roadmap to detectability loss

Preliminaries:

- The change-detection approach
- The measure of change detectability
- The change magnitude

The *detectability loss*

- Analytical results
- Empirical analysis

C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016,

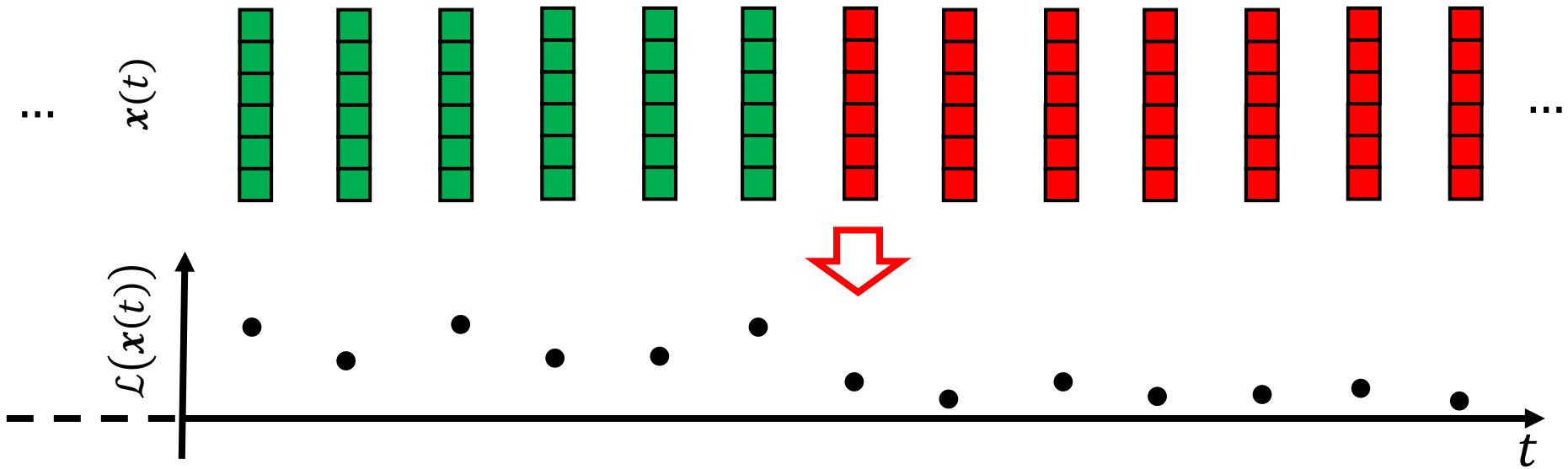
How? Monitoring the Log-likelihood

A typical approach to monitor the log-likelihood

1. During training, estimate $\hat{\phi}_0$ from TR
2. During testing, compute

$$\mathcal{L}(\boldsymbol{x}(t)) = -\log(\hat{\phi}_0(\boldsymbol{x}(t)))$$

3. Monitor $\{\mathcal{L}(\boldsymbol{x}(t)),\ t = 1, \ldots\}$



Roadmap to detectability loss

Preliminaries:

- The change-detection approach
- The measure of change detectability
- The change magnitude

The *detectability loss*

- Analytical results
- Empirical analysis

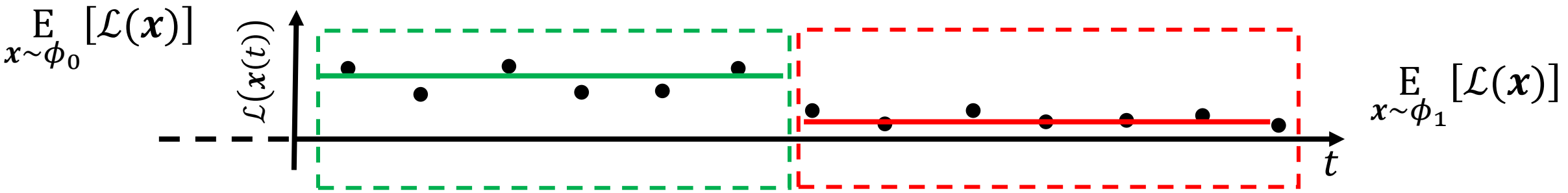
C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016,

The Change Detectability

The ***Signal to Noise Ratio of the change***

$$\mathrm{SNR}(\phi_0 \to \phi_1) = \frac{\left(\underset{x\sim\phi_0}{\mathrm{E}} [\mathcal{L}(\boldsymbol{x})] - \underset{x\sim\phi_1}{\mathrm{E}} [\mathcal{L}(\boldsymbol{x})] \right)^2}{\underset{x\sim\phi_0}{\mathrm{var}} [\mathcal{L}(\boldsymbol{x})] + \underset{x\sim\phi_1}{\mathrm{var}} [\mathcal{L}(\boldsymbol{x})]}$$

measures the extent to which $\phi_0 \to \phi_1$ is **detectable by statistical tools** designed to **detect changes** in $\mathrm{E}[\mathcal{L}(\boldsymbol{x})]$



Roadmap to detectability loss

Preliminaries:

- The change-detection approach
- The measure of change detectability
- The change magnitude

The *detectability loss*

- Analytical results
- Empirical analysis

C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016,

The Change Magnitude

We measure the **magnitude of a change** $\phi_0 \rightarrow \phi_1$ by the *symmetric Kullback-Leibler divergence*

$$\mathrm{sKL}(\phi_0, \phi_1) = \mathrm{KL}(\phi_0, \phi_1) + \mathrm{KL}(\phi_1, \phi_0) =$$
$$= \int \log\left(\frac{\phi_0(\boldsymbol{x})}{\phi_1(\boldsymbol{x})}\right)\phi_0(\boldsymbol{x})d\boldsymbol{x} + \int \log\left(\frac{\phi_1(\boldsymbol{x})}{\phi_0(\boldsymbol{x})}\right)\phi_1(\boldsymbol{x})d\boldsymbol{x}$$

In practice, **large values** of $\mathrm{sKL}(\phi_0, \phi_1)$ correspond to **changes** $\phi_0 \rightarrow \phi_1$that are very apparent, since $\mathrm{sKL}(\phi_0, \phi_1)$ identifies an upperbound of the power of hypothesis tests designed to detect either $\phi_0 \rightarrow \phi_1$ or $\phi_1 \rightarrow \phi_0$

T. Dasu, K. Shankar, S. Venkatasubramanian, K. Yi, *"An information-theoretic approach to detecting changes in multi-dimensional data streams"* In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications, 2006

Our Approach

To study the impact of the **sole data dimension** d in **change-detection problems** we need to:

1. Consider a **change-detection approach**
2. Define a measure of **change detectability** that well correlates with traditional performance measures
3. Define a measure of **change magnitude** that refers only to differences between ϕ_0 and ϕ_1

Our goal (reformulated):
Studying how the **change detectability** $\mathrm{SNR}(\phi_0 \to \phi_1)$ **varies** in **change-detection problems** that have

- **different data dimensions** d
- **constant change magnitude** $\mathrm{sKL}(\phi_0, \phi_1)$

C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016,

Roadmap to detectability loss

Preliminaries:

- The change-detection approach
- The measure of change detectability
- The change magnitude

The *detectability loss*

- Analytical results
- Empirical analysis

The Detectability Loss

Theorem (IJCAI16)

Let $\phi_0 = \mathcal{N}(\mu_0, \Sigma_0)$ *and let* $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ *where* $Q \in \mathbb{R}^{d \times d}$ *and orthogonal ,* $\boldsymbol{v} \in \mathbb{R}^d$*, then*

$$\mathrm{SNR}(\phi_0 \to \phi_1) < \frac{C}{d}$$

Where C is a constant that depends only on $\mathrm{sKL}(\phi_0, \phi_1)$

C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016,

The Detectability Loss: Remarks

Theorem

Let $\phi_0 = \mathcal{N}(\mu_0, \Sigma_0)$ *and let* $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ *where* $Q \in \mathbb{R}^{d \times d}$ *and orthogonal ,* $\boldsymbol{v} \in \mathbb{R}^d$*, then*

$$\mathrm{SNR}(\phi_0 \to \phi_1) < \frac{C}{d}$$

where C *is a constant that depends only on* $\mathrm{sKL}(\phi_0, \phi_1)$

Remarks:

- Changes of a given magnitude, $\mathrm{sKL}(\phi_0, \phi_1)$, become more difficult to detect when d increases
- DL does not depend on the change parameters
- DL does not depend on the specific detection rule
- DL does not depend on estimation errors on $\hat{\phi}_0$

The Detectability Loss: The Change Model

Theorem

Let $\phi_0 = \mathcal{N}(\mu_0, \Sigma_0)$ *and let* $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ *where* $Q \in \mathbb{R}^{d\times d}$ *and orthogonal ,* $\boldsymbol{v} \in \mathbb{R}^d$*, then*

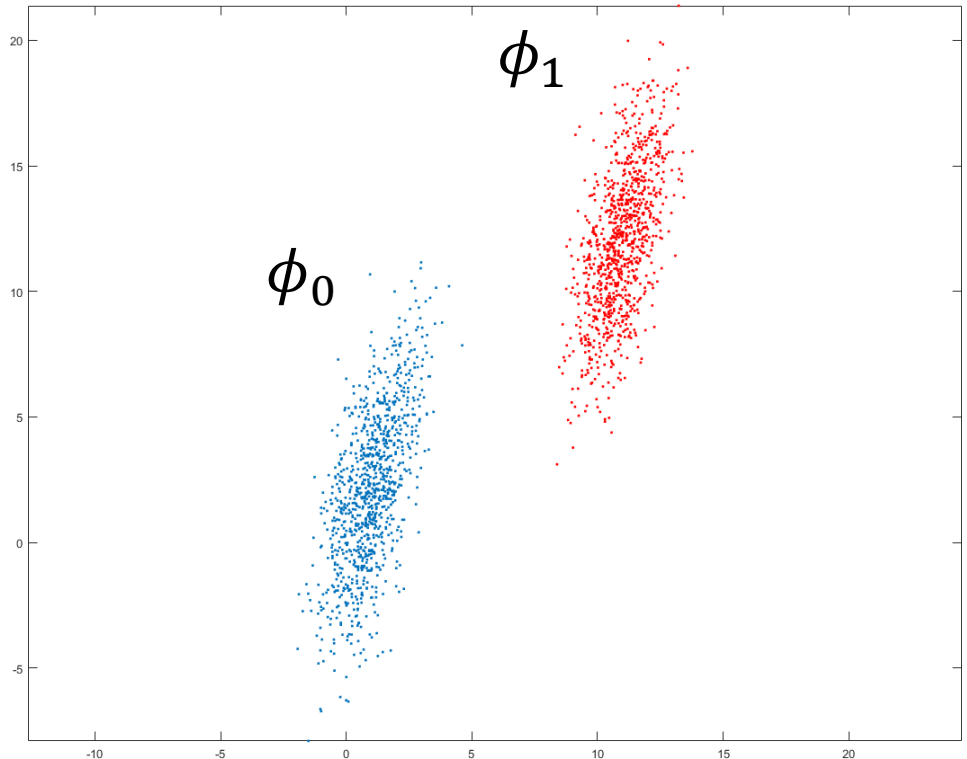
$$\mathrm{SNR}(\phi_0 \to \phi_1) < \frac{C}{d}$$

where C *is a constant that depends only on* $\mathrm{sKL}(\phi_0, \phi_1)$

The Detectability Loss: The Change Model

The change model $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ includes:

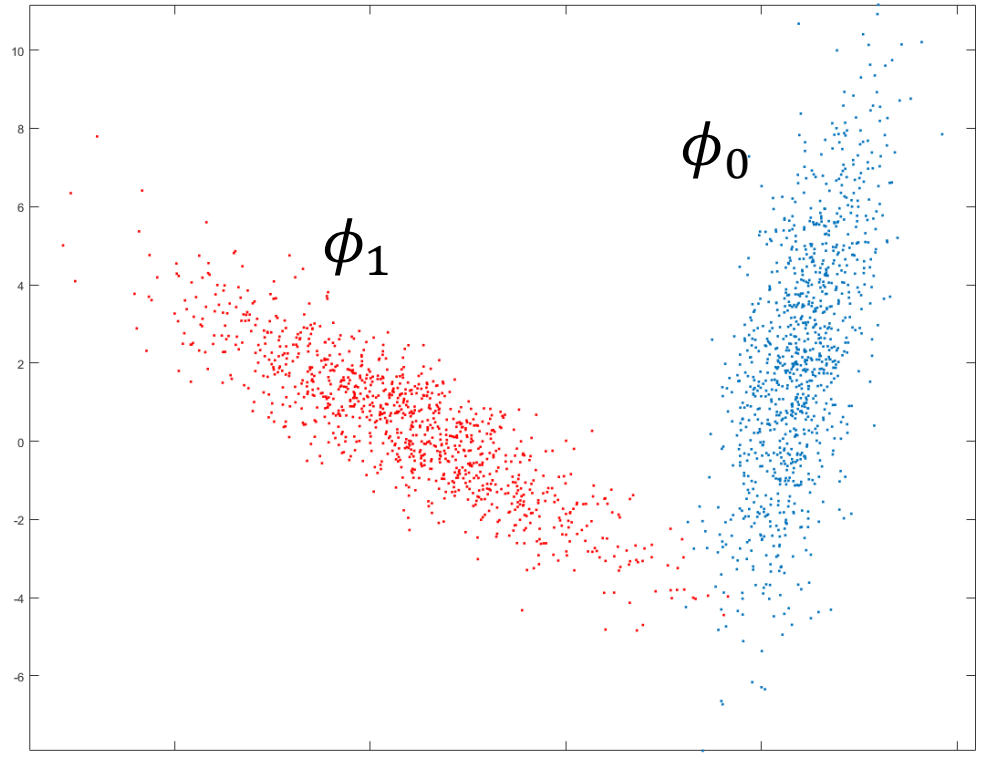
- Changes in the location of ϕ_0 (i.e, $+\boldsymbol{v}$)



The Detectability Loss: The Change Model

The change model $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ includes:

- Changes in the location of ϕ_0 (i.e, $+\boldsymbol{v}$)
- Changes in the correlation of $\boldsymbol{x}$ (i.e, $Q\boldsymbol{x}$)



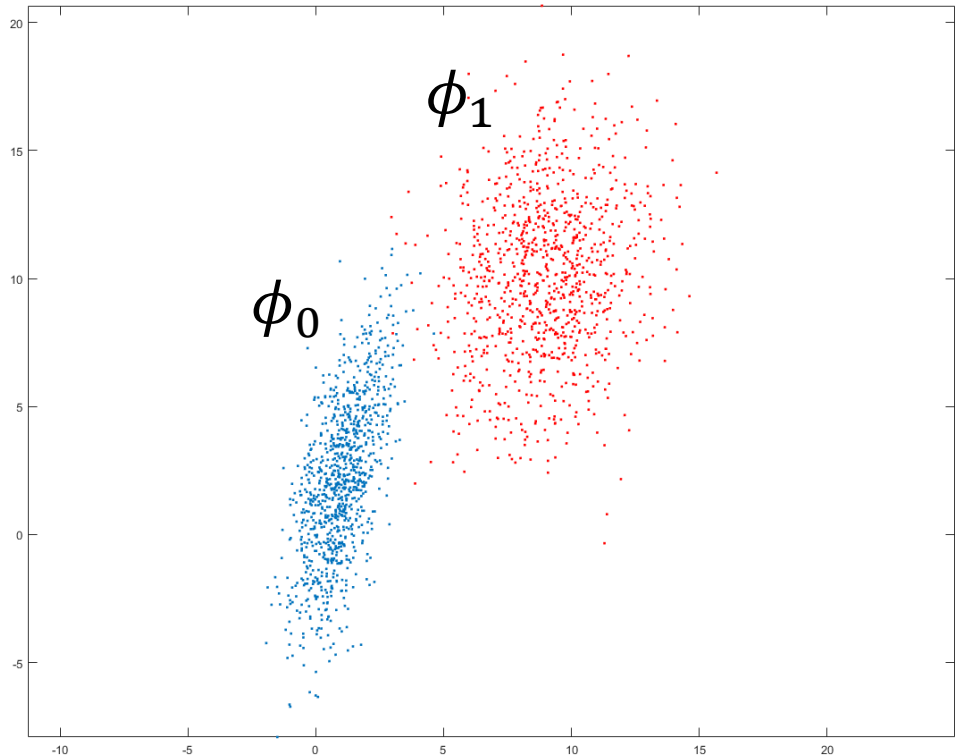
Boracchi, Trovò

The Detectability Loss: The Change Model

The change model $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ includes:

- Changes in the location of ϕ_0 (i.e, $+\boldsymbol{v}$)
- Changes in the correlation of $\boldsymbol{x}$ (i.e, $Q\boldsymbol{x}$)

It does not include changes in the scale of ϕ_0 that can be however detected monitoring $||\boldsymbol{x}||$



Boracchi, Trovò

The Detectability Loss: The Gaussian Assumption

Theorem

Let $\phi_0 = \mathcal{N}(\mu_0, \Sigma_0)$ *and let* $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ *where* $Q \in \mathbb{R}^{d \times d}$ *and orthogonal ,* $\boldsymbol{v} \in \mathbb{R}^d$*, then*

$$\mathrm{SNR}(\phi_0 \to \phi_1) < \frac{C}{d}$$

where C *is a constant that depends only on* $\mathrm{sKL}(\phi_0, \phi_1)$

The Detectability Loss: The Gaussian Assumption

Assuming $\phi_0 = \mathcal{N}(\mu_0, \Sigma_0)$ looks like a severe limitation.

- Other distributions are not easy to handle analytically
- We can prove that DL occurs **also in random variables having independent components**
- The result have been **empirically confirmed** in case of approximations of $\mathcal{L}(\cdot)$ typically used **for Gaussian mixtures**

Roadmap to detectability loss

Preliminaries:

- The change-detection approach
- The measure of change detectability
- The change magnitude

The *detectability loss*

- Analytical results
- Empirical analysis

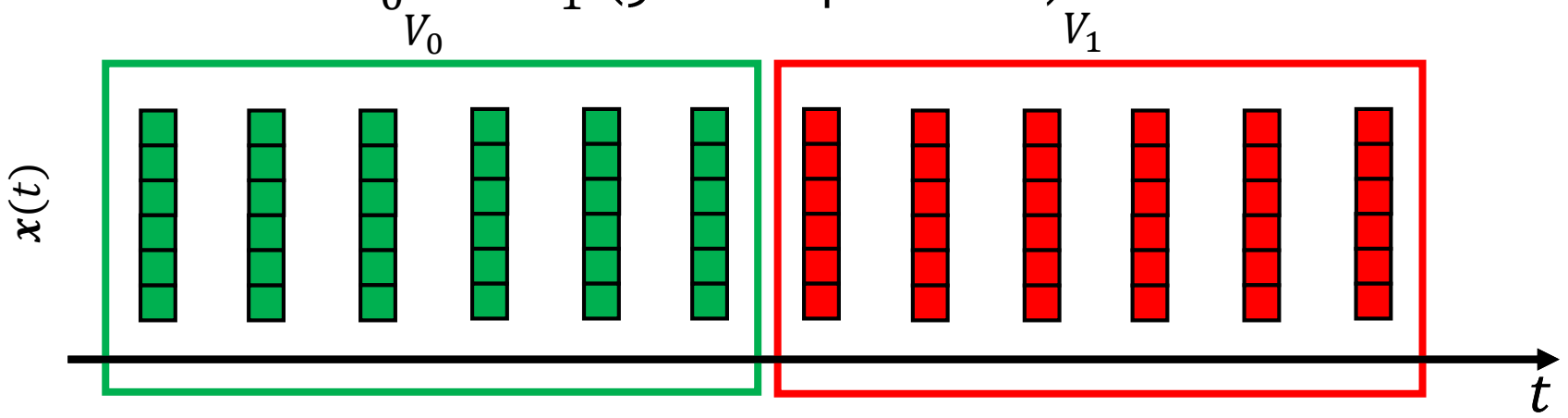
The Detectability Loss: Empirical Analysis

The data

- Synthetically generate streams with different dimensions d
- Estimate $\hat{\phi}_0$ by GM from a **stationary training set**
- In each stream we introduce $\phi_0 \to \phi_1$ such that

$$\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v}) \text{ and } \mathrm{sKL}(\phi_0, \phi_1) = 1$$

- **Test data: two windows** V_0 and V_1 (500 samples each) before and after the change.



Carrera, D. and Boracchi, G. *"Generating high-dimensional datastreams for change detection"*. Big Data Research, 2017.

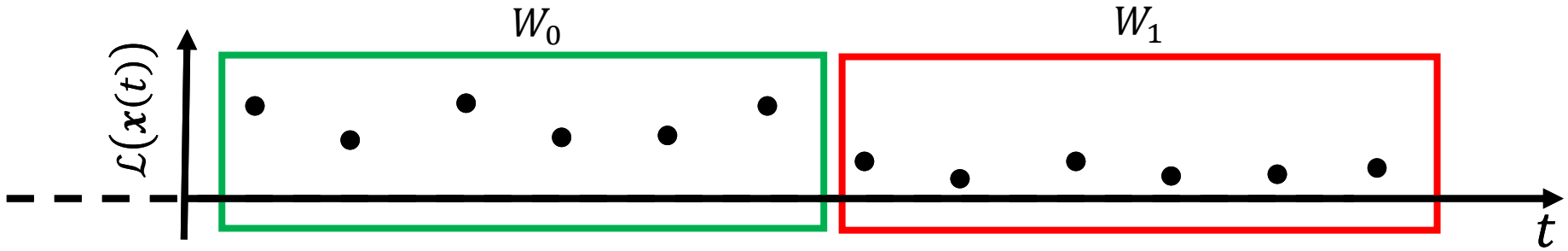
The Detectability Loss: Empirical Analysis

We measure the change-detectabiltity as:

- Compute $\mathcal{L}\left(\hat{\phi}_0(\boldsymbol{x})\right)$ from V_0 and V_1, obtaining W_0 and W_1
- **Compute a test statistic** $\mathcal{T}(W_0, W_1)$ to compare the two
- Detect a change by an **hypothesis test**

$$\mathcal{T}(W_0, W_1) \lessgtr h$$

  where h controls the amount of false positives
- Use the **power** of this **test** to assess change detectability



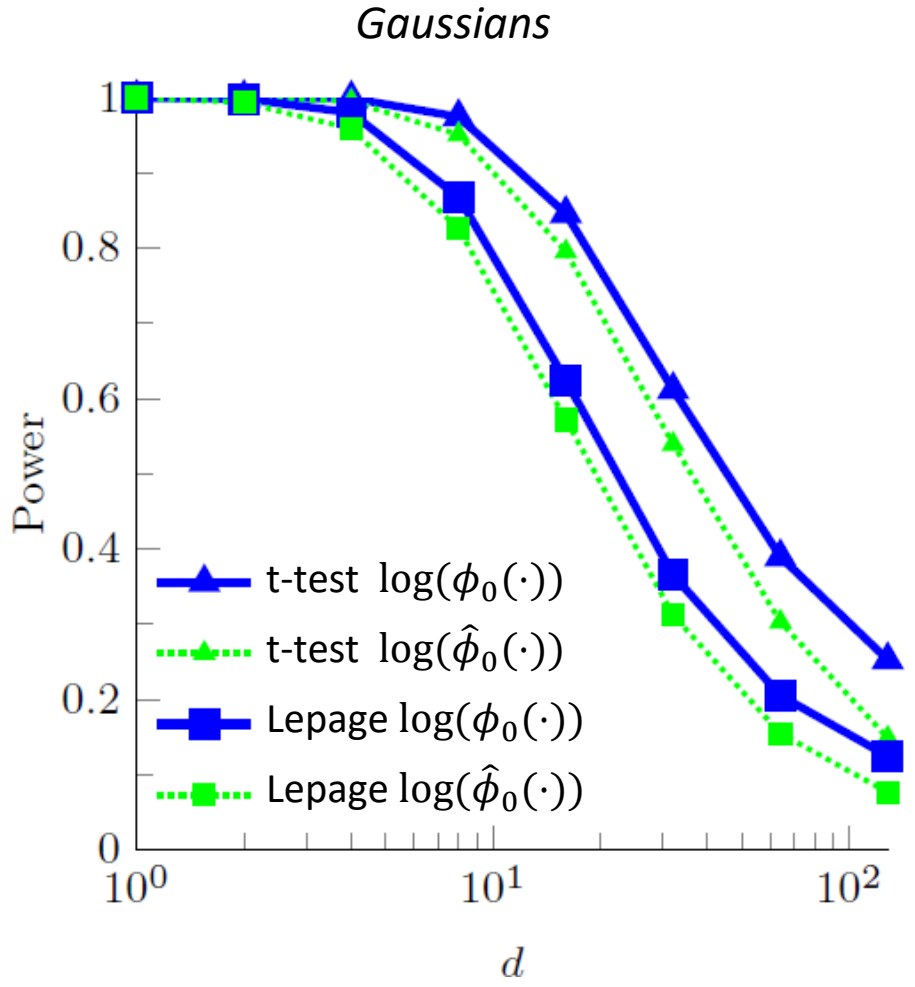
Power of Hypothesis Tests on Gaussian Streams

Remarks:

- ϕ_1 is defined analytically
- The t-test detects changes in the expectation of $\log(\phi_0(\cdot))$
- The Lepage test detects changes in the location and scale of $\log(\phi_0(\cdot))$

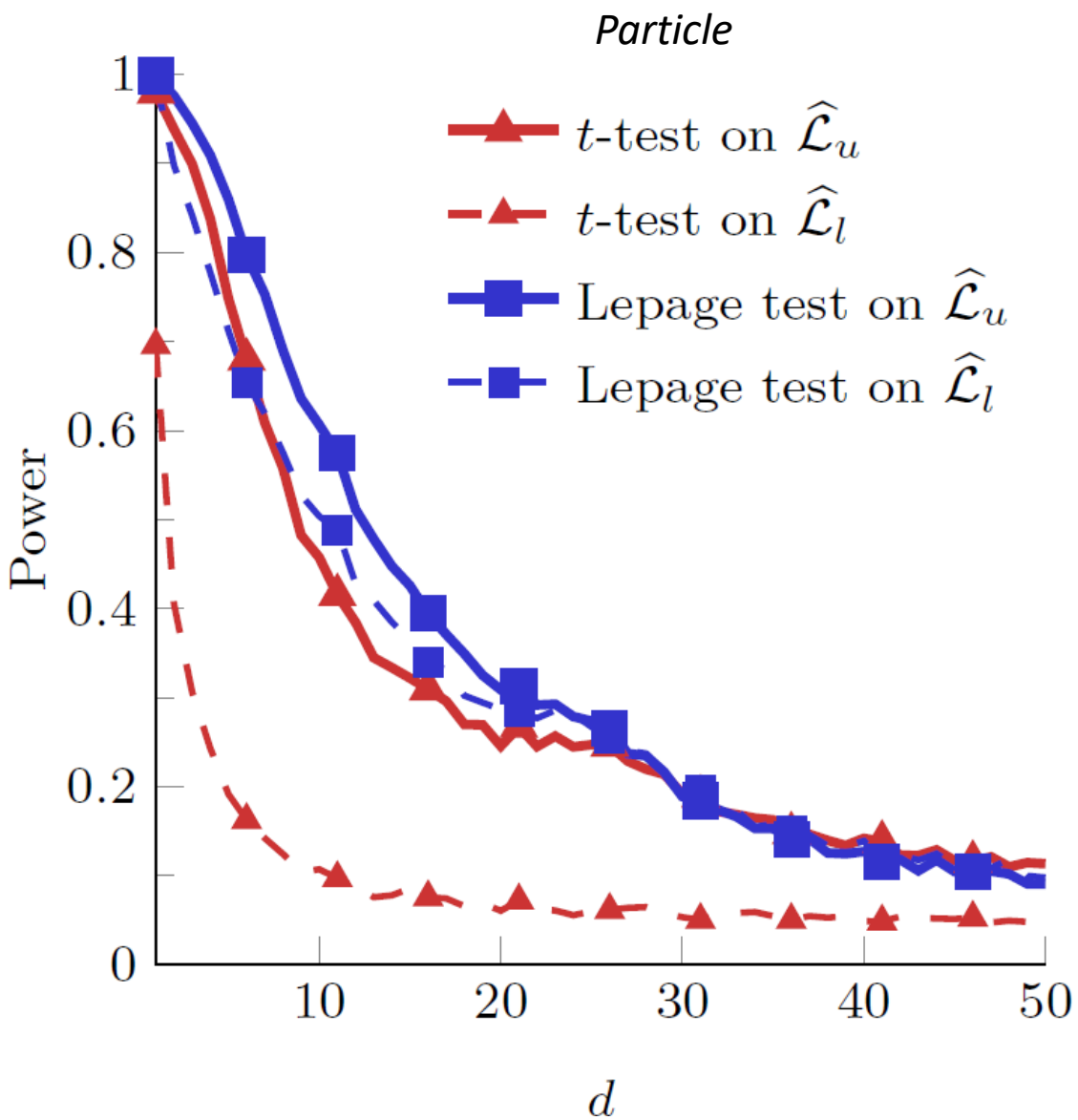
Results

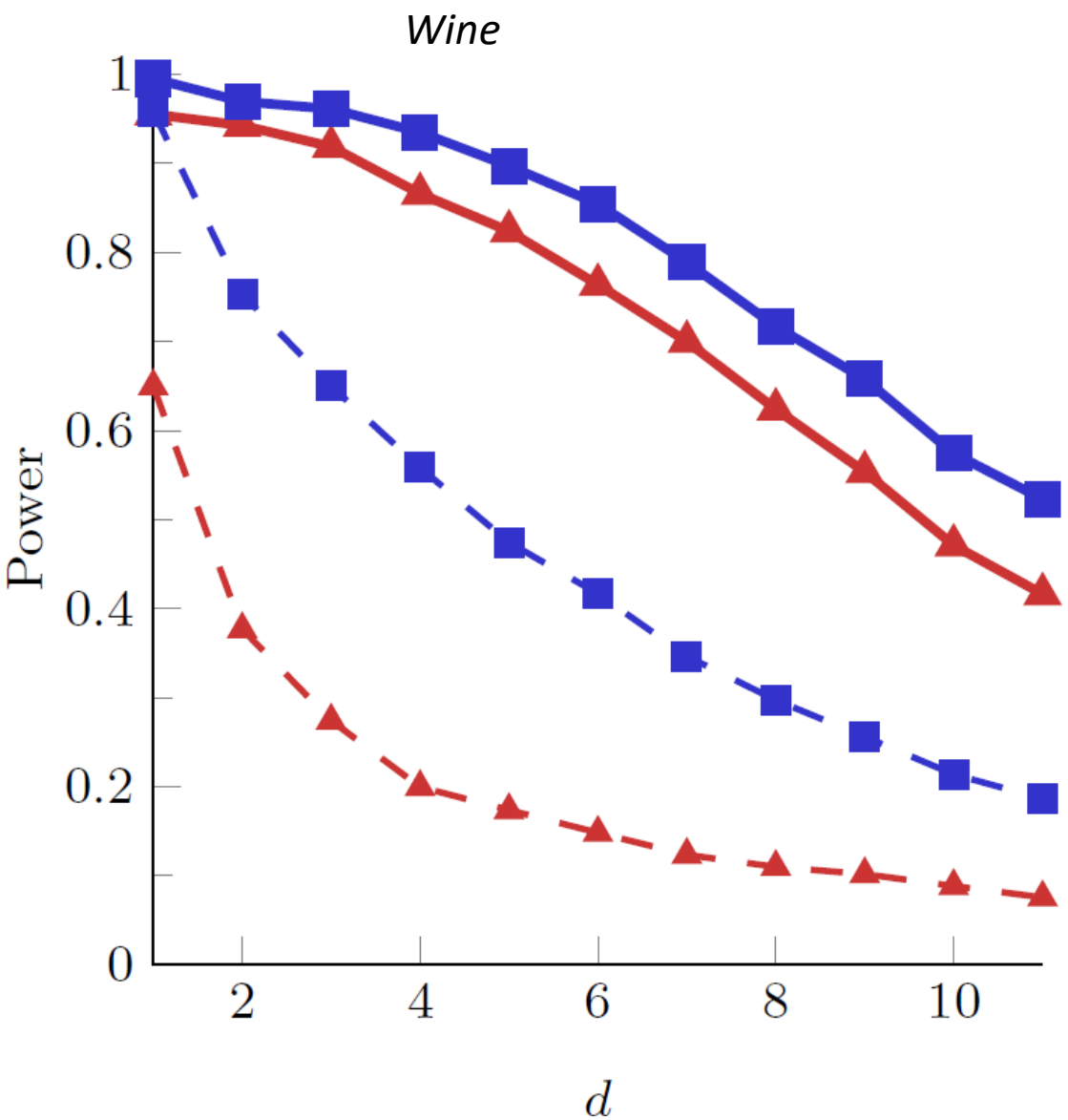
- The HT power decays with d: DL does not only concern the upperbound of SNR.
- DL is not due to estimation errors, but these make things worst.
- Also the power of the Lepage HT decreases, which indicates that the change is more difficult to detect even when monitoring the variance



C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016,

Power of Hypothesis Tests on UCI datasets





C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016,

The Power on Particle Dataset

Remarks:

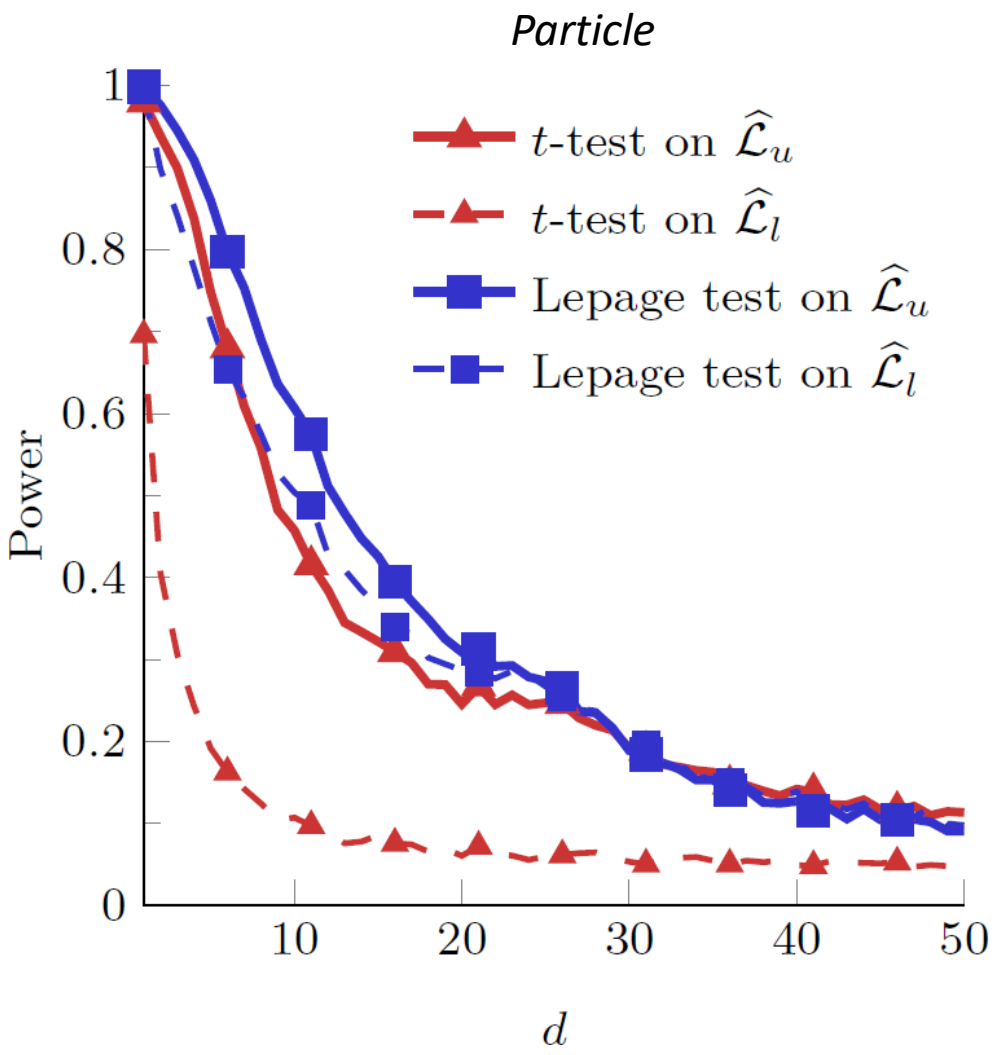
- ϕ_1 is defined through CCM a framework to control the change magnitude and yield $sKL(\phi_0, \phi_1) \approx 1$
- ϕ_0 is a Gaussian Mixture where k is selected by cross-validation

Approximated expression of $\mathcal{L}(\cdot)$ to prevent numerical approximations

Results:

DL occurs also in non-Gaussian data approximated by GM

DL is clearly visible at quite a low dimensions



C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016,

Hierarchical Change Detection

Hierarchical Change-detection tests

In nonparametric sequential monitoring it is convenient to

- **online sequential CDTs** for detection purposes
- **offline hypothesis tests** for validation purposes.

Alippi, C., Boracchi, G., Roveri, M. *"Hierarchical Change-Detection Tests"* TNNLS 2016 pp 1- 13

Hierarchical Change-detection tests

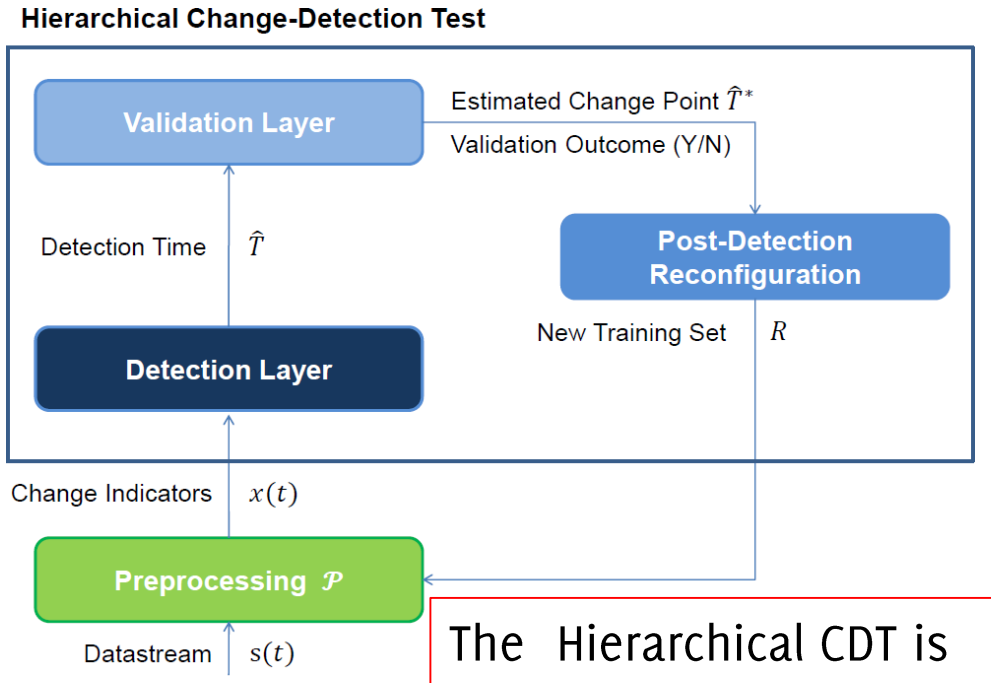
In nonparametric sequential monitoring it is convenient to

- **online sequential CDTs** for detection purposes
- **offline hypothesis tests** for validation purposes.

This results in two-layered (hierarchical) CDTs

Offline HT is activated to validate any detection

Online CDT detects process changes in the input datastream

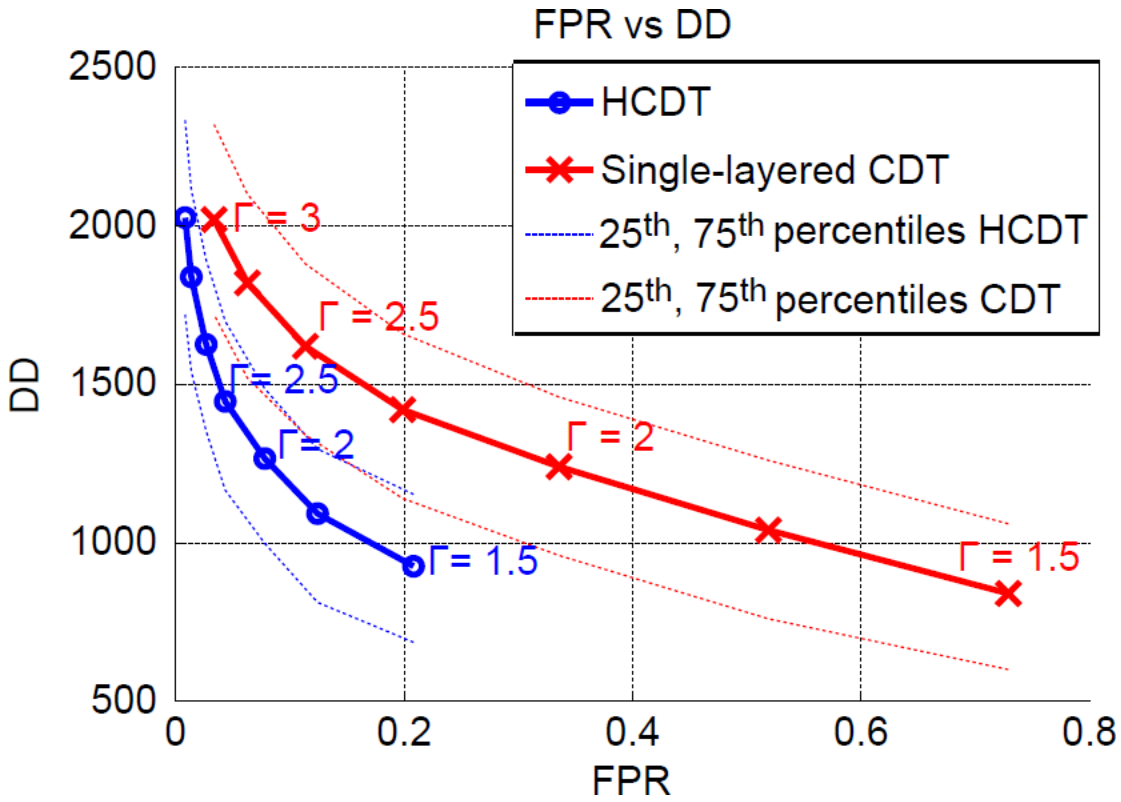


The Hierarchical CDT is automatically reconfigured

Alippi, C., Boracchi, G., Roveri, M. *"Hierarchical Change-Detection Tests"* TNNLS 2016 pp 1- 13

Hierarchical Change-detection tests

Hierarchical CDTs can achieve a far **more advantageous trade-off** between false-positive rate and detection delay **than their single-layered**, more traditional, **counterpart**.



Alippi, C., Boracchi, G., Roveri, M. *"Hierarchical Change-Detection Tests"* TNNLS 2016 pp 1- 13

Conclusions

Concluding Remarks

So far we have been addressing an hypothesis testing framework

- We expect the same results concerning power holds in sequential monitoring scheme that adopts the same statistics.
- However controlling ARL is more complicated and we are currently investigating that using QuantTrees

Controlling the change magnitude was essential for our study

A note on experimental practices

To correctly assess the **change-detection performance**, in particular when changing the input size, it is **necessary to control the magnitude of injected changes**

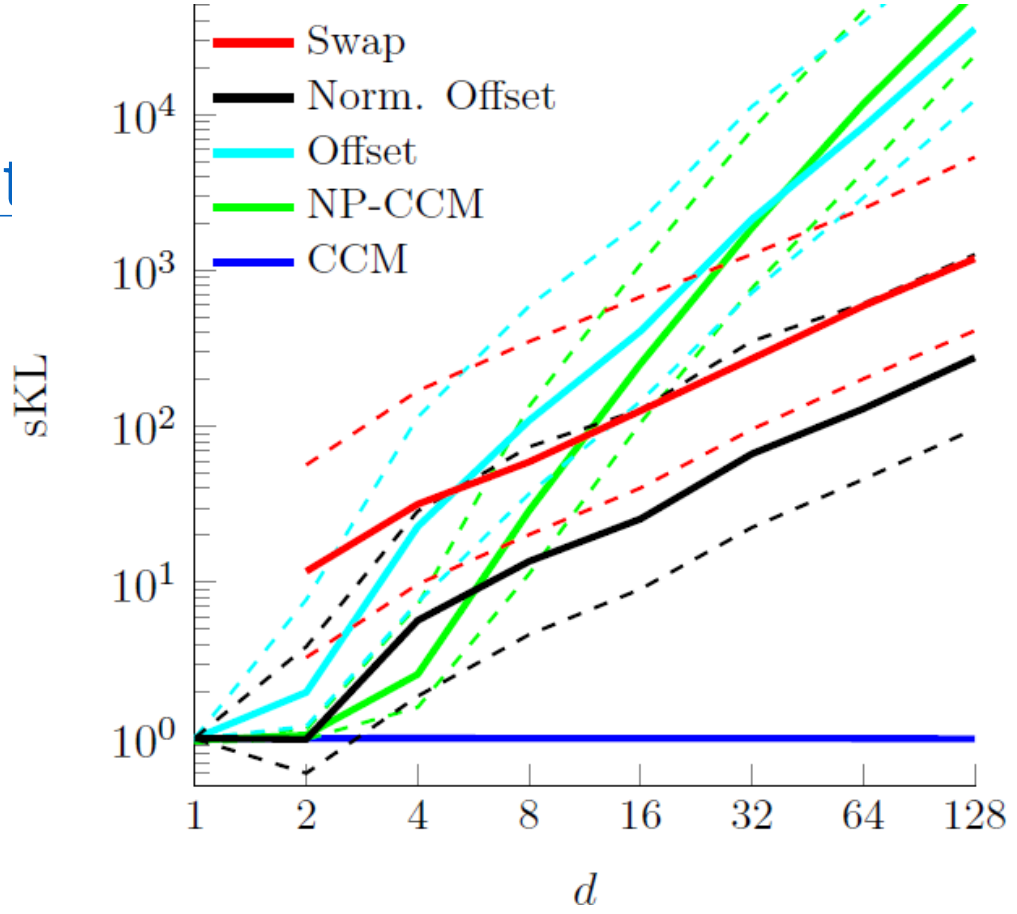
Controlling the change magnitude provides results that are better interpretable and reproducible.

D. Carrera, G. Boracchi *"Generating High-Dimensional Datastreams for Change Detection"* Big Data Research, Elsevier, 17 pages, 2017

Ccm: controlling the change magnitude

Watch out: **Most experimental practices** to manipulate datastreams for change-detection purposes, **lead to changes that steadily increase with d**

https://home.deib.polimi.it/carrerad/projects.ht



D. Carrera, G. Boracchi *"Generating High-Dimensional Datastreams for Change Detection"* Big Data Research 2017

Resources

All the preprints are available from my webpage:

http://home.deib.polimi.it/boracchi/publications.html

QuantTrees and CCM codes are instead available from github (link on my website)