CCM: Controlling the Change Magnitude in High Dimensional Data

C. Alipi, G. Boracchi, D. Carrera

October, 25th , 2016

giacomo.boracchi@polimi.it

INNS Big Data 2016

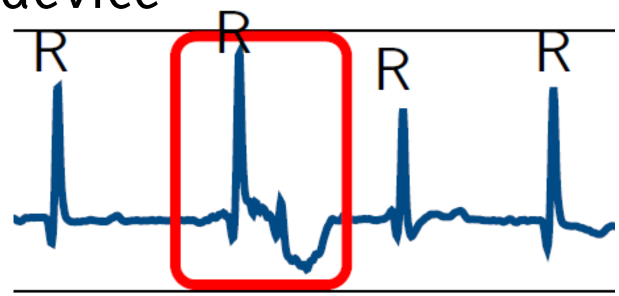
Examples of Change-Detection Applications

Stream mining: online classification systems, fraud-detection systems

Environmental/Industrial monitoring: quality inspection systems, fault-detection systems

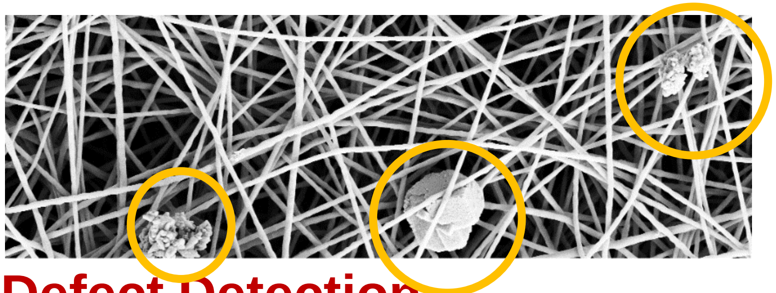
Health monitoring: arrhythmias detection, detection of mispositioning of monitoring device

Fraud Detection



ECG Monitoring

Environmental Monitoring



Defect Detection

Motivations

- The trend is to address **change-detection** problems in increasingly **high-dimensional spaces.**
- To reliable assess **algorithm performance**, a **large number of dataset** is needed
- Unfortunately, there are **not many** suitable real-world **datasets**
- In practice, researchers typically resort to:
  - **Synthetically** generating datasets (**pros:** stable performance measures, **cons:** simplistic distributions and changes)

Motivations

- The trend is to address **change-detection** problems in increasingly **high-dimensional spaces.**
- To reliable assess **algorithm performance**, a **large number of dataset** is needed
- Unfortunately, there are **not many** suitable real-world **datasets**
- In practice, researchers typically resort to:
  - **Synthetically** generating datasets (**pros:** stable performance measures, **cons:** simplistic distributions and changes)
  - **Manipulating real world** dataset (**pros:** realistic data, stable performance, **cons:** changes are arbitrarily introduced)

Our Contributions

- **CCM: Controlling change magnitude** a framework to:
  - **Manipulate real-world datasets** of **arbitrary dimension**
  - Make experiments reproducible
  - Allow to study the impact of data-dimension on change-detection performance
- The framework relies on two iterative algorithms whose convergence is **analytically proved**
- Our experiments show that **common approaches** considerably **increase** the **change magnitude** when data dimension scales

PROBLEM FORMULATION

Introduce changes in real-world datasets

Manipulating Real-World Dataset: Problem Formulation

Let $S \subset \mathbb{R}^d$ be a **dataset** of **stationary data** containing i.i.d. samples from an unknown distribution ϕ_0.

We want to **generate a datastream** $X = \{x(t), t = 1, \dots \tau, \dots\}$ **affected by a change at** $t = \tau$ such that

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & t < \tau \\ \phi_1 & t \geq \tau \end{cases}, \qquad \text{where } \phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$$

where $Q \in \mathbb{R}^{d \times d}$ is an orthogonal matrix and $v \in \mathbb{R}^d$

Manipulating Real-World Dataset: Problem Formulation

Let $S \subset \mathbb{R}^d$ be a **dataset** of **stationary data** containing i.i.d. samples from an unknown distribution ϕ_0.

We want to **generate a datastream** $X = \{x(t), t = 1, \dots \tau, \dots\}$ **affected by a change at** $t = \tau$ such that

$$\boldsymbol{x}(t) \sim \begin{cases} \phi_0 & t < \tau \\ \phi_1 & t \geq \tau \end{cases}, \qquad \text{where } \phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$$

where $Q \in \mathbb{R}^{d \times d}$ is an orthogonal matrix and $v \in \mathbb{R}^d$

In particular, $X = \{x(t), t = 1, \dots \tau, \dots\}$ is obtained as

- When $t < \tau$, $\boldsymbol{x}(t)$ is randomly selected from S
- When $t \geq \tau$, $\boldsymbol{x}(t)$ is obtained by roto-translating remaining samples in S according to ϕ_1

We reshuffle S, repeat the process, **obtain another datastream**

Manipulating Real-World Dataset: Problem Formulation

Our goal: define Q and v such that the change $\phi_0 \rightarrow \phi_1$ has a magnitude equal (or arbitrarily close) to κ

Manipulating Real-World Dataset: Problem Formulation

Our goal: define Q and v such that the change $\phi_0 \to \phi_1$ has a magnitude equal (or arbitrarily close) to κ

We define the **change magnitude** as the symmetric **Kullback-Leibler divergence**

$$\text{sKL}(\phi_0, \phi_1) = \text{KL}(\phi_0, \phi_1) + \text{KL}(\phi_1, \phi_0) =$$

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

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 goal: define Q and v such that the change $\phi_0 \rightarrow \phi_1$ has a magnitude equal (or arbitrarily close) to κ

We define the **change magnitude** as the symmetric **Kullback-Leibler divergence**

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

Our goal: define Q and v such that

$$\text{sKL}(\phi_0, \phi_1) = \text{sKL}(\phi_0, \phi_0(Q \cdot\ + v)) \approx \kappa$$

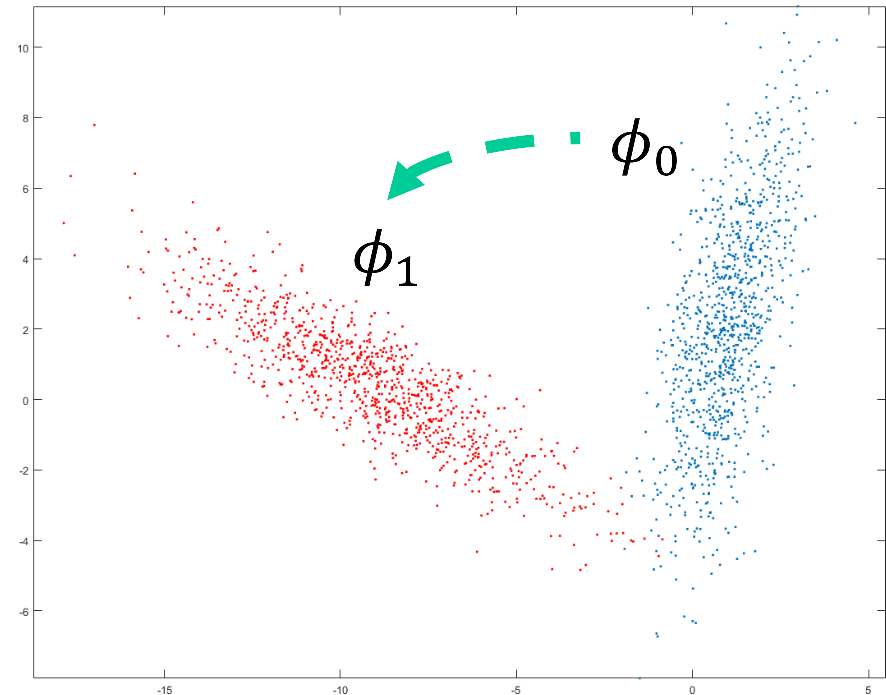
T. Dasu, K. Shankar, S. Venkatasubramanian, K. Yi, *"An information-theoretic approach to detecting changes in multi-dimensional data streams"* In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications, 2006

Manipulating Real-World Dataset: The change model

Assuming $\phi_1(\boldsymbol{x}) = \phi_0(Q\boldsymbol{x} + \boldsymbol{v})$ is quite a **general change model** which includes

- shifts in the mean
- changes in the correlation among components of $\boldsymbol{x}$

Thus, it requires a multivariate monitoring scheme!



CCM: CONTROLLING THE CHANGE MAGNITUDE

Method description

Main Components of CCM

- Fitting pre-change distribution
- Change parametrization
- Initialization
- Iteration

Fitting pre-change distribution

Since ϕ_0 is typically **unknown**, we compute an estimate $\tilde{\phi}_0$ by **fitting** a **Gaussian Mixture** on the **whole dataset** S

Parametrization

To ease our developments we parametrize Q and $\boldsymbol{v}$ as follows:

- Q is expressed w.r.t. its rotation angles $\boldsymbol{\theta} \in \mathbb{R}^{\lfloor d/2 \rfloor}$ and a coordinate system $P \in \mathbb{R}^{d \times d}$ (orthogonal matrix)

$$Q(\boldsymbol{\theta}, P) = P\, S(\boldsymbol{\theta})\, P'$$

  where $S(\boldsymbol{\theta})$ is the rotation matrix w.r.t. angles in $\boldsymbol{\theta}$

- $\boldsymbol{v}$ is expressed as

$$\boldsymbol{v}(\rho, \boldsymbol{u}) = \rho \boldsymbol{u}$$

  where $\boldsymbol{u} \in \mathbb{R}^d, ||\boldsymbol{u}|| = 1$ indicates the translation direction and $\rho > 0$ the translation magnitude

CCM – Algorithm 1

Initialization: Define Q^0 and v^0 such that

$$\mathrm{sKL}\left(\tilde{\phi}_0, \tilde{\phi}_0(Q^0 \cdot\ + \boldsymbol{v}^0)\right) \geq \kappa$$

Algorithm 1

1. **Input:** $\widetilde{\phi}_0$, target value κ of $\mathrm{sKL}(\widetilde{\phi}_0, \phi_1)$
2. **Output:** Roto-translation parameters $\boldsymbol{\theta}^{(0)}$, P, $\rho^{(0)}$, $\mathbf{u}$
3. Set $\rho^{(0)} = 1$.
4. **repeat**
5. Randomly generate m angles $\boldsymbol{\theta}^{(0)}$ in $[-\pi/2, \pi/2]^m$ and a unitary vector $\mathbf{u}$.
6. Generate a matrix $A \in \mathbb{R}^{d \times d}$ with Gaussian entries.
7. Set P as the orthogonal matrix of the QR decomposition of A.
8. Set $Q^{(0)}(\boldsymbol{\theta}^{(0)}, P) = PS(\boldsymbol{\theta}^{(0)})P'$ and $\mathbf{v}(\rho^{(0)}, \mathbf{u}) = \rho^{(0)}\mathbf{u}$.
9. Compute $s^{(0)} = \mathrm{sKL}(\widetilde{\phi}_0, \phi_1)$, where $\phi_1 = \widetilde{\phi}_0(Q^{(0)} \cdot + \mathbf{v}^{(0)})$
10. $\rho^{(0)} = 2\rho^{(0)}$.
11. **until** $s^{(0)} > \kappa$;

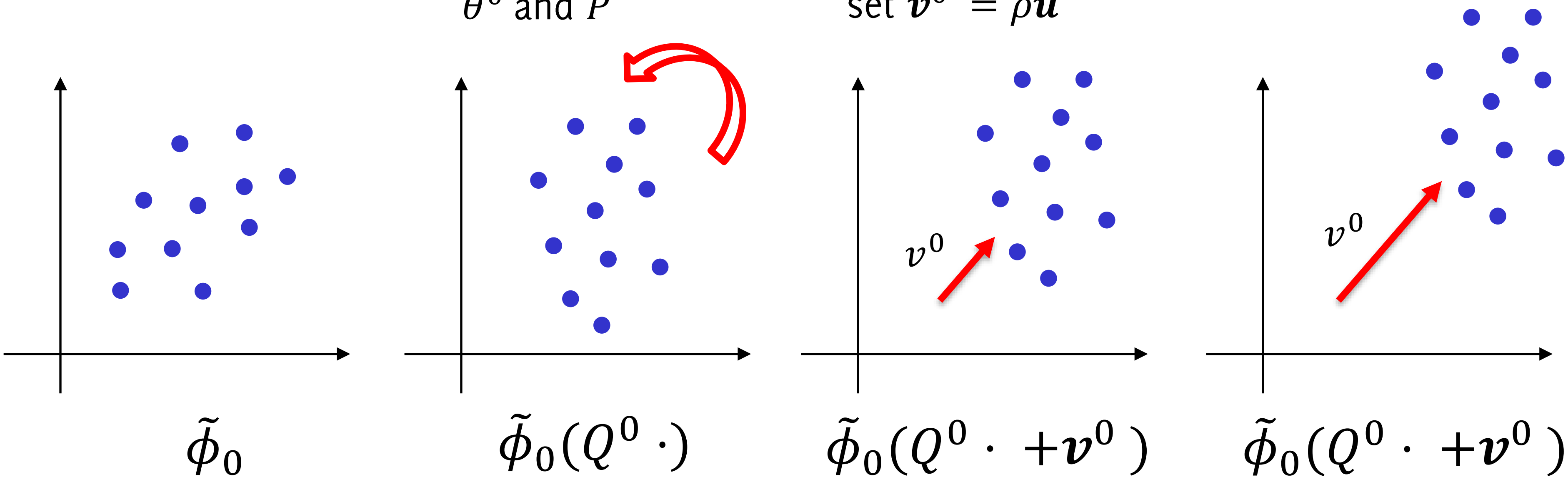
Algorithm1: Define Q^0 and v^0 such that

$$\mathrm{sKL}\left(\tilde{\phi}_0, \tilde{\phi}_0(Q^0\cdot\ + \boldsymbol{v}^0)\right) \geq \kappa$$

1) Randomly choose θ^0 and P

2) Randomly choose $\boldsymbol{u}$, set $\boldsymbol{v}^0 = \rho \boldsymbol{u}$

3) Increase ρ



CCM – Theorem 1

Theorem 1. *Let $\widetilde{\phi}_0$ be a Gaussian mixture. Then, for any $\kappa > 0$, Algorithm 1 converges in a finite number of iterations.*

To prove Theorem 1 it is enough to show that

$$\mathrm{sKL}\left(\tilde{\phi}_0, \tilde{\phi}_0(Q\cdot\ +\boldsymbol{v})\right) \rightarrow \infty$$

for any Q when $||\boldsymbol{v}|| \rightarrow \infty$ or that one it admits a diverging lower bounds

(see the paper for details…)

Iteratively adjust Q and $\boldsymbol{v}$ towards

$$\mathrm{sKL}\left(\tilde{\phi}_0, \tilde{\phi}_0(Q\cdot\ +\boldsymbol{v})\right) \to \kappa$$

Algorithm 2

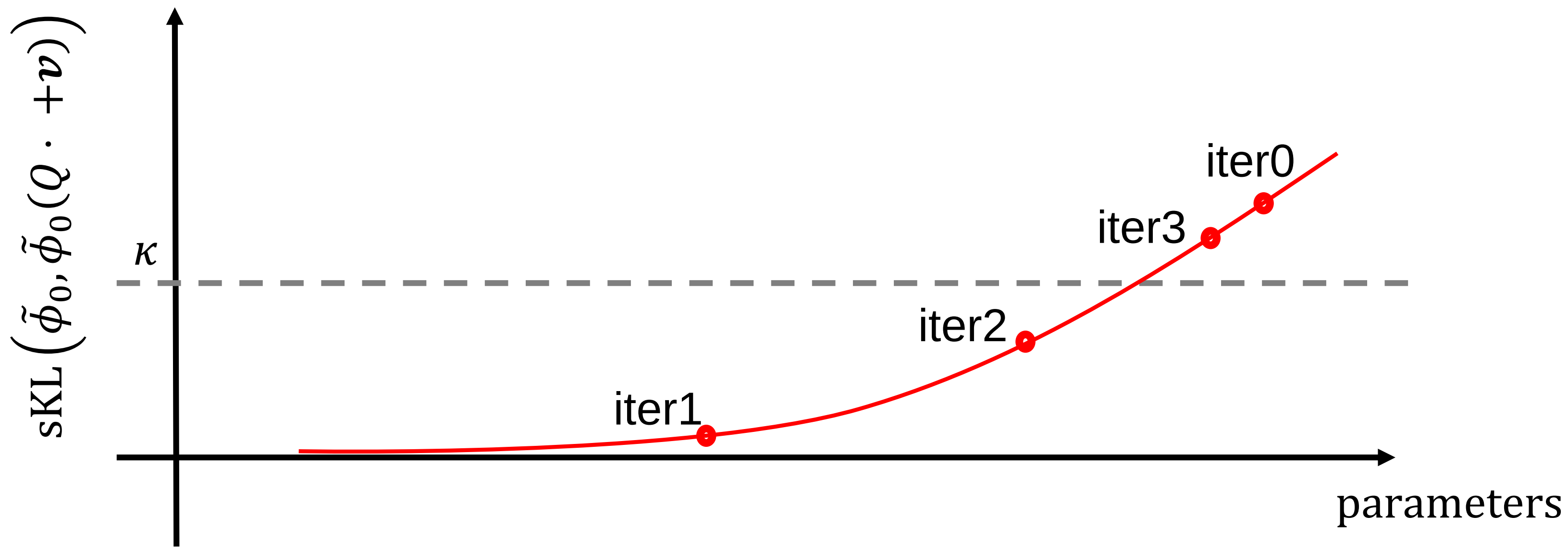
1. **Input:** $\boldsymbol{\theta}^{(0)}$, P ,$\rho^{(0)}$, $\mathbf{u}$ from Algorithm 1, $\widetilde{\phi}_0$, κ and tolerance ε
2. **Output:** Q and $\mathbf{v}$ defining the roto-translation yielding desired change magnitude
3. Set the lower bounds parameters $\boldsymbol{\theta}_l^{(1)} = 0$, $\rho_l^{(1)} = 0$.
4. Set the upper bounds parameters $\boldsymbol{\theta}_u^{(1)} = \boldsymbol{\theta}^{(0)}$, $\rho_u^{(1)} = \rho^{(0)}$.
5. Set $j = 1$.
6. **repeat**
7. Compute $\boldsymbol{\theta}^{(j)} = (\boldsymbol{\theta}_l^{(j)} + \boldsymbol{\theta}_u^{(j)})/2$, and $Q^{(j)}(\boldsymbol{\theta}^{(j)}, P)$ as in (6).
8. Compute $\rho^{(j)} = (\rho_l^{(j)} + \rho_u^{(j)})/2$, and $\mathbf{v}^{(j)}(\rho^{(j)}, \mathbf{u})$ as in (7).
9. Compute $s^{(j)} = \mathrm{sKL}(\widetilde{\phi}_0, \phi_1^{(j)})$, where $\phi_1^{(j)}(\cdot) = \widetilde{\phi}_0(Q^{(j)}\cdot + \mathbf{v}^{(j)})$.
10. **if** $s^{(j)} < \kappa$ **then**
11. $\boldsymbol{\theta}_l^{(j+1)} = \boldsymbol{\theta}^{(j)}$, $\rho_l^{(j+1)} = \rho^{(j)}$.
12. **else**
13. $\boldsymbol{\theta}_u^{(j+1)} = \boldsymbol{\theta}^{(j)}$, $\rho_u^{(j+1)} = \rho^{(j)}$.
14. **end**
15. $j = j + 1$;
16. **until** $\left|s^{(j)} - \kappa\right| < \varepsilon$;
17. Set $Q = Q^{(j)}$, $\mathbf{v} = \mathbf{v}^{(j)}$.

CCM – Algorithm 2

Algorithm2: Implements a bisection method to compute $\boldsymbol{\theta}$ and ρ yielding the desired sKL value.

Bisection is performed w.r.t. both $\boldsymbol{\theta}$ and ρ, and we stop when

$$\left|\text{sKL}\left(\tilde{\phi}_0, \tilde{\phi}_0(Q\cdot\ +\boldsymbol{v})\right) - \kappa\right| < \epsilon$$



CCM – Algorithm 2

Theorem 2. *Let* $\widetilde{\phi}_0$ *be a Gaussian mixture. Then, for any* $\kappa > 0$*, Algorithm 2 converges in a finite number of iterations.*

To prove Theorem 2 it is enough to show that the function used in the bisection is continuous (see the paper for details…)

EXPERIMENTS

Why controlling the change-magnitude is important

Experiments

Goals:

- Show the limitations of commonly used approaches that are primarily heuristic
- Demonstrate the effectiveness of CCM

Dataset: of MiniBooNE Particle Dataset from the UCI repository

- $d = 50$, components have been standardized
- 93108 samples (only one class)
- We fit a GMM having 2 degrees of freedom
- We generate multiple datasets

Figures of merit:

- The magnitude of the introduced change
- The change-detection performance (power of HT)

Considered Approaches

Methods to manipulate the dataset:

- **CCM**: configured to yield $\mathrm{sKL}(\tilde{\phi}_0, \tilde{\phi}_1) = 1, \; \forall d$
- **offset**: add an offset $v = 1$ to each component of the standardized data. This corresponds to $\phi_1 = \phi_0(\boldsymbol{x} + \mathbf{1}_d)$
- **Swap**: two components, randomly chosen, are swapped. This change model has Q equal to the corresponding permutation matrix and $\boldsymbol{v} = \mathbf{0}$

All these approaches are tested by introducing changes in datasets having different dimensions

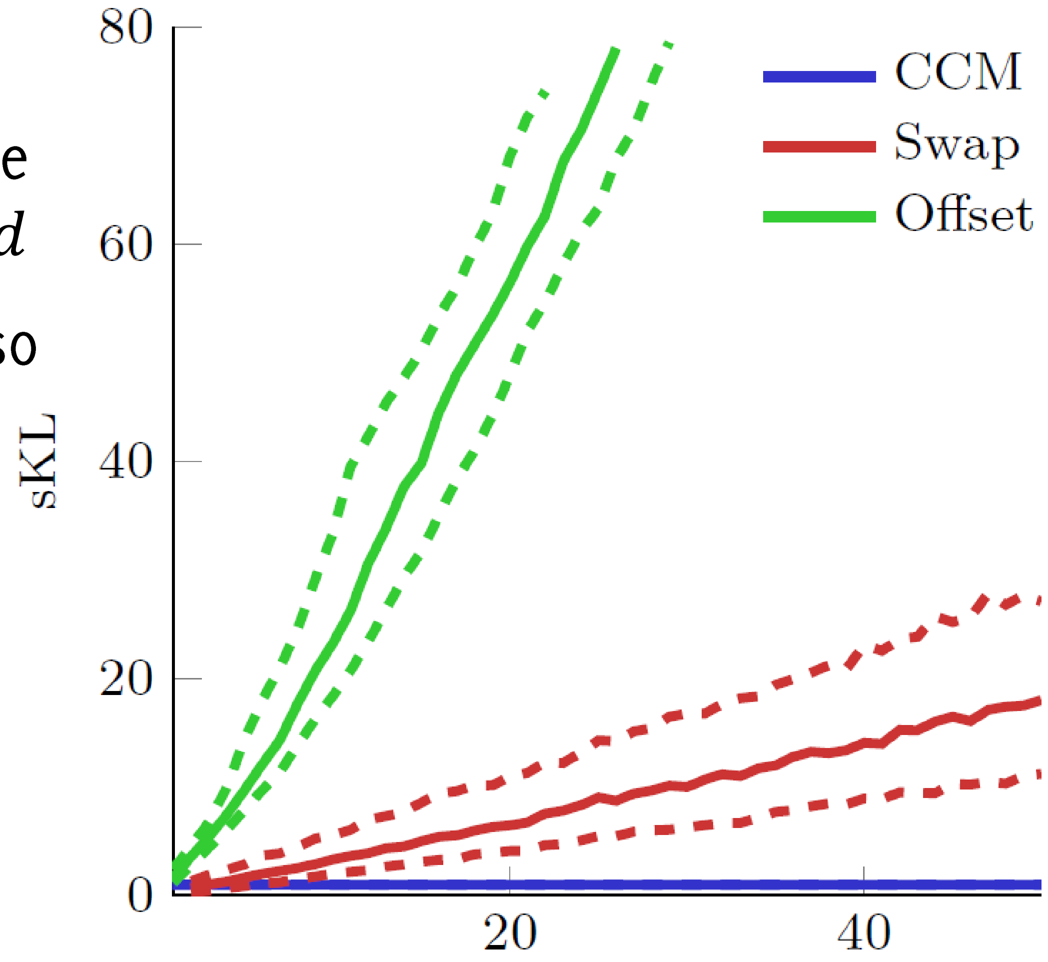
L. I. Kuncheva, *"Change detection in streaming multivariate data using likelihood detectors,"* IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 5, 2013.

A. Zimek, E. Schubert, H.P. Kriegel. *"A survey on unsupervised outlier detection in high-dimensional numerical data"* Statistical Analysis and Data Mining: The ASA Data Science Journal, 5(5), 2012.

Experiments on the Change Magnitude

Distribution of $\mathrm{sKL}(\tilde{\phi}_0, \tilde{\phi}_1)$ computed from manipulated datasets

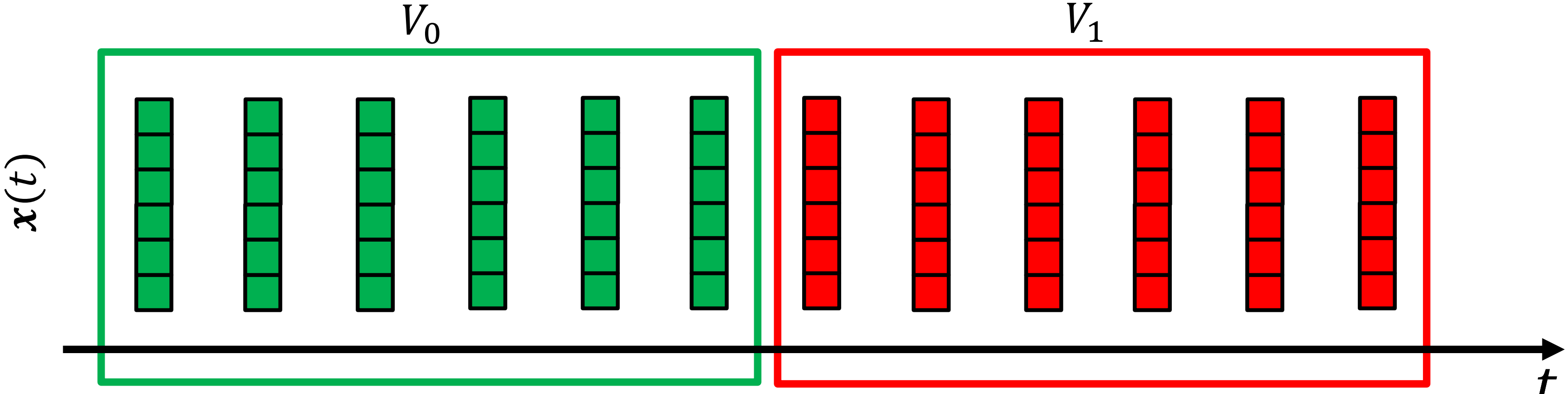
- Only CCM preserves the change magnitude
- Swap and Offset introduce changes increasing with d
- The dispersion of sKL also increases with d



Empirical Analysis on the Change-Detectability

The change-detectabiltity measure:

- **Test data: two windows** V_0 and V_1 (200 samples each) selected before and after the change.



Empirical Analysis on the Change-Detectability

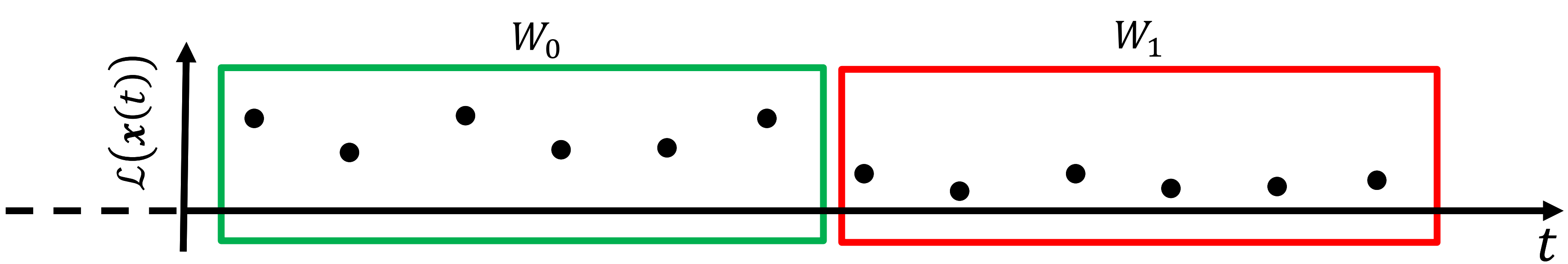
The change-detectabiltity measure:

- **Test data: two windows** V_0 and V_1 (200 samples each) selected before and after the change.
- Compute $\log\left(\hat{\phi}_0(\boldsymbol{x})\right)$ from V_0 and V_1, obtaining W_0 and W_1
- Compute the Lepage statistic $\mathcal{T}(W_0, W_1)$ to compare them
- Detect a change by an hypothesis test

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

where h controls the amount of false positives

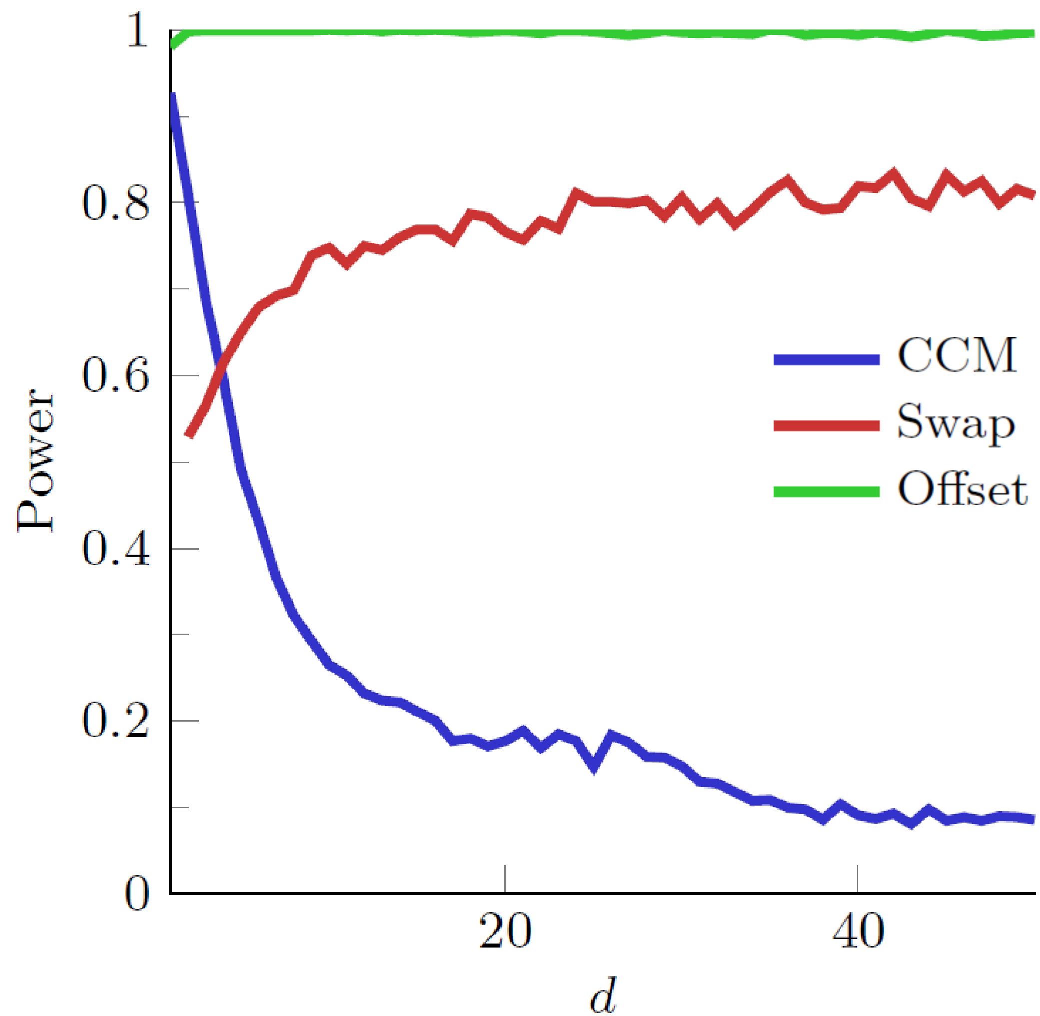
- Use the **power** of this **test** to assess change detectability



Results

The power of HT indicates that:

- Changes introduced by CCM becomes more difficult to detect when d increases. This is coherent with our theoretical analysis shown in [IJCAI2016]
- Changes introduced by other methods are more prominent and easier to detect when d grows.



C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016, New York, USA, July 9 - 13

Conclusions

CCM is a rigorous framework to introduce changes having a controlled magnitude in multivariate datasets

The convergence of its algorithms have been proved

CCM is implemented in Matlab and is freely available for download at

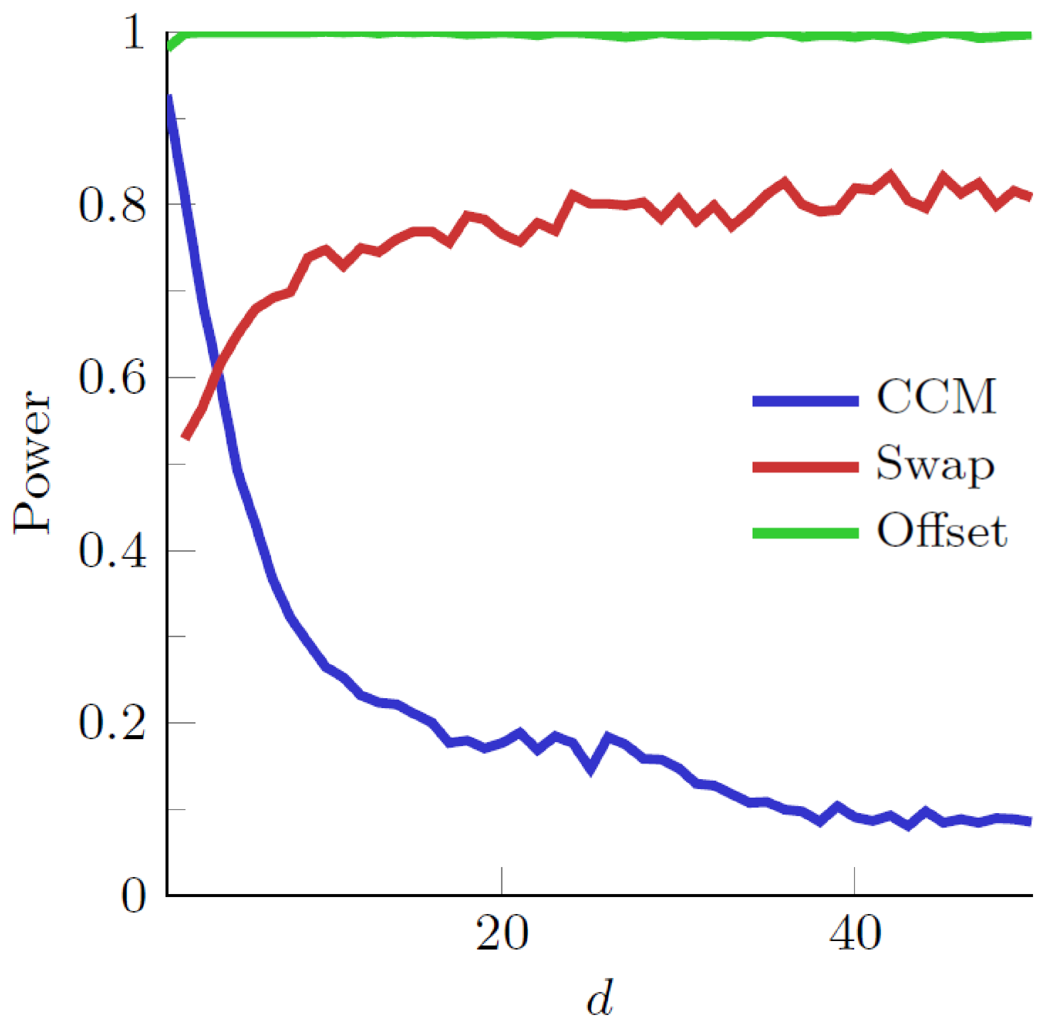
https://home.deib.polimi.it/carrerad/projects.html

Our experiments remark the importance of controlling the change magnitude when manipulating real-world datasets

- to fairly assess detection performance when d increases
- to make experiments more easily reproducible

Ongoing work concerns extending the framework to more general change models

Thanks, Questions?



C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "*Change Detection in Multivariate Datastreams: Likelihood and Detectability Loss*" IJCAI 2016, New York, USA, July 9 - 13