CCM: Controlling the Change Magnitude in High Dimensional Data

C. Alippi, 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
Motivations

- The trend is to address **change-detection** problems in increasingly **high-dimensional spaces**.
- To reliably 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 reliably 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** is 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
Let $S \subset \mathbb{R}^d$ be a dataset of stationary data containing i.i.d. samples from an unknown distribution $\phi_0$.

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

$$x(t) \sim \begin{cases} 
\phi_0 & t < \tau \\
\phi_1 & t \geq \tau 
\end{cases},$$

where $\phi_1(x) = \phi_0(Qx + \nu)$

where $Q \in \mathbb{R}^{d \times d}$ is an orthogonal matrix and $\nu \in \mathbb{R}^d$
Let $S \subset \mathbb{R}^d$ be a dataset of stationary data containing i.i.d. samples from an unknown distribution $\phi_0$.

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

$$x(t) \sim \begin{cases} 
\phi_0 & t < \tau \\
\phi_1 & t \geq \tau \end{cases},$$

where $\phi_1(x) = \phi_0(Qx + \nu)$

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

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

- When $t < \tau$, $x(t)$ is randomly selected from $S$
- When $t \geq \tau$, $x(t)$ is obtained by roto-translating remaining samples in $S$ according to $\phi_1$

We reshuffle $S$, repeat the process, obtain another datastream
Our goal: define $Q$ and $\nu$ such that the change $\phi_0 \to \phi_1$ has a magnitude equal (or arbitrarily close) to $\kappa$
Our goal: define $Q$ and $v$ such that the change $\phi_0 \rightarrow \phi_1$ has a magnitude equal (or arbitrarily close) to $\kappa$

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

$$sKL(\phi_0, \phi_1) = KL(\phi_0, \phi_1) + KL(\phi_1, \phi_0) =$$

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

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

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

$$sKL(\phi_0, \phi_1) = KL(\phi_0, \phi_1) + KL(\phi_1, \phi_0) =$$

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

Our goal: define $Q$ and $\nu$ such that

$$sKL(\phi_0, \phi_1) = sKL(\phi_0, \phi_0(Q \cdot + \nu)) \approx \kappa$$

---

Assuming $\phi_1(x) = \phi_0(Qx + v)$ is quite a general change model which includes

- shifts in the mean
- changes in the correlation among components of $x$

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 $\phi_0$ 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 $\nu$ as follows:

- $Q$ is expressed w.r.t. its rotation angles $\theta \in \mathbb{R}^{[d/2]}$ and a coordinate system $P \in \mathbb{R}^{d \times d}$ (orthogonal matrix)
  \[ Q(\theta, P) = P \, S(\theta) \, P' \]
  where $S(\theta)$ is the rotation matrix w.r.t. angles in $\theta$

- $\nu$ is expressed as
  \[ \nu(\rho, u) = \rho u \]
  where $u \in \mathbb{R}^d, \|u\| = 1$ indicates the translation direction and $\rho > 0$ the translation magnitude
Initialization: Define $Q^0$ and $\nu^0$ such that
\[
\text{sKL} \left( \tilde{\phi}_0, \tilde{\phi}_0(Q^0 \cdot + \nu^0) \right) \geq \kappa
\]

**Algorithm 1**

1. **Input:** $\tilde{\phi}_0$, target value $\kappa$ of sKL($\tilde{\phi}_0, \phi_1$)
2. **Output:** Roto-translation parameters $\theta^{(0)}$, $P$, $\rho^{(0)}$, $u$
3. Set $\rho^{(0)} = 1$.
4. **repeat**
   5. Randomly generate $m$ angles $\theta^{(0)}$ in $[-\pi/2, \pi/2]^m$ and a unitary vector $u$.
   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 $Q^{(0)}(\theta^{(0)}, P) = PS(\theta^{(0)})P' \text{ and } \nu(\rho^{(0)}, u) = \rho^{(0)} u$.
   9. Compute $s^{(0)} = \text{sKL}(\tilde{\phi}_0, \phi_1)$, where $\phi_1 = \tilde{\phi}_0(Q^{(0)} \cdot + \nu^{(0)})$
   10. $\rho^{(0)} = 2\rho^{(0)}$.
5. **until** $s^{(0)} > \kappa$;
**Algorithm 1:** Define $Q^0$ and $v^0$ such that

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

1) Randomly choose $\theta^0$ and $P$
2) Randomly choose $u$, set $v^0 = \rho u$
3) Increase $\rho$
Theorem 1. Let $\tilde{\phi}_0$ 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

$$\text{sKL} \left( \tilde{\phi}_0, \tilde{\phi}_0(Q \cdot + \nu) \right) \to \infty$$

for any $Q$ when $||\nu|| \to \infty$ or that one it admits a diverging lower bounds

(see the paper for details...)
Iteratively adjust $Q$ and $\nu$ towards

$$s\text{KL} \left( \tilde{\phi}_0, \tilde{\phi}_0 (Q \cdot +\nu) \right) \rightarrow \kappa$$

**Algorithm 2**

1. **Input:** $\theta^{(0)}$, $P$, $\rho^{(0)}$, $u$ from Algorithm 1, $\tilde{\phi}_0$, $\kappa$ and tolerance $\varepsilon$
2. **Output:** $Q$ and $\nu$ defining the roto-translation yielding desired change magnitude
3. Set the lower bounds parameters $\theta_l^{(1)} = 0$, $\rho_l^{(1)} = 0$.
4. Set the upper bounds parameters $\theta_u^{(1)} = \theta^{(0)}$, $\rho_u^{(1)} = \rho^{(0)}$.
5. Set $j = 1$.
6. repeat
7. Compute $\theta^{(j)} = (\theta_l^{(j)} + \theta_u^{(j)})/2$, and $Q^{(j)}(\theta^{(j)}, P)$ as in (6).
8. Compute $\rho^{(j)} = (\rho_l^{(j)} + \rho_u^{(j)})/2$, and $\nu^{(j)}(\rho^{(j)}, u)$ as in (7).
9. Compute $s^{(j)} = s\text{KL}(\tilde{\phi}_0, \tilde{\phi}_1^{(j)})$, where $\phi_1^{(j)}(\cdot) = \tilde{\phi}_0 (Q^{(j)} \cdot +\nu^{(j)})$.
10. if $s^{(j)} < \kappa$ then
    11.    $\theta_l^{(j+1)} = \theta^{(j)}$, $\rho_l^{(j+1)} = \rho^{(j)}$.
    12. else
    13.    $\theta_u^{(j+1)} = \theta^{(j)}$, $\rho_u^{(j+1)} = \rho^{(j)}$.
    14. end
15. $j = j + 1$;
16. until $|s^{(j)} - \kappa| < \varepsilon$;
17. Set $Q = Q^{(j)}$, $\nu = \nu^{(j)}$. 
Algorithm 2: Implements a bisection method to compute $\theta$ and $\rho$ yielding the desired sKL value.

Bisection is performed w.r.t. both $\theta$ and $\rho$, and we stop when

$$\left| sKL\left(\tilde{\phi}_0, \tilde{\phi}_0(Q \cdot +v)\right) - \kappa \right| < \epsilon$$
Theorem 2. Let $\tilde{\phi}_0$ 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 $\text{sKL}(\tilde{\phi}_0, \tilde{\phi}_1) = 1$, $\forall d$

- **offset**: add an offset $\nu = 1$ to each component of the standardized data. This corresponds to $\phi_1 = \phi_0(x + 1_d)$

- **Swap**: two components, randomly chosen, are swapped. This change model has $Q$ equal to the corresponding permutation matrix and $\nu = 0$

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


Experiments on the Change Magnitude

Distribution of $s\text{KL}({\tilde{\phi}}_0, {\tilde{\phi}}_1)$ computed from manipulated datasets

- Only CCM preserves the change magnitude
- Swap and Offset introduce changes increasing with $d$
- The dispersion of $s\text{KL}$ also increases with $d$
The change-detectability measure:

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

- **Test data:** two windows $V_0$ and $V_1$ (200 samples each) selected before and after the change.
- Compute $\log(\hat{\phi}_0(x))$ 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
  \[ \mathcal{T}(W_0, W_1) \leq h \]
  where $h$ controls the amount of false positives.
- Use the **power** of this test to assess change detectability.
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.
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.
C. Alippi, G. Boracchi, D. Carrera, M. Roveri, "Change Detection in Multivariate Data Streams: Likelihood and Detectability Loss" IJCAI 2016, New York, USA, July 9 - 13