Functional Isolation Forest

Guillaume Staerman*

Joint work with

Pavlo Mozharovskyi*, Stéphan Clémençon* and Florence D'Alché-Buc*

MLITRW, October 02, 2019

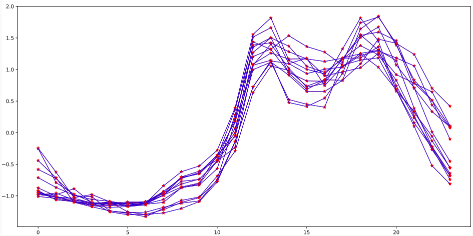
*LTCI, Telecom Paris, Institut Polytechnique de Paris.

Functional Data Framework

- Let $X = \{X(t) \in \mathbb{R}^d,\ t \in [0,1]\}$ be a random variable that takes its values in a (multivariate) functional space.

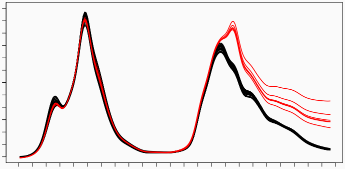
- In practice, we only have access to the realization of X at a finite number of arguments/times, $x = \{x(t_1), \ldots, x(t_p)\}$ such that $0 \leq t_1 < \cdots < t_p \leq 1$.

- The first step: reconstruct functional object from partial observations (time-series) with interpolation or basis decomposition.

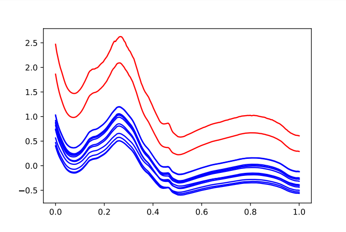


Anomaly detection and functional data

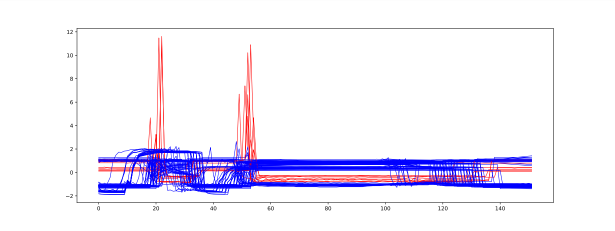
Shape anomalies



Shift anomalies



Isolated anomalies



Functional Isolation Forest

- This ensemble learning algorithm builds a collection of *functional isolation trees*.
- *Functional isolation tree* : binary tree based on a recursive and randomized tree-structured partitioning procedure.
- **General principle**:
  1. Select a function $\mathbf{d}$ into a dictionary $\mathcal{D}$.
  2. Compute the dot products $\langle \cdot, \cdot \rangle$ between $\mathbf{d}$ and the data.
  3. Draw randomly a treshold κ on the real line.
  4. Split the space by a perpendicular hyperplan along $\mathbf{d}$ going through κ.
  5. Repeat this procedure until every data are isolated!!!
- The trick : an anomaly should be isolated faster than normal data.

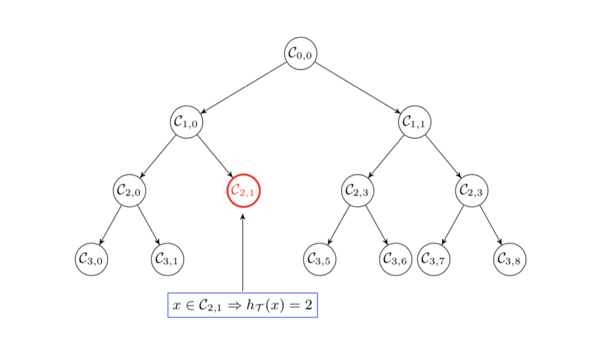
Anomaly score prediction

- One may then define the piecewise constant function $h_\tau : \mathcal{X} \to \mathbb{N}$ by: $\forall x \in \mathcal{X}$,

  $h_\tau(x) = j$ if and only if $x \in \mathcal{C}_{j,k}$ and $\mathcal{C}_{j,k}$ is associated to a terminal node.

- Considering a collection of F-*i*tree $\mathcal{T}_1, \ldots, \mathcal{T}_N$, the scoring function is defined by

$$s_n(x) = 2^{-\frac{1}{Nc(n)}\sum_{l=1}^{N} h_{\tau_l}(x)},$$



Thank you !

All codes are available at `https://github.com/Gstaerman/FIF`