Signature Isolation Forest

Marta Campi[1], **Guillaume Staerman**[2], **Gareth W. Peters**[3], **Tomoko Matsui**[4]

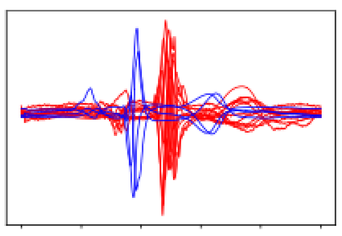
[1]Institut Pasteur, Institut de l'Audition, IHU reConnect, Paris, France
[2]INRIA, CEA, Univ. Paris-Saclay, France
[3]Department of Statistics & Applied Probability, University of California Santa Barbara, USA
[4]Institute of Statistical Mathematics, Tokyo, Japan

1. Context

Functional Anomaly Detection Setting:

- Multivariate functional data $\mathbf{X} \in \mathcal{F}^d$ such that $X(t) \in \mathbb{R}^d, \forall t \in [0,1]$.
- Data observed as $\mathcal{X} = \{x_i(t_1), \dots, x_i(t_p)\}_{i=1}^n$ on a finite discretization.

Functional Isolation Forest[1] (FIF) has limitations: its inner product and dictionary choices significantly impact performance. We introduce Signature Isolation Forest leveraging rough path theory[2].



2. Contributions

Two new Functional Anomaly Detection (FAD) algorithms based on isolation forest structure and the signature approach:

- **Kernel Signature Isolation Forest (K-SIF)**: Extension of Functional Isolation Forest (FIF) based on the kernel signature.
- **Signature Isolation Forest (SIF)**: Isolation Forest based algorithm relying on the coordinate signature.

3. Signature Method

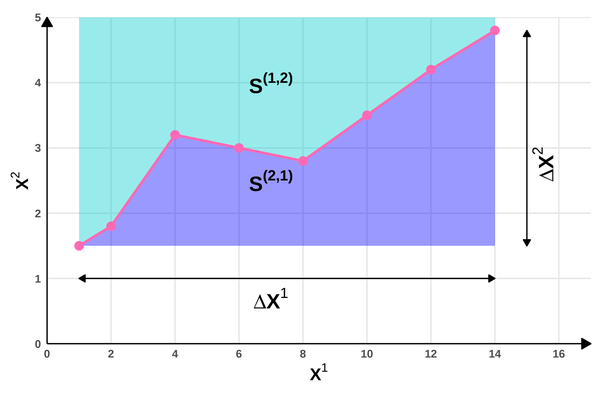
Let $\mathbf{X} \in \mathcal{F}^d$ be a r.v. of bounded variation. For any set of coordinates $\{i_1, \dots, i_k\} \subset \{1, \dots, d\}^k, k \in \mathbb{N}_*$, and $[s,t] \subset [0,1]$, the associated **coordinate signature**[4] is defined by:

$$S_{(i_1,\dots,i_k)}(\mathbf{X})_{[s,t]} = \int_{s \le u_1 < \dots < u_k \le t} \cdots \int \mathrm{d}X_{u_1}^{i_1} \dots \mathrm{d}X_{u_k}^{i_k} \tag{1}$$

Given an order of truncation $k \in \mathbb{N}_*$, the **truncated signature** is the vector of finite length:

$$S^k(\mathbf{X}) = \big(1, S_1(\mathbf{X}), \dots, S_d(\mathbf{X}), S_{(1,1)}(\mathbf{X}), \dots, S_{(d,\dots,d)}(\mathbf{X})\big) \in \mathbb{R}^C \tag{2}$$

where $\mathrm{C} = \big|\{1, \dots, d\}^{\le k}\big|$



Geometric visualization of depth-2 signature terms, where $S^{(1,2)}$ (cyan region) and $S^{(2,1)}$ (purple region) represent areas corresponding to coordinate signatures.

References: **1**. Staerman, G., et al. (2019). Functional isolation forest. In Asian Conference on Machine Learning, pages 332–347. PMLR. **2**. Lyons, T., et al. (2007). Differential equations driven by rough paths. Springer. **3**. Kiraly, F. J., and Oberhauser, H. (2019). Kernels for sequentially ordered data. Journal of Machine Learning Research, 20. **4**. Fermanian, A. (2021). Embedding and learning with signatures. Computational Statistics and Data Analysis, 157:107148.

4. Kernel Truncated Signature

The signature can be seen as a feature map[3] that embeds a function or a path into the tensor algebra. It is defined as the map $K^k : \mathcal{F}^d \times \mathcal{F}^d \to \mathbb{R}$ such that

$$K^k(\mathbf{X}, \mathbf{Y}) = \langle S^k(\mathbf{X}), S^k(\mathbf{Y}) \rangle.$$

It captures non-linear relationships by embedding paths in higher-dimensional space, preserving sequential information and allowing comparison through geometric properties. This formulation effectively detects complex patterns in functional data.

5. Kernel-SIF

Input: Subsample $\{\mathbf{x}_{i_j}\}_{j=1}^m$, dictionary $\mathcal{D}$, measure $\boldsymbol{\nu}$, signature level k, split windows ω.
(a) Root node $(0,0)$ corresponds to $\mathcal{C}_{0,0} = \mathcal{F}^d$.
(b) If node (p,q) is terminal, stop; otherwise go to **(c)**.
(c) Split node (p,q) as follows:

1. Choose $\mathbf{d}$ from $\mathcal{D}$ according to $\boldsymbol{\nu}$.
2. Select γ uniformly from

$$\Big[\min_{\mathbf{x} \in \mathcal{X}_{p,q}} \langle S^k(\mathbf{x}), S^k(\mathbf{d}) \rangle, \max_{\mathbf{x} \in \mathcal{X}_{p,q}} \langle S^k(\mathbf{x}), S^k(\mathbf{d}) \rangle\Big]$$

3. Form children subsets and datasets:

$$\begin{aligned} \mathcal{C}_{p+1,2q} &= \mathcal{C}_{p,q} \cap C^L_{\text{K-SIF}} \\ \mathcal{C}_{p+1,2q+1} &= \mathcal{C}_{p,q} \cap C^R_{\text{K-SIF}} \\ \mathcal{X}_{p+1,2q} &= \mathcal{X}_{p,q} \cap \mathcal{C}_{p+1,2q} \\ \mathcal{X}_{p+1,2q+1} &= \mathcal{X}_{p,q} \cap \mathcal{C}_{p+1,2q+1} \end{aligned}$$

(d) Apply steps **(b)-(c)** to nodes $(p+1, 2q)$ and $(p+1, 2q+1)$
Output: Partition $(\mathcal{C}_{0,0}, \mathcal{C}_{1,1}, \dots)$

6. Signature Isolation Forest

Input: Subsample $\{\mathbf{x}_{i_j}\}_{j=1}^m$, signature level k, split windows ω.
(a) Root node $(0,0)$ corresponds to $\mathcal{C}_{0,0} = \mathcal{F}^d$.
(b) If node (p,q) is terminal, stop; otherwise go to **(c)**.
(c) Split node (p,q) as follows:

1. Choose coordinate $(i_1, \dots, i_\ell)$ uniformly from $\{(i_1, \dots, i_\ell) \in [\![1, d]\!]^\ell; \quad 1 \le \ell \le k\}$.
2. Select γ uniformly from

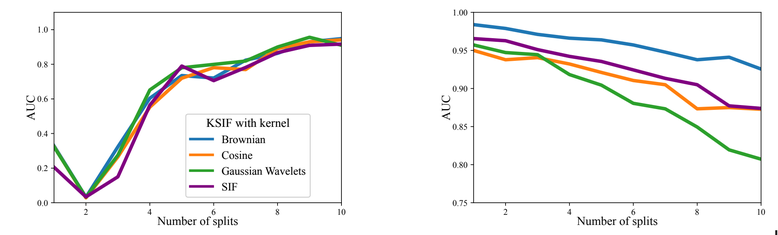
$$\Big[\min_{\mathbf{x} \in \mathcal{X}_{p,q}} S_{(i_1,\dots,i_\ell)}(\mathbf{x}), \max_{\mathbf{x} \in \mathcal{X}_{p,q}} S_{(i_1,\dots,i_\ell)}(\mathbf{x})\Big]$$

3. Form children subsets and datasets:

$$\begin{aligned} \mathcal{C}_{p+1,2q} &= \mathcal{C}_{p,q} \cap C^L_{\text{SIF}} \\ \mathcal{C}_{p+1,2q+1} &= \mathcal{C}_{p,q} \cap C^R_{\text{SIF}} \\ \mathcal{X}_{p+1,2q} &= \mathcal{X}_{p,q} \cap \mathcal{C}_{p+1,2q} \\ \mathcal{X}_{p+1,2q+1} &= \mathcal{X}_{p,q} \cap \mathcal{C}_{p+1,2q+1} \end{aligned}$$

(d) Apply steps **(b)-(c)** to nodes $(p+1, 2q)$ and $(p+1, 2q+1)$
Output: Partition $(\mathcal{C}_{0,0}, \mathcal{C}_{1,1}, \dots)$

7. Parameter Analysis



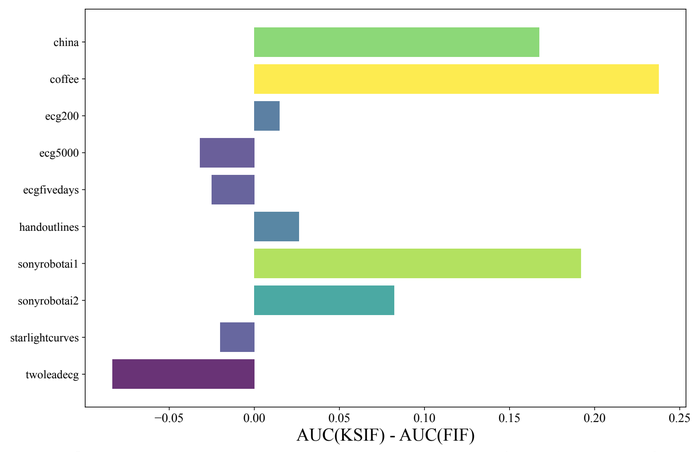
AUC vs. number of split windows: isolated anomalies (left) and persistent anomalies (right)

Key Findings:

- **Split Windows (ω)**:
  - Critical for isolated anomalies - performance improves with more windows
  - Less important for persistent anomalies - stable across window counts
  - Optimal setting $\omega = 10$ balances performance and computational cost
- **Truncation Level (k)**:
  - $k = 1$: Captures only displacements (often insufficient)
  - $k = 2$: Includes path area information (good balance)
  - $k > 2$: Diminishing returns relative to computational cost

8. Real-World Performance

Dataset	SIF	K-SIF$_C$	K-SIF$_B$	FIF$_B$
Chinatown	**1.00**	0.99	**1.00**	0.83
SonyRobotAI1	**0.99**	0.95	0.95	0.76
SonyRobotAI2	**0.93**	0.92	**0.93**	0.84
ECGFiveDays	0.93	0.92	0.90	0.93
ECG5000	0.90	**0.97**	0.91	0.88



Performance gains: K-SIF$_B$ vs FIF$_B$ (positive values = K-SIF better)

Computational Cost:

Method	SIF/K-SIF	AutoE	AnoGan
Handoutlines	30-50s	593s	681s
StarLightC.	13-20s	202s	500s

Signature methods are 10-30× faster than deep learning

9. Conclusion

- **Novel Methods**: Two functional anomaly detection algorithms using signatures
- **Key Advantages**: K-SIF offers non-linear projections; SIF provides dictionary-free detection; both excel with sequential patterns
- **Results**: SIF achieves top performance on 50% of datasets; both methods show 5-10× speed improvement over deep learning
- **Impact**: Robust solution for functional anomaly detection with temporal ordering and non-linear patterns