Karlsruhe Institute of Technology

Multi-kernel Time Series Outlier Detection

Discovery Science 2023

Florian Kalinke, Edouard Fouché, Haiko Thiessen and Klemens Böhm | Oct. 11th, 2023

Motivation

- Time-series data is ubiquitous across various domains.
- **Goal:** Detect outlying time series in a pool of time series.
- Outlier detection is best formulated as an unsupervised task.
  - No hyperparameter tuning.
  - Feature selection problem.
- Feature selection problem especially prevalent in Active Learning, they often use support vector data description (SVDD; Tax and Duin [5]).

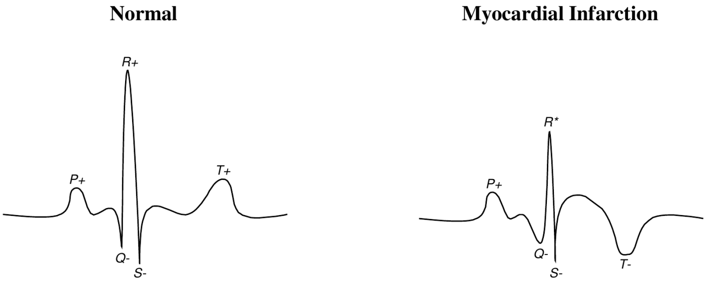

Figure: Anomalous heartbeat identified as outlying time-series [2].

Proposed Approach

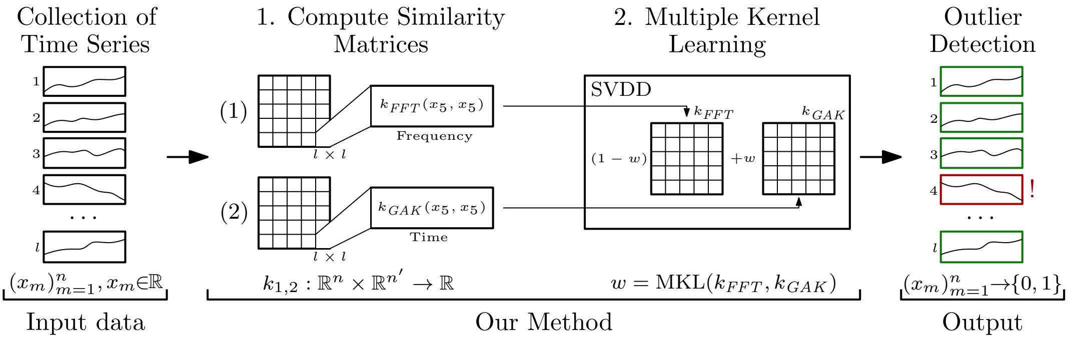

Figure: Schematic representation of the proposed outlier detection method.

Kernel function $\Longleftrightarrow$ Inner product

Karlsruhe Institute of Technology

Definition (Kernel, feature map, feature space)

Definition: Let $\mathcal{X}$ be a non-empty set. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a **kernel** on $\mathcal{X}$ if there exists a Hilbert space $\mathcal{H}$ and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that for all $x_1, x_2 \in \mathcal{X}$ it holds that

$$k(x_1, x_2) = \langle \phi(x_2), \phi(x_1) \rangle .$$

The function ϕ is called **feature map** and $\mathcal{H}$ is the associated **feature space**.

Lemma (Additivity; Steinwart and Christmann [4])

Let $\mathcal{X}$ be a set, $\alpha \geq 0$, and k, k_1, and k_2 be kernels on $\mathcal{X}$. Then αk and $k_1 + k_2$ are also kernels on $\mathcal{X}$.

- $\rightarrow wk_1 + (1 - w)k_2$ is also a kernel on $\mathcal{X}$ for $w \in [0, 1]$.

Support Vector Data Description (SVDD)

- One-class classifier basing on SVM.
- Enclose the data with a (hyper)sphere; points inside are inliers, points outside outliers.
- Optimization problem:
  - $\min R^2 + C \sum_i \xi_i$
  - with constraints
    $\|\phi(x_i) - a\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0 \quad \forall i$
- Parameter C controls the amount of allowed outliers. Recommendation: $C = \frac{1}{\#\text{outliers}}$

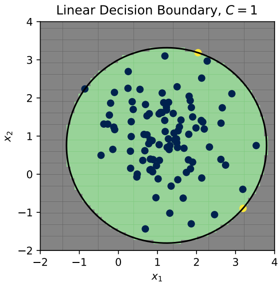

Figure: Example of a linear decision boundary ($n = 100$, $k(\mathbf{x}_1, \mathbf{x}_2) = \langle \mathbf{x}_1, \mathbf{x}_2 \rangle$).

Effects of a Gaussian / RBF Kernel

- For illustrative purposes: $k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\gamma \left\|\mathbf{x}_1 - \mathbf{x}_2\right\|^2\right)$.

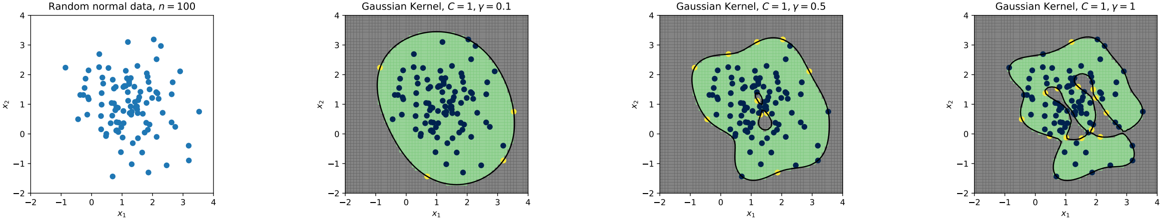

Figure: Effects of parameter γ on the decision boundary, with C fixed.

Kernels for Time Series

- Time series $x = (x_i)_{i=1}^{n}$, $y = (y_i)_{i=1}^{n'}$ $(x_i, y_i \in \mathbb{R})$ are special because the may have different lengths, thus rendering Euclidean distance ($\|x - y\|$?) useless.
- Even with the same lengths: Small shifts in data may have a huge impact.
- People use feature extraction to transform their time series to vectors of fixed size.
  - $\rightarrow$ RBF kernel with the extracted features.
  - Works okay in general. However, as too many features usually reduce the effectiveness of machine learning algorithms — which features to select?
- Feature extraction:
  - needs domain-knowledge
  - automated methods need class labels $\rightarrow$ not applicable to outlier detection (unsupervised).
  - induces information loss (by data-processing inequality).
- Examples: mean, variance, trend, ...
- Typically, people use dynamic time warping (DTW) but DTW is not a distance and does not yield a valid kernel.
  - $\rightarrow$ The theory underlying kernel functions (existence of feature space, optimality of result) does not hold.

Global Alignment Kernels (GAK)

- Similar idea to DTW.
- However, instead of considering the minimum over all alignments, Cuturi et al. [1] consider the "Soft-Minimum" instead $\rightarrow$ the sum over all valid alignments.
- The idea is that if many alignments provide a good fit, then time series can not be too different.
- Why has this not been applied to outlier detection before?
    - We do not know $\rightarrow$ but maybe because there was a bug in libsvm which does not allow the use of precomputed Gram matrices with SVDD.

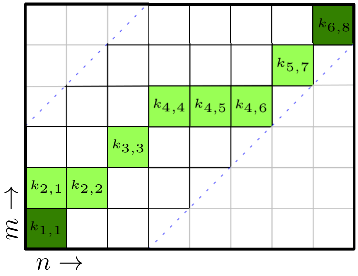

Figure: Alignment Grid

Fourier Transformation

Karlsruhe Institute of Technology

- GAKs allow incorporating time information.
  - We propose to also integrate frequency information of time series.
- In particular, we consider the "Fourier transform kernel", which truncates the vectors obtained by the Fourier transformation of the time series as follows:
  - The Fourier transformation of a time series $x = (x_m)_{m=1}^{n}$ is the sequence $X = (X_k)_{k=1}^{n}$ of the Fourier coefficients

$$X_k = \sum_{m=1}^{n} x_m \exp\left\{-2\pi i \frac{(k-1)(m-1)}{n}\right\}, \quad k = 1, \ldots, n,$$

    with $i^2 = -1$ the imaginary number.
  - We truncate the Gaussian kernel taken over the Fourier transformations of x and y, that is,

$$k_{FFT}(x, y) := \exp\left\{-\gamma \sum_{j=1}^{t} (X_j - Y_j)^2\right\}, \tag{1}$$

    with smoothing parameter γ, and $1 \leq t \leq \min(n, n')$. Hence, parameter t controls the quality of the approximation by restricting the number of coefficients.

Multiple Kernel Learning (MKL)

Karlsruhe Institute of Technology

- Consider a convex combination k of kernels k_m

$$k(x, x') = \sum_{m=1}^{M} d_m k_m(x, x'), \text{ with } d_m \geq 0, \quad \sum_{m=1}^{M} d_m = 1,$$

  then finding the Lagrange multipliers α_i and weights d_m is known as multiple kernel learning problem.
- Rakotomamonjy et al. [3] propose the SimpleMKL algorithm, combining gradient descent to find the weights d_m with the standard SVM optimization to find the multipliers α_i.
- To run the gradient descent algorithm, we need the gradient w.r.t. d_m of the problem we wish to optimize (see our article).

Performance

Table: Mean balanced accuracy over 10 runs. Bold print highlights the best results. N/A did not complete in 24 hours.

Data set	MK	DTW	HDR	DOTS	α-hull	ADSL	LOF-DTW
ArrowHead	$\mathbf{0.70 \pm 0.2}$	0.58 ± 0.2	0.67 ± 0.1	0.51 ± 0.1	0.67 ± 0.2	0.49 ± 0.0	0.52 ± 0.1
CBF	$\mathbf{0.66 \pm 0.0}$	0.49 ± 0.1	0.50 ± 0.0	0.49 ± 0.0	0.50 ± 0.0	0.50 ± 0.0	0.65 ± 0.1
Ch.Concent.	0.49 ± 0.0	0.48 ± 0.0	0.50 ± 0.0	0.50 ± 0.0	0.50 ± 0.0	0.50 ± 0.0	$\mathbf{0.63 \pm 0.0}$
ECG200	$\mathbf{0.67 \pm 0.1}$	0.55 ± 0.1	0.50 ± 0.0	0.55 ± 0.1	0.50 ± 0.1	0.52 ± 0.0	0.65 ± 0.1
ECGFiveDays	0.64 ± 0.0	0.58 ± 0.0	0.52 ± 0.0	0.54 ± 0.0	0.52 ± 0.0	0.50 ± 0.0	$\mathbf{0.77 \pm 0.0}$
GunPoint	$\mathbf{0.72 \pm 0.1}$	0.61 ± 0.1	0.49 ± 0.0	0.64 ± 0.1	0.50 ± 0.0	0.62 ± 0.1	0.70 ± 0.1
Ham	$\mathbf{0.51 \pm 0.1}$	0.48 ± 0.1	0.49 ± 0.0	0.48 ± 0.0	0.49 ± 0.0	0.49 ± 0.0	0.49 ± 0.0
Herring	$\mathbf{0.52 \pm 0.1}$	0.51 ± 0.1	0.50 ± 0.1	0.50 ± 0.1	0.47 ± 0.0	0.50 ± 0.0	0.50 ± 0.1
Lightning2	0.57 ± 0.2	0.49 ± 0.2	0.48 ± 0.0	0.50 ± 0.1	0.51 ± 0.1	0.64 ± 0.1	$\mathbf{0.72 \pm 0.2}$
MoteStrain	$\mathbf{0.70 \pm 0.0}$	0.62 ± 0.1	0.52 ± 0.0	0.61 ± 0.0	0.52 ± 0.0	0.51 ± 0.0	0.55 ± 0.0
Strawberry	0.69 ± 0.1	0.70 ± 0.0	0.47 ± 0.0	0.68 ± 0.0	0.48 ± 0.0	0.56 ± 0.0	$\mathbf{0.76 \pm 0.0}$
ToeSeg1	0.65 ± 0.1	0.50 ± 0.1	0.49 ± 0.0	0.47 ± 0.0	0.48 ± 0.0	0.61 ± 0.0	$\mathbf{0.73 \pm 0.1}$
ToeSeg2	$\mathbf{0.67 \pm 0.1}$	0.48 ± 0.1	0.51 ± 0.0	0.48 ± 0.0	0.52 ± 0.0	0.60 ± 0.0	0.61 ± 0.1
Wafer	$\mathbf{0.65 \pm 0.0}$	0.64 ± 0.0	0.49 ± 0.0	N/A	0.49 ± 0.0	0.50 ± 0.0	0.56 ± 0.0
Wine	0.48 ± 0.1	0.50 ± 0.1	0.60 ± 0.1	0.56 ± 0.1	$\mathbf{0.65 \pm 0.2}$	0.54 ± 0.1	0.58 ± 0.1

- $\rightarrow$ competitive results on standard benchmark data (best on 9/15).

Runtime

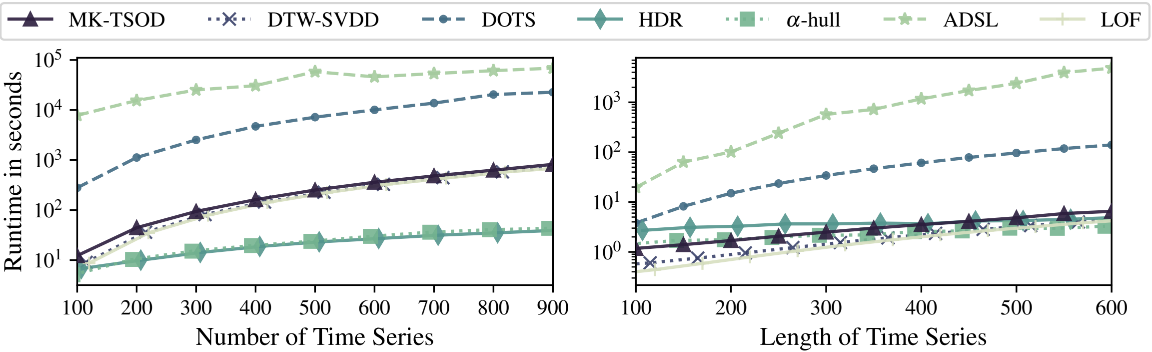

Figure: Runtime analysis. We report the median runtime of five independent runs.

Ablation Study

Table: Ablation analysis. Mean BA over five runs. Bold print highlights the best results.

Data set	MK-TSOD	FFT-SVDD	GAK-SVDD
ArrowHead	**0.70 $\pm$ 0.2**	0.65 $\pm$ 0.1	0.61 $\pm$ 0.2
CBF	**0.66 $\pm$ 0.0**	0.60 $\pm$ 0.1	**0.66 $\pm$ 0.0**
Ch.Concent.	0.49 $\pm$ 0.0	**0.52 $\pm$ 0.0**	0.48 $\pm$ 0.0
ECG200	**0.67 $\pm$ 0.1**	0.63 $\pm$ 0.1	0.62 $\pm$ 0.1
ECGFiveDays	0.64 $\pm$ 0.0	**0.65 $\pm$ 0.0**	0.62 $\pm$ 0.1
GunPoint	**0.72 $\pm$ 0.1**	0.65 $\pm$ 0.1	0.64 $\pm$ 0.1
Ham	**0.51 $\pm$ 0.1**	0.47 $\pm$ 0.1	0.50 $\pm$ 0.1
Herring	**0.52 $\pm$ 0.1**	0.49 $\pm$ 0.1	0.51 $\pm$ 0.1
Lightning2	0.57 $\pm$ 0.2	**0.67 $\pm$ 0.1**	0.47 $\pm$ 0.1
MoteStrain	**0.70 $\pm$ 0.0**	0.62 $\pm$ 0.0	0.67 $\pm$ 0.1
Strawberry	0.69 $\pm$ 0.1	0.71 $\pm$ 0.1	**0.73 $\pm$ 0.0**
ToeSeg1	**0.65 $\pm$ 0.1**	**0.65 $\pm$ 0.1**	0.62 $\pm$ 0.1
ToeSeg2	**0.67 $\pm$ 0.1**	0.55 $\pm$ 0.1	0.57 $\pm$ 0.1
Wafer	**0.65 $\pm$ 0.0**	0.62 $\pm$ 0.0	**0.65 $\pm$ 0.0**
Wine	0.48 $\pm$ 0.1	**0.54 $\pm$ 0.1**	0.42 $\pm$ 0.0

Summary

- We combine global alignment kernels, Fourier transform kernels, and multiple kernel learning with support vector data description for time series outlier detection.
- Contributions:
  - SVDD-based outlier detection algorithm.
  - GAK+FFT only, no feature engineering.
  - One hyperparameter, C, the expected outlier ratio.
- Possibilities for future work:
  - Domain-dependent kernels, for example, for energy data.
  - Other heuristics for γ.
  - Apply within active learning.

References I

[1] Marco Cuturi et al. “A Kernel for Time Series Based on Global Alignments”. In: *ICASSP (2)*. 2007.

[2] Robert T Olszewski. “Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data”. PhD thesis. School of Computer Science Carnegie Mellon University, 2001.

[3] Alain Rakotomamonjy et al. “SimpleMKL”. In: *Journal of Machine Learning Research* 9 (2008).

[4] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. 2008.

[5] David M. J. Tax and Robert P. W. Duin. “Support Vector Data Description”. In: *Machine Learning* 54.1 (2004).