Sinkhorn Divergence of Topological Signature Estimates for Time Series Classification

Colin Stephen
School of Computing, Electronics and Mathematics, Coventry University UK
Email: colin.stephen@coventry.ac.uk

***Abstract*—Distinguishing between classes of time series sampled from dynamic systems is a common challenge in systems and control engineering, for example in the context of health monitoring, fault detection, and quality control. The challenge is increased when no underlying model of a system is known, measurement noise is present, and long signals need to be interpreted. In this paper we address these issues with a new non parametric classifier based on topological signatures. Our model learns classes as weighted kernel density estimates (KDEs) over persistent homology diagrams and predicts new trajectory labels using Sinkhorn divergences on the space of diagram KDEs to quantify proximity. We show that this approach accurately discriminates between states of chaotic systems that are close in parameter space, and its performance is robust to noise.**

I. Introduction

Automatic labelling of time series is a significant challenge in many scientific applications: predicting cardiac pathologies from electrocardiogram (ECG) traces, the imminent failure of engine components from vibration measurements, or the type of a star from its light spectrum variations for example. Features that are invariant under transformations such as nonlinear warping in the time domain are often important for effective predictions in these contexts because measurement sampling rates may differ between samples, the same physical processes may evolve at different rates for different samples, or the morphology of the signal during important events may be the deciding factor between classes. So a wide range of methods for time series classification based on different invariances are available, offering a variety of performance characteristics suited to different applications [1], [2]. In the case of dynamic systems, it is also known that signal decomposition and interpretation methods such as spectral and cepstral analysis, and phase space reconstructions using Takens embedding theorem, provide useful features to interpret and compare the states of systems [3]–[5].

The approach taken in this paper compares global descriptors of time series, like symbolic aggregate approximations (SAX) or bags of patterns (BOP) do, but without the need to decompose signals and using a metric that is invariant to nonlinear time domain transformations, like dynamic time warping (DTW) is. In particular *we show how to summarise the topological features of classes of time series in a concise way, and how to quickly quantify similarity between the topology of an unlabelled time series and these class summaries.*

Our method applies techniques from topological data analysis (TDA) in the form of persistent homology (PH) to characterize signals [6], [7]. The central object of interest in PH is the persistence diagram (PD) or barcode which provides a concise and stable formal representation of the topological features present in a data set at *all metric scales* simultaneously [8], [9]. This global multi-scale perspective allows TDA to expose features otherwise overlooked by conventional nonlinear dimensionality reduction techniques. For an introduction see [9], for algorithmic aspects [10], and for examples of metric space representations on the space of PDs needed to use TDA in off-the-shelf machine learning pipelines see [11], [12].

II. Related Work

Time series have previously been analysed using TDA, largely via PH of point clouds constructed using delay embeddings [13]–[16]. However delay embeddings require heuristic estimation of the embedding dimension and delay size to be computed beforehand, increase the influence of noise, and lead to a rapid increase in the complexity of computing PH of filtrations on the point cloud, requiring subsampling techniques such as the witness complex [17]–[19]. One previous study avoids embeddings but relies on a coarse grained statistic derived from PDs to characterise time series, the persistent entropy [20]. Our method is similar to this since we use a filtration on the time series directly, however we use a metric on the space of persistence diagrams directly rather than on the space of persistent entropy histograms.

Our method uses persistence images (PIs) as its underlying stable representation of PH, however rather than using these as feature vectors in a support vector classifier as in [11], we apply a distance metric to them. Following [21] we treat PIs as kernel estimators of the density of expected PDs for a class, and to measure distance between these density estimates and new PIs we use the Sinkhorn divergence [22]. The latter is an upper bound approximation to the Wasserstein distance between distributions which can be computed very quickly [23]–[25]. The Sinkhorn divergence has recently been applied to scalable clustering and averaging of large PDs [26], but not yet to classification using topological features.

Much previous work has been done on characterising and comparing trajectories of dynamical systems, but this often focuses on methods for distinguishing between chaotic and non-chaotic (periodic, quasi-periodic, intermittent) behaviour, for example by analysing spectra of Lyapunov exponents or textures of recurrence plots [27], [28]. The method we develop here is suited to this type of classification but to showcase

978-1-5386-6805-4/18/$31.00 ©2018 IEEE
DOI 10.1109/ICMLA.2018.00113

its fine-grained capabilities we focus numerical experiments on showing that it can distinguish between different chaotic regimes that lie very close to one another in parameter space.

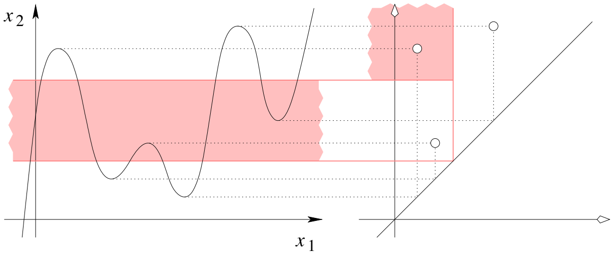
III. Persistent Homology and Optimal Transport

We first outline persistent homology in terms of sub level sets of functions, then the persistence image representation of persistence diagrams, before introducing entropy regularized optimal transport metrics between probability distributions. These topics form the backbone of the classifier pipeline defined and applied in following sections.

A. Persistent Homology and Persistence Images

Given a bounded continuous function $f : \mathbb{X} \to \mathbb{R}$ on a topological space $\mathbb{X}$ define sublevel sets $\mathbb{X}_a := f^{-1}(-\infty, a]$ for each a in $\mathbb{R}$. Then given $a \leq b$ the inclusion $\mathbb{X}_a \subseteq \mathbb{X}_b$ induces a homomorphism of homology groups: $\mathbf{f}_l^{a,b} : H_l(\mathbb{X}_a) \to H_l(\mathbb{X}_b)$ for each dimension l. Under mild conditions on f, for any $\delta > 0$ the homomorphism $\mathbf{f}_l^{c-\delta,c}$ is *not* an isomorphism for only finitely many values of $c \in \mathbb{R}$ for all l, and $H_l(\mathbb{X}_a)$ is finitely generated [29]. This guarantees that the following procedure results in a finite data structure: step through the values of c at which the homology of $\mathbb{X}_a$ changes and record the maximal intervals $[b, d] \subset \mathbb{R}$ such that homology classes appearing in some $\mathbb{X}_a$ live in precisely one interval and no where else. The filtration values $b, d \in \mathbb{R}$ describing each such interval are often called the *birth* and *death* values of the corresponding topological feature in the filtration. The translation between a one dimensional space and its sub level set (birth, death) pairs is particularly easy to visualize (Fig.1).

Fig. 1. Mapping critical points of a function (left) to (birth, death) pairs in a persistence diagram (right). From Edelsbrunner and Harer, "Persistent Homology – A Survey" [30].



Persistent homology became an effective way to quantify changes in topology across scales when it was recognised that by adding points in $\Delta = \{(x, x) | x \in \mathbb{R}\}$ with countably infinite multiplicity to the multiset of (birth, death) pairs arising from the process above, the resulting persistent homology *barcode* or *persistence diagram* (PD) is a stable representation under perturbations to f [29]. In other words there is then a natural metric on PDs bounded above by some distance between the underlying functions [9], [31], [32]. Frequently this stability is stated as a bound on the p-Wasserstein distance between sublevel set PDs of Lipschitz functions with respect to the L_∞ metric:

$$d_W^p(X, Y) := \left(\inf_\gamma \sum_{x \in X} \|x - \gamma(x)\|_\infty^p \right)^{\frac{1}{p}} \leq C \|f - g\|_\infty^d$$

where diagrams X, Y are PDs for f, g, the constants C, d are independent of f, g, and γ ranges over all bijections between the countably infinite multisets X, Y (see [8] for details). Because of this the Wasserstein distance has been at the heart of applications of TDA as it has developed. However due to the complexity of optimizing over bijections [33] and because it imposes a complicated geometry on the space of diagrams [34] there has been a glut of alternative vector space representations.

One such representation is the discrete persistence image (PI) which in turn is defined in terms of a continuous persistence surface (PS) arising from a PD [11]. The latter is a way to estimate the distribution from which a diagram is sampled while ensuring that the estimate itself is stable. Given a diagram $D \in \mathbb{R}_{y \geq x}$ as defined above, note that the persistence value $p := d - b$ is at least zero for all points so we can translate D to the positive quadrant via $T : (b, d) \mapsto (b, p)$. Next consider a weighted sum of Gaussians centered at points $x \in T(D)$:

$$\rho_D(z) := \frac{1}{2\pi\sigma^2} \sum_{x \in T(D)} f(x) e^{-\frac{\|z - x\|^2}{2\sigma^2}},$$

where $f : \mathbb{R}^2 \to \mathbb{R}$ is a continuous, piecewise differentiable function that decays to zero on the horizontal axis. This is the *persistence surface* associated to D. It is a stable estimate of the distribution underlying D given the constrains on f (see [11], [21] for details). To provide a vector space representation of ρ_D suitable for use in machine learning applications, a finite regular grid G is then placed over part of $\mathbb{R}_+^2$ and used to quantize ρ_D as a collection of pixels $I(\rho_D)_g := \iint_g \rho_D(z)\, dz$, corresponding to each cell $g \in G$. This collection of pixels is the *persistence image* representation of D and it is a suitable vector representation for off-the-shelf classifiers such as support vector machines (SVMs) to use [11].

B. Entropy Regularized Discrete Optimal Transport

The question of optimizing transport costs between distributions of resources has been studied in various forms since the 1700s, and in its modern guise is central to statistical learning theory [35], [36]. Given two n-bin histograms r, c, a *transport plan* between r and c is a matrix $P \in \mathbb{R}_+^{n \times n}$ satisfying $\sum_i p_{ij} = r$ and $\sum_j p_{ij} = c$. Equivalently, P is a joint probability $P = P(X, Y)$ for two multinomial random variables X and Y taking values in $\{0, 1, \ldots, n-1\}$, whose marginals are r and c. We write $U_n(r, c)$ for the set of all transport plans for n-bin histograms r, c, dropping the subscript n when it is clear from context.

Given a transport plan P, we interpret p_{ij} as a mass to be transported from the i-th component of r to the j-th component of c. If the cost of this operation is $m_{ij} \in \mathbb{R}$ per unit of mass transported, then the *discrete optimal transport* (OT) *problem* is to minimise the sum of transport costs over all possible plans for r, c given M:

$$OT_M(r,c) := \min_{P \in U(r,c)} \langle P, M \rangle$$

where $\langle P, M \rangle$ is the Frobenius product $\sum_{i,j=1}^{n} p_{ij} m_{ij}$. The function OT_M is a metric on the space of histograms when M is itself a metric distance matrix [35], [37] but computing it exactly is difficult in practice, with the worst case time complexity $O(n^3 \log n)$ being reached with certain values of r, c and M [38].

Recent work showed that regularizing the classical OT_M problem by adding a convex constraint can lead to fast approximations [22]–[25], [39], [40]. Define the entropy of a transport plan P as $H(P) := -\sum_{i=1}^{n} p_{ij} \log p_{ij}$, then minimising the sum of transport costs over *high entropy* transport plans as in

$$ROT_M^{\lambda \geq 0}(r,c) := \langle P^{\lambda}, M \rangle$$

where

$$P^{\lambda} := \operatorname{argmin}_{P \in U(r,c)} \left(\langle P, M \rangle - \lambda H(P) \right)$$

gives an upper bound approximation to OT_M that has complexity $O(n^2 \log n)$ [23].[1] This upper bound is called the *Sinkhorn divergence* between the histograms since it has a natural parallel implementation based on iterated matrix-vector products known as the Sinkhorn Knopp (SK) algorithm [22]. Early evidence suggests that it gives better classification results than OT_M in a number of experiments and that it converges very quickly in practice [23]. Moreover, when the cost matrix M is highly structured as is the case for L_p distances on regular grids, the matrix operations of the algorithm can be speeded up further via FFT based convolutions [25]. This speed and accuracy advantage has increased interest in Sinkhorn divergence and other regularized variants of OT_M for a variety of problems ranging from color transfer in image processing to model optimization in machine learning [24], [41]. In the following sections we show that it can be integrated successfully within classification pipelines based on topological features as well.

IV. Sinkhorn Divergence of Topological Signature Estimates

The problem we address now is to classify time series generated by deterministic dynamical systems. Suppose we are given time series samples from two classes corresponding to a choice of parameters in a single dynamical model. Can we characterize the data in terms of their shared or distinct topological properties without resorting to embeddings or parametric methods, and use this to effectively predict the class labels of new series? Represent a dataset containing m samples each of length n using an array $X = (x_i^j)_m^n$, and represent class labels using a 0-1 vector $y = (y_i)_m$.[2]

Training:

T1 For each sequence x_i directly compute its sublevel set persistence diagram D_i.

T2 Partition the set of m PDs according to their associated class labels in y, giving two sets of PDs: $D^{\dagger} = \{D_i \,|\, y_i = 0\}$ and its compliment $D^{\ddagger} = \{D_i \,|\, y_i = 1\}$.

T3 For each of $D^{\dagger}$ and $D^{\ddagger}$ overlay the points in its member diagrams to form a combined persistence diagram representing the whole class: $\overline{D}^{\dagger} = \bigcup D^{\dagger}$ and $\overline{D}^{\ddagger} = \bigcup D^{\ddagger}$.

T4 Choose a continuous and piecewise differentiable function $f : \mathbb{R}_+^2 \to \mathbb{R}$ such that $f(x, 0) = 0$ for all x, and a smoothing radius σ. Construct the smoothed persistence surfaces $\rho_{\overline{D}^{\dagger}}$ and $\rho_{\overline{D}^{\ddagger}}$.

T5 Choose a $d \times d$ square grid G that extends beyond the largest values of b and p in $\overline{D}^{\dagger} \cup \overline{D}^{\ddagger}$. Compute the persistence images $I(\rho_{\overline{D}^{\dagger}})$ and $I(\rho_{\overline{D}^{\ddagger}})$ over the cells of G.

In practice the form of f, the value σ, and the size of G can all be set at this stage using cross validation on the training data. After stage T5 we have for each class a stable kernel estimate of the density of its expected persistence diagram, which naturally leads to the following prediction pipeline.

Prediction:

P1 Given an unlabelled query sequence q compute its persistence image I_q using the pipeline above but skipping the diagram overlay steps T2 and T3, and reusing the same values for f, σ and G chosen in T4.

P2 Choose a p value for an underlying L_p metric on the grid G, which induces a cost matrix M_p on G. Also choose a regularization parameter $\lambda \geq 0$ for computing Sinkhorn divergences $ROT_{M_p}^{\lambda}$. Compute the Sinkhorn divergences

$$d^{\dagger} = ROT_{M_p}^{\lambda}(I_q, I(\rho_{\overline{D}^{\dagger}})), \qquad d^{\ddagger} = ROT_{M_p}^{\lambda}(I_q, I(\rho_{\overline{D}^{\ddagger}})).$$

P3 If $d^{\dagger} < d^{\ddagger}$ then predict $y = 0$, if $d^{\ddagger} < d^{\dagger}$ then predict $y = 1$, else predict $y = 0$ or $y = 1$ with equal probability.

In practice the values of p and λ can both be optimized using cross validation during the training phase.

Thus our model predicts labels for new time series based on the closest expected persistence diagram for each class in the training set, using the entropy regularized optimal transport distance between the distributions.

Implementation: Computing the sub level set persistence of each time series at stages T1 and P1 requires determining its critical points (local maxima and minima) and also noting for each local maximum which of its two neighbouring minima is closer in value. Thus the critical points must be sorted as part of the process, which is an $O(n \log n)$ operation at worst and $O(mn \log n)$ for each class.

To compute the values of the persistence image pixels $I_g(\rho_{\overline{D}^{\dagger}})$ and $I_g(\rho_{\overline{D}^{\ddagger}})$ for $g \in G$ various numerical integration and approximation methods are available. In particular if we assume that each point appearing in a cell is centered in that cell we can approximate the persistence surfaces $\rho_{\overline{D}^{\dagger}}$ and $\rho_{\overline{D}^{\ddagger}}$ by convolving their underlying $d \times d$ f-weighted histograms with a discrete filter corresponding to our chosen Gaussian.

[1] Different penalty functions lead to approximations with different convergence characteristics [24], but we consider only entropy in this paper.

[2] The model we define is equally valid for classifying time series of different lengths and with more than two classes, with minor adjustments.

This allows us to compute the persistence images generated by T4 and T5 in a single step, in $O(d^2 \log d)$ for our grid.

Finally during prediction, computing the regularized optimal transport cost between two d^2-bin histograms for a cost matrix M_p corresponding to L_p distances on the grid is $O(d^2 \log d)$. This is because M_p is a block Toeplitz of Toeplitz blocks (BTTB) matrix in this situation, meaning the matrix-vector products appearing in the Sinkhorn-Knopp (SK) algorithm used to compute ROT_{M_p} can be computed using FFT enhanced convolutions. See [22], [23] for details of the SK algorithm and in paricular Chapter 5 of [42] for details of how to speed up the matrix-vector operations.

The result is that once the size d of the grid has been set during training the complexity of the model is $O(n \log n)$ in time series length.

V. Classification Experiments[3]

We call the method above 'Persistence Image Classification using Regularized Optimal Transport', or PICROT for short. This section assesses its performance against one classifier using signal frequency and rate of change analysis, and one classifier based on persitent entropy as defined and successfully applied to similar problems in [43].[4]

C_1 PICROT compares kernel estimates of PD densities using the Sinkhorn divergence. We fix a weight function for PICROT that increases rapidly from zero to one in an interval less than the persistence value of any off diagonal points processed. Thus in effect we apply uniform weights when constructing persistence images. The smoothing parameter σ and grid size d for PIs, and the regularization parameter λ for the Sinkhorn metric, are all estimated using 5-fold cross validation over a grid of candidate values during training.

C_2 CEPS is the one nearest neighbor classifier using Euclidean distance between coefficients of the discrete cosine transforms of the cepstra of the time series being compared:

$$d_{\text{CEPS}}(T_1, T_2) := \left(\sum_i |\text{CEPS}(T_1)_i - \text{CEPS}(T_2)_i|^2 \right)^{\frac{1}{2}}$$

where

$$\text{CEPS}(T) := \text{DCT}\left(\left| \mathcal{F}^{-1} \left\{ \log(|\mathcal{F}(t)|^2) \right\} \right|^2 \right),$$

$\mathcal{F}$ is the Fourier transform, and the sum is over all coefficients

C_3 PENT is the one nearest neighbor classifier using the absolute difference between persistent entropies [20]. If $D(T) = \{(b_i, d_i) \,|\, i \in I\}$ is an indexed set of the off-diagonal points in the persistence diagram associated to T, $p_i := (d_i - p_i)$ is the persistence of each point and $P := \sum_i p_i$ is the total persistence of the diagram, then

$$d_{\text{PENT}}(T_1, T_2) := |\text{PENT}(T_1) - \text{PENT}(T_2)|$$

where

$$\text{PENT}(T) := -\sum_i \frac{p_i}{P} \log\left(\frac{p_i}{P}\right).$$

CEPS is not commonly used as a general time series similarity measure but it is effective when the series have an underlying regularity or cyclicity [5], [44] as with the problems here. The measure d_{CEPS} captures information about the relative rates of change of the two signals across their frequency bands. Cross validating the number of leading terms compared in the cepstral classifier did not improve results, so we compare the entire cepstra. On the other hand d_{PENT} compares signals using a coarse grained statistic derived from their persistence diagrams. We also use one nearest neighbour here because it shows higher accuracy than the receiver operating characteristic (ROC) optimized threshold approach originally appearing in [20].

Data: Binary time series classification problems were generated using the combinations of parameter values for the systems defined in Table I (13 parameter value combinations). The parameter ranges were chosen to ensure that all trajectories studied here are chaotic. Initial conditions and parameter values themselves were varied uniformly in the given intervals to ensure a wide variety of trajectories were observed.

In the case of the Henon and Lorenz systems only the x values were used. In all cases the raw time series were z normalized before any training or predictions and the first 1,000 sequence values were discarded. The subsequent lengths used for classification varied from 2,500 to 15,000 in steps of 2,500 (6 lengths total). White noise was added to each sample varying in standard deviation from 0.0 (no noise) to 0.75 in steps of 0.125 (7 noise levels in total). Thus there were 546 classification experiment configurations in total. Two example time series snippets from different classes in the fourth Henon configuration are shown in Figure 2.

For each of the 546 configurations 200 samples balanced between the two classes were generated, giving a total of 109,200 time series to be processed on each run. The data for each experiment were split randomly in to 170 training and 30 test time series to be classified. This process was then repeated for 10 runs per experiment, using different random train-test splits each time.

Results: The classifier PICROT outperformed the benchmarks in the vast majority of the experiments, as shown in Figure 4. In particular it tended to maintain higher relative accuracy of predictions against the benchmarks as noise levels were increased. This is most easily seen in Figure 3 where we visualize typical accuracy profiles. In the case of the Lorenz system experiments, PICROT was much more accurate than the PENT classifier, outperformed CEPS on shorter sequences,

[3] Sklearn-compatible Python code implementing the classifiers described here can be found at https://github.com/colinstephen/icmla2018

[4] General purpose time series classifiers such as those benchmarked in [1] do not seem to perform well for the dynamic systems considered here. Initial results using *dynamic time warping* (DTW) and *random forests* were not competitive in terms of accuracy, while the potentially high-performance *collective of transformation ensembles* (COTE) and the *elastic ensemble* (EE) methods were too slow to evaluate due to the lengths of time series used here.

TABLE I

PARAMETERS USED TO GENERATE TIME SERIES.

System	Model	Class 0	Class 1
Logistic	$x_{n+1} = ax_n(1 - x_n)$ $x_0 \in (0, 1)$ uniform	$a = 3.9995 \pm 0.0005$	$a = 3.9945 \pm 0.0005$ $a = 3.9895 \pm 0.0005$ $a = 3.9845 \pm 0.0005$ $a = 3.9795 \pm 0.0005$
Henon	$x_{n+1} = 1 - ax_n^2 + y_n$ $y_{n+1} = bx_n$ $x_0, y_0 \in (1, 2)$ uniform	$a = 1.4 \pm 0.0025$ $b = 0.3035 \pm 0.0025$	$b = 0.3085 \pm 0.0025$ $b = 0.3135 \pm 0.0025$ $a = 1.395 \pm 0.0025$ $a = 1.390 \pm 0.0025$ $a = 1.385 \pm 0.0025$
Lorenz	$\dot{x} = \sigma(y - x)$ $\dot{y} = x(\rho - z) - y$ $\dot{z} = xy - \beta z$ $x_0, y_0, z_0 \in (0, 1)$ uniform	$\beta = 8/3$ $\sigma = 10$ $\rho = 28$	$\rho = 27.75$ $\rho = 27.50$ $\rho = 27.25$ $\rho = 27.00$

and matched its performance on longer ones. In experiments generated by the Henon system PICROT outperformed both benchmarks for almost all combinations of sequence length and noise level when the parameter a was varied. This is clearly illustrated in the almost completely white grayscale maps representing the rank performance of PICROT against both of its competitors in Figure 4. Another trend visible in Figure 4 is that the sweet spot for accuracy of PICROT outperforming the others is at mid-range levels of noise (around a standard deviation of 0.375) and mid-range series lengths of around 7500 to 10000 points. Nevertheless white areas constitute over 85% of these maps on average, suggesting that PICROT is a strong performer in a wide variety of situations.

One additional benefit of using PICROT on time series is that it is possible to visualise the topological signatures of the training data classes. This helps to gain an insight into whether or not this topological feature is a ‘strong candidate’ for discriminating classes in a given situation. For example in Figure 2 we visualise the difference in PD class densities associated to the generated time series snippets. The structure and clear separation of locations of the ‘plumes’ in the third diagram suggest that the estimated PD density is likely to be a good feature in this case, while a more diffuse pattern may indicate that hyperparameters of the model may need to be tuned further or that noise is too dominant.

VI. CONCLUSION

We defined a stable and scalable metric on the space of persistence diagrams based on entropic smoothing of optimal transport distances. Following [11] our approach makes use of weight functions applied to kernel density estimates on PDs to ensure our representations are 1-Wasserstein stable, but we treat the resulting quantized kernel density estimates as two dimensional histograms after renormalization. This allows direct application of regularized transport methods [22], [25].

Unlike existing topological methods for time series PICROT avoids the complexity explosion and noise amplification associated with high dimensional reconstructions based on Takens’ embedding theorem. In contrast to traditional signal decomposition methods noise removal is not required for the method to perform well. After cross validation during training the prediction compexity of the model is $O(n \log n)$ with respect to time series length, so it can be applied to long sequences effectively. In addition the underlying metric has a natural parallel implementation (the Sinkhorn Knopp algorithm) suitable for GPUs meaning it is highly scalable. To illustrate these benefits we conducted experiments classifying trajectories of chaotic deterministic systems under a range of signal to noise ratios and parameter values, finding that PICROT is more accurate than two specialized benchmarks in the majority of situations.

REFERENCES

[1] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, “The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances,” *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.

[2] P. Esling and C. Agon, “Time-series data mining,” *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, pp. 1–34, 2012.

[3] H. Kantz and T. Schreiber, *Nonlinear time series analysis*. Cambridge university press, 2004, vol. 7.

[4] L. H. Koopmans, *The spectral analysis of time series*. Elsevier, 1995.

[5] R. B. Randall, “A history of cepstrum analysis and its application to mechanical problems,” *Mechanical Systems and Signal Processing*, vol. 97, pp. 3–19, 2017.

[6] G. Carlsson, “Topology and data,” *Bulletin of the American Mathematical Society*, vol. 46, pp. 255–308, 2009.

[7] H. Edelsbrunner and D. Morozov. (2014) Persistent Homology: Theory and Practice. [Online]. Available: http://pub.ist.ac.at/ edels/Papers/2012-P-11-PHTheoryPractice.pdf

[8] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko, “Lipschitz Functions Have L_p-Stable Persistence,” *Foundations of Computational Mathematics*, vol. 10, pp. 127–139, 2010.

[9] R. Ghrist, “Barcodes: the persistent topology of data,” *Bulletin of the American Mathematical Society*, vol. 45, no. 1, pp. 61–75, 2008.

[10] H. Edelsbrunner and J. Harer, *Computational topology: An introduction*. American Mathematical Society, 2010.

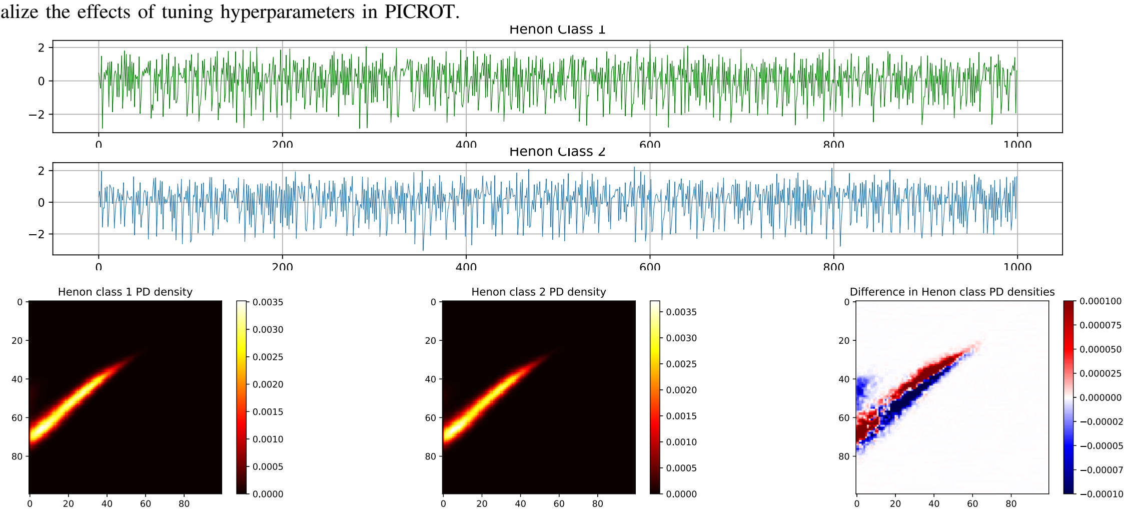
[11] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier, “Persistence images: A stable vector representation of persistent homology,” *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 218–252, 2017.

[12] J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt, “A Stable Multi-Scale Kernel for Topological Machine Learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4741–4748.

[13] F. A. Khasawneh and E. Munch, “Chatter detection in turning using persistent homology,” *Mechanical Systems and Signal Processing*, vol. 70, pp. 527–541, 2016.

[14] J. A. Perea and J. Harer, “Sliding Windows and Persistence: An Application of Topological Methods to Signal Analysis,” *Foundations of Computational Mathematics*, vol. 15, no. 3, pp. 799–838, 2014.

Fig. 2. Short segments from two configurations of a Henon system (top). The associated expected class densities for the full training set (bottom). The latter can help visualize the effects of tuning hyperparameters in PICROT.



[15] C. M. M. Pereira and R. F. de Mello, “Persistent homology for time series and spatial data clustering,” *Expert Systems with Applications*, vol. 42, pp. 6026–6038, 2015.

[16] L. M. Seversky, S. Davis, and M. Berger, “On Time-series Topological Data Analysis: New Data and Opportunities,” *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 59–67, 2016.

[17] Z. Alexander, E. Bradley, J. D. Meiss, and N. Sanderson, “Simplicial multivalued maps and the witness complex for dynamical analysis of time series,” *SIAM J. Applied Dynamical Systems*, vol. 14, no. 3, pp. 1278–1307, 2015.

[18] V. De Silva and G. Carlsson, “Topological estimation using witness complexes,” in *Proceedings of the First Eurographics Conference on Point-Based Graphics*, ser. SPBG’04. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2004, pp. 157–166.

[19] N. Sanderson, E. Shugerman, S. Molnar, J. D. Meiss, and E. Bradley, “Computational Topology Techniques for Characterizing Time-Series Data,” in *International Symposium on Intelligent Data Analysis*. Springer, 2017, pp. 284–296.

[20] M. Rucco, R. Gonzalez-Diaz, M.-J. Jimenez, N. Atienza, C. Cristalli, E. Concettoni, A. Ferrante, and E. Merelli, “A new topological entropy-based approach for measuring similarities among piecewise linear functions,” *Signal Processing*, vol. 134, pp. 130–138, 2017.

[21] F. Chazal and V. Divol, “The density of expected persistence diagrams and its kernel based estimation,” *arXiv preprint arXiv:1802.10457*, 2018.

[22] M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transport,” in *Advances in neural information processing systems*, 2013, pp. 2292–2300.

[23] J. Altschuler, J. Weed, and P. Rigollet, “Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration,” in *Advances in neural information processing systems*, 2017, pp. 1964–1974.

[24] M. Blondel, V. Seguy, and A. Rolet, “Smooth and Sparse Optimal Transport,” *arXiv preprint arXiv:1710.06276*, 2017.

[25] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas, “Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 66, 2015.

[26] T. Lacombe, M. Cuturi, and S. Oudot, “Large Scale computation of Means and Clusters for Persistence Diagrams using Optimal Transport,” *arXiv preprint arXiv:1805.08331*, 2018.

[27] N. Marwan, M. C. Romano, M. Thiel, and J. Kurths, “Recurrence plots for the analysis of complex systems,” *Physics Reports*, vol. 438, no. 5-6, pp. 237–329, 2007.

[28] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, “Determining lyapunov exponents from a time series,” *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, 1985.

[29] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, “Stability of persistence diagrams,” *Discrete & Computational Geometry*, vol. 37, no. 1, pp. 103–120, 2007.

[30] H. Edelsbrunner and J. Harer, “Persistent homology—a survey,” in *Surveys on discrete and computational geometry: twenty years later: AMS-IMS-SIAM Joint Summer Research Conference, June 18-22, 2006, Snowbird, Utah*, vol. 453, 2008, pp. 257–282.

[31] M. d’Amico, P. Frosini, and C. Landi, “Optimal matching between reduced size functions,” *DISMI, Univ. di Modena e Reggio Emilia, Italy, Technical report*, no. 35, 2003.

[32] M. Vejdemo-Johansson, “Sketches of a platypus: persistent homology and its algebraic foundations,” *arXiv preprint arXiv:1212.5398*, 2012.

[33] M. Kerber, D. Morozov, and A. Nigmetov, “Geometry helps to compare persistence diagrams,” *Journal of Experimental Algorithmics (JEA)*, vol. 22, pp. 1–4, 2017.

[34] K. Turner, “Means and medians of sets of persistence diagrams,” *arXiv preprint arXiv:1307.8300*, 2013.

[35] C. Villani, *Topics in optimal transportation*. American Mathematical Society, 2003.

[36] ——, *Optimal transport: old and new*. Springer, 2008.

[37] D. Avis, “On the extreme rays of the metric cone,” *Canad. J. Math*, vol. 32, no. 1, pp. 126–144, 1980.

[38] O. Pele and M. Werman, “Fast and robust earth mover’s distances,” in *Computer vision, 2009 IEEE 12th international conference on*. IEEE, 2009, pp. 460–467.

[39] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, “Iterative Bregman projections for regularized transportation problems,” *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. A1111–A1138, 2015.

[40] A. Dessein, N. Papadakis, and J.-L. Rouas, “Regularized optimal transport and the ROT mover’s distance,” *arXiv preprint arXiv:1610.06447*, 2016.

[41] A. Genevay, G. Peyré, and M. Cuturi, “Learning generative models with sinkhorn divergences,” *arXiv preprint arXiv:1706.00292*, 2017.

[42] C. R. Vogel, *Computational methods for inverse problems*. Siam, 2002, vol. 23.

[43] M. Rucco, E. Concettoni, C. Cristalli, A. Ferrante, and E. Merelli, “Topological classification of small DC motors,” *2015 IEEE 1st International Forum on Research and Technologies for Society and Industry, RTSI 2015 - Proceedings*, pp. 192–197, 2015.

[44] K. Kalpakis, D. Gada, and V. Puttagunta, “Distance measures for effective clustering of ARIMA time-series,” in *Proceedings 2001 IEEE International Conference on Data Mining*, 2001, pp. 273–280.

Fig. 3. Representative profiles of classifier accuracy vs noise for three system configurations. Lorenz, Henon, and Logistic systems with a fixed chosen set of parameters appear in columns 1 to 3 respectively. Time series length increases down the page. The classifier PICROT based on Sinkhorn divergences between persistence density estimates consistently outperforms the benchmarks.

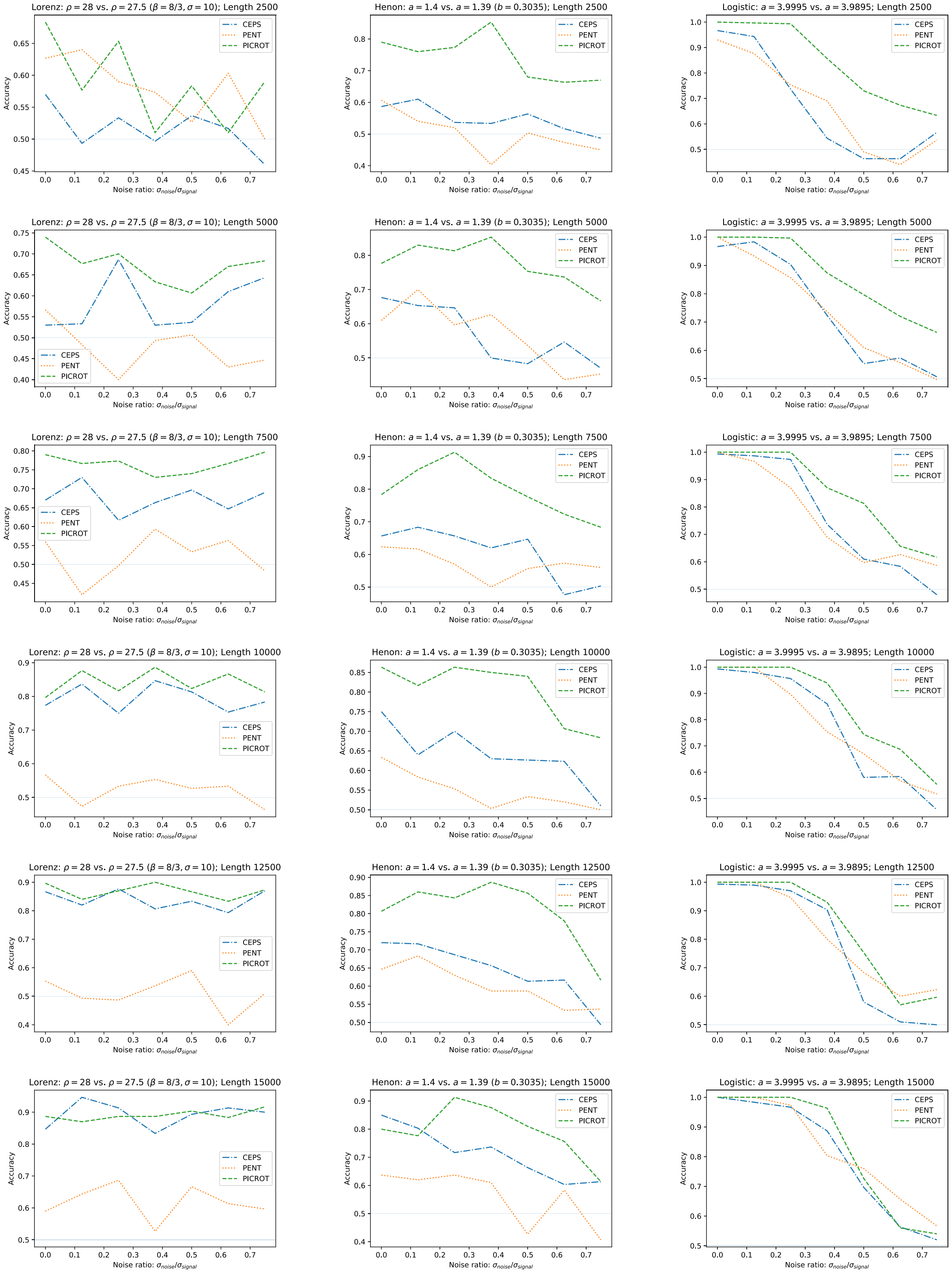
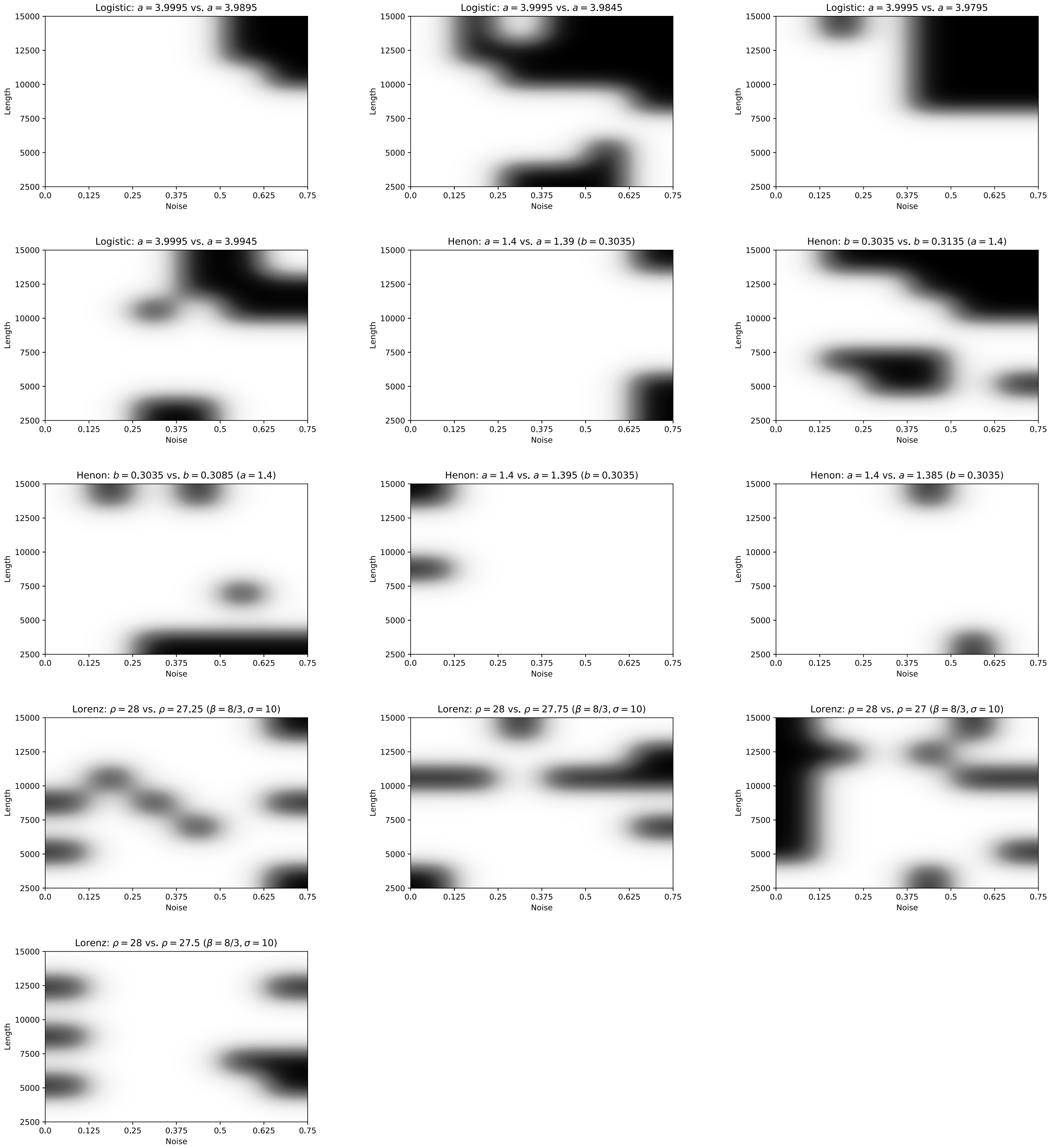


Fig. 4. Grayscale maps representing the rank performance of PICROT against the two benchmarks for combinations of additive noise and signal length. White implies PICROT outperforms *both* benchmarks, while black implies it is outperformed by at least one of the benchmarks. All 13 system configurations outlined in Table I are presented. The majority of each map is white, indicating strong performance of PICROT in a range of situations.


Logistic: $a = 3.9995$ vs. $a = 3.9895$
Logistic: $a = 3.9995$ vs. $a = 3.9845$
Logistic: $a = 3.9995$ vs. $a = 3.9795$
Logistic: $a = 3.9995$ vs. $a = 3.9945$
Henon: $a = 1.4$ vs. $a = 1.39$ ($b = 0.3035$)
Henon: $b = 0.3035$ vs. $b = 0.3135$ ($a = 1.4$)
Henon: $b = 0.3035$ vs. $b = 0.3085$ ($a = 1.4$)
Henon: $a = 1.4$ vs. $a = 1.395$ ($b = 0.3035$)
Henon: $a = 1.4$ vs. $a = 1.385$ ($b = 0.3035$)
Lorenz: $\rho = 28$ vs. $\rho = 27.25$ ($\beta = 8/3, \sigma = 10$)
Lorenz: $\rho = 28$ vs. $\rho = 27.75$ ($\beta = 8/3, \sigma = 10$)
Lorenz: $\rho = 28$ vs. $\rho = 27$ ($\beta = 8/3, \sigma = 10$)
Lorenz: $\rho = 28$ vs. $\rho = 27.5$ ($\beta = 8/3, \sigma = 10$)
Length
Noise