Sequence Likelihood Divergence For Fast Time Series Comparison

Yi Huang, Victor Rotaru and Ishanu Chattopadhyay*
University of Chicago, Chicago, IL

***Abstract*—Comparing and contrasting subtle historical patterns is central to time series analysis. Here we introduce a new approach to quantify deviations in the underlying hidden stochastic generators of sequential discrete-valued data streams. The proposed measure is universal in the sense that we can compare data streams without any feature engineering step, and without the need of any hyper-parameters. Our core idea here is the generalization of the Kullback-Leibler (KL) divergence, often used to compare probability distributions, to a notion of divergence between finite-valued ergodic stationary stochastic processes. Using this notion of process divergence, we craft a measure of deviation on finite sample paths which we call the sequence likelihood divergence (SLD) which approximates a metric on the space of the underlying generators within a well-defined class of discrete-valued stochastic processes. We compare the performance of SLD against the state of the art approaches, $e.g.$, dynamic time warping (DTW) [1] with synthetic data, real-world applications with electroencephalogram (EEG) data and in gait recognition, and on diverse time-series classification problems from the University of California, Riverside (UCR) time series classification archive [2]. We demonstrate that the new tool is at par or better in classification accuracy, while being significantly faster in comparable implementations. Released in the publicly domain, we are hopeful that SLD will enhance the standard toolbox used in classification, clustering and inference problems in time series analysis.**

***Index Terms*—Universal Metric; Dynamic Time Warping; Probabilistic Finite State Automata; Time Series Clustering**

I. Introduction

EFFICIENTLY learning stochastic processes is key to analyzing time-dependency in domains where randomness cannot be ignored. For such learning to occur, we need to define either a measure of deviation or, more generally, a measure of similarity to compare time series. Examples of such similarity measures from the literature include the classical l_p distances and l_p distances with dimensionality reduction [3], the short time series distance (STS) [4], which takes into account of irregularity in sampling rates, the edit based distances [5] with generalizations to continuous sequences [6], the dynamic time warping (DTW) [1], which is used extensively in the speech recognition community, and the data smashing algorithm [7]. A challenge in th existing techniques is differentiating complex stochastic processes with subtle variations in their generative parameters. When presented with finite sample paths from non-trivial processes, a vast majority of the state-of-the-art techniques often focus on their point-wise distance, instead of intrinsic differences in their (potentially hidden) generating processes. Thus, finding subtle deviations in long-memory processes might be difficult. Additionally, in the age of big data, computational cost is crucially important. Despite its impressive performance, the computational speed of the fastest contender (DTW) might still pose a bottleneck in big data applications. SLD addresses both these limitations: it is significantly faster, and demonstrably differentiates data streams in challenge cases indistinguishable to the DTW algorithm (See Section VI-D).

When presented with finite sample paths, SLD estimates deviation between the underlying generators. Our intuition follows from a basic result in information theory: If we know the true distribution $\mathbf{p}$ of the random variable, we could construct a code with average description length $h(\mathbf{p})$, where $h(\cdot)$ is the entropy of a distribution. If, instead, we used the code for a distribution $\mathbf{q}$, we would need $h(\mathbf{p}) + D_{\text{KL}}(\mathbf{p} \,\|\, \mathbf{q})$ bits on average to describe the random variable. Thus, deviation in the distributions show up as an additional contribution from the KL divergence term [8], [9], [10]. This shows that the KL divergence has a concrete meaning: it is the average number of additional bits that a source must transmit to a receiver in order to communicate a random variable with an altered distribution. Now, if we can generalize the notion of KL divergence to processes, then it might be possible to quantify deviations in process dynamics via an increase in the entropy rate by the corresponding divergence.

The generalization of KL divergence to independent identically distributed (i.i.d.) processes is trivial. SLD works by generalizing the idea beyond i.i.d. cases and to a class of stationary and ergodic finite-valued processes. The class of processes we consider are those generated by Probabilistic Finite State Automaton (PFSA) [11], [12],

*Corresponding Author E-mail: ishanu@uchicago.edu

[7], [13]. PFSA are semantically succinct and can model discrete-valued stochastic processes of any finite Markov order and a subclass of processes of no finite Markov order [14], and can approximate arbitrary Hidden Markov Models [12] (HMM). Importantly, the data smashing algorithm [7] is also a PFSA based approach designed to measure similarity between hidden generators. However, the algorithms are distinct (data smashing aims to "invert" the hidden generators algorithmically), and SLD is vastly more efficient.

The remaining of the paper is organized as follows. In Sec. II, we motive the sequence likelihood divergence approach with i.i.d. processes, and propose the SLD measure. We introduce PFSA in Sec. III and the main theoretical considerations for sequence likelihood divergence are developed in Sec. IV. We discuss specifics of implementation issues for SLD in Sec. V and compare SLD with state of the art baselines for clustering and classification tasks in Sec. VI. Finally, in Sec. VII, we demonstrate real world applications.

II. Motivating Example

Continuing with the intuition above, consider sequences of length n generated by two processes $\mathscr{P}_1 = B(.5)$ and $\mathscr{P}_2 = B(.8)$, where $B(p)$ is the Bernoulli process with parameter p [15]. Our objective is to estimate deviations in the binary sample paths generated by these processes. If we treat the sequences as vector of dimension n and use E_{ij} to denote the expected hamming distance between sequences generated by $\mathscr{P}_i$ and $\mathscr{P}_j$, we conclude $E_{11} = E_{12} = E_{21} = .5n$, which implies that two sequences both generated by $B(.5)$ are *not* more alike or dislike than two sequences with one generated by $B(.5)$ and the other, $B(.8)$. Let

$$\begin{aligned} h_1 &= h([.5,.5]), h_2 = h([.8,.2]),\\ d_{12} &= D_{\mathsf{kl}}\left([.5,.5] \,||\, [.8,.2]\right),\\ d_{21} &= D_{\mathsf{kl}}\left([.8,.2] \,||\, [.5,.5]\right), \end{aligned}$$

and $L(x, B(p))$ denote the log-likelihood of $B(p)$ generating x. Defining random vector

$$\mathbf{v}_x = \left(L(x, B(.5)), L(x, B(.8))\right),$$

then, by law of large number [16], we have

$$\lim_{n\to\infty} \mathbf{v}_x = \begin{cases} (h_1, h_1 + d_{12}) & \text{if } x \text{ is generated by } B(.5),\\ (h_2 + d_{21}, h_2) & \text{if } x \text{ is generated by } B(.8). \end{cases}$$

The two limit points are identical if and only if $d_{12} = d_{21}$, which implies $\mathscr{P}_1 = \mathscr{P}_2$, *i.e.*, the vectors $\mathbf{v}_x$ and $\mathbf{v}_y$ are close if the two sample paths x, y are generated by similar processes. While in this particular example the processes might be differentiated from generating a different fraction of 1s, we can construct distinct *non-i.i.d. processes* with same frequency of 1s to make both Hamming distance and such a frequency check fail. Thus, we define:

Definition 1 (Sequence Likelihood Divergence)**.** The SLD $\theta_{\mathcal{G}}$ between two finite sample paths x, y is:

$$\theta_{\mathcal{G}}(x,y) = \sum_{G\in\mathcal{G}} |L(x,G) - L(y,G)| \tag{1}$$

where $\mathcal{G}$ is a fixed finite set of processes which may or may not include the generators for x, y, and $L(x, G)$ is the log-likelihood of x being generated by the model $G \in \mathcal{G}$.

We will show that such a calculation may be carried out efficiently if our process models are PFSAs, and that a random yet finite choice of the set $\mathcal{G}$ is acceptable, giving us a metric almost surely with enough data.

III. Background

Definition 2 (PFSA)**.** A probabilistic finite-state automaton G is a quadruple $(Q, \Sigma, \delta, \widetilde{\pi})$, where Q is a finite set of states, Σ is a finite alphabet, $\delta : Q \times \Sigma \to Q$ called transition map, and $\widetilde{\pi} : Q \times \Sigma \to [0, 1]$ specifies observation probabilities, with $\forall q \in Q, \sum_{\sigma\in\Sigma} \widetilde{\pi}(q, \sigma) = 1$.

We use lower case Greeks (e.g. σ or τ) for symbols in Σ and lower case Latins (e.g. x or y) to denote sequence of symbols, with the empty sequence denoted by λ. The length of a sequence x is denoted by $|x|$. The set of sequences of length d is denoted by Σ^d.

The directed graph (not necessarily simple with possible loops and multi-edges) with vertices in Q and edges specified by δ is called the graph of the PFSA and, unless stated otherwise, assumed to be strongly connected [17].

Definition 3 (Observation and Transition Matrices)**.** Given a PFSA $(Q, \Sigma, \delta, \widetilde{\pi})$, the observation matrix $\widetilde{\Pi}_G$ is the $|Q| \times |\Sigma|$ matrix with the (q, σ)-entry given by $\widetilde{\pi}(q, \sigma)$, and the transition matrix Π_G is the $|Q| \times |Q|$ matrix with the (q, q')-entry, written as $\pi(q, q')$, given by

$$\pi(q, q') = \sum_{\sigma:\delta(q,\sigma)=q'} \widetilde{\pi}(q, \sigma). \tag{2}$$

Both Π_G and $\widetilde{\Pi}_G$ are stochastic, *i.e.* non-negative with rows of sum 1. Since the graph of a PFSA is strongly connected, there is a unique probability vector $\mathbf{p}_G$ that satisfies $\mathbf{p}_G^T \Pi_G = \mathbf{p}_G^T$ [18], and is called the stationary distribution of G.

Definition 4 (Γ-Expression)**.** δ and $\widetilde{\pi}$ may be encoded by a set of $|Q| \times |Q|$ matrices $\mathbf{\Gamma} = \{\Gamma_\sigma | \sigma \in \Sigma\}$, where

$$\Gamma_\sigma\Big|_{q,q'} = \begin{cases} \widetilde{\pi}(q,\sigma) & \text{if } \delta(q,\sigma) = q',\\ 0 & \text{if otherwise.} \end{cases} \tag{3}$$

We extend the definition to $\Sigma^\star$ by $\Gamma_x = \prod_{i=1}^n \Gamma_{\sigma_i}$ for $x = \sigma_1 \ldots \sigma_n$ with $\Gamma_\lambda = I$, where I is the identity matrix.

Definition 5 (Sequence-Induced Distributions)**.** For a PFSA $G = (Q, \Sigma, \delta, \widetilde{\pi})$, the distribution on Q induced by a sequence x is given by $\mathbf{p}_G^T(x) = \left\llbracket \mathbf{p}_G^T \Gamma_x \right\rrbracket$, where $\llbracket \mathbf{v} \rrbracket = \mathbf{v}/\left\|\mathbf{v}\right\|_1$.

Definition 6 (Stochastic process generated by PFSA)**.** Let $G = (Q, \Sigma, \delta, \widetilde{\pi})$ be a PFSA, the Σ-valued stochastic process $\{X_t\}_{t\in\Sigma}$ generated by G satisfies that X_1 follows the distribution $\mathbf{p}_G^T \widetilde{\Pi}_G$ and X_{t+1} follows the distribution $\mathbf{p}_G\left(X_1 \cdots X_t\right)^T \widetilde{\Pi}_G$ for $t \in \mathbb{N}$.

We denote the probability an PFSA G producing a sequence x by $p_G(x)$. We can verify that $p_G(x) = \left\|\mathbf{p}_G^T \Gamma_x\right\|_1$.

IV. Process KL Divergence Measures

Definition 7 (Entropy rate and Process KL divergence)**.** The entropy rate $\mathcal{H}(G)$ of a PFSA G is the entropy rate of the process generated by G [19]. Similarly, the KL divergence $\mathcal{D}_{\mathsf{KL}}\left(G \,\|\, G'\right)$ of a PFSA G' from the PFSA G is the KL divergence of the process generated by the G' from that of G [20]:

$$\mathcal{H}(G) = -\lim_{d\to\infty} \frac{1}{d} \sum_{x\in\Sigma^d} p_G(x) \log p_G(x) \tag{4}$$

$$\mathcal{D}_{\mathsf{KL}}\left(G \,\|\, G'\right) = \lim_{d\to\infty} \frac{1}{d} \sum_{x\in\Sigma^d} p_G(x) \log \frac{p_G(x)}{p_{G'}(x)}, \tag{5}$$

whenever the limits exist.

Lemma 1. *For any PFSA G, H, KL divergence satisfies:*

$$\mathcal{D}_{KL}\left(G \,\|\, H\right) \geqq 0 \tag{6}$$

$$\mathcal{D}_{KL}\left(G \,\|\, H\right) = 0 \textit{ iff } G = H \tag{7}$$

where we interpret equality of PFSA G, H as

$$\forall x \in \Sigma^\star, p_G(x) = p_H(x) \Rightarrow G = H \tag{8}$$

Proof: Follows from the standard argument for non-negativity of KL divergence for probability distributions [19]. ■

Definition 8 (Log-likelihood)**.** The log-likelihood [19] of a PFSA G generating $x \in \Sigma^d$ is given by

$$L(x, G) = -\frac{1}{d} \log p_G(x).$$

Theorem 1 (Convergence of Log-likelihood)**.** *Let G and H be two irreducible PFSA, and let $x \in \Sigma^d$ be a sequence generated by G. Then we have*

$$L(x, H) \to \mathcal{H}(G) + \mathcal{D}_{KL}\left(G \,\|\, H\right),$$

in probability as $d \to \infty$.

Proof: By chain rule

$$\sum_{x\in\Sigma^d} p_G(x) \log \frac{p_G(x)}{p_H(x)}$$

$$= \sum_{x\in\Sigma^{d-1}} \sum_{\sigma\in\Sigma} p_G(x) \mathbf{p}_G^T(x)\, \widetilde{\Pi}_G\Big|_\sigma \log \frac{p_G(x)\mathbf{p}_G(x)^T\, \widetilde{\Pi}_G\Big|_\sigma}{p_H(x)\mathbf{p}_H(x)^T\, \widetilde{\Pi}_H\Big|_\sigma}$$

$$= \sum_{x\in\Sigma^{d-1}} p_G(x) \log \frac{p_G(x)}{p_H(x)}$$

$$+ \underbrace{\sum_{x\in\Sigma^{d-1}} p_G(x) \sum_{\sigma\in\Sigma} \mathbf{p}_G(x)^T\, \widetilde{\Pi}_G\Big|_\sigma \log \frac{\mathbf{p}_G(x)^T\, \widetilde{\Pi}_G\Big|_\sigma}{\mathbf{p}_H(x)^T\, \widetilde{\Pi}_H\Big|_\sigma}}_{D_d}.$$

By induction, we have $\mathcal{D}_{\mathsf{KL}}\left(G \,\|\, H\right) = \lim_{d\to\infty} \frac{1}{d} \sum_{i=1}^{d} D_i$, and hence by Cesàro summation theorem [21], we have

$$\mathcal{D}_{\mathsf{KL}}\left(G \,\|\, H\right) = \lim_{d\to\infty} D_d.$$

Let $x = \sigma_1\sigma_2 ... \sigma_n$ be a sequence generated by G and $x^{[i-1]}$ be the truncation of x at the $(i-1)$-th symbols, we have

$$-\frac{1}{n} \sum_{i=1}^{n} \log \mathbf{p}_H\left(x^{[i-1]}\right)^T\, \widetilde{\Pi}_H\Big|_{\sigma_i}$$

$$= \underbrace{\frac{1}{n} \sum_{i=1}^{n} \log \frac{\mathbf{p}_G\left(x^{[i-1]}\right)^T\, \widetilde{\Pi}_G\Big|_{\sigma_i}}{\mathbf{p}_H\left(x^{[i-1]}\right)^T\, \widetilde{\Pi}_H\Big|_{\sigma_i}}}_{A_{x,n}}$$

$$\underbrace{-\frac{1}{n} \sum_{i=1}^{n} \log \mathbf{p}_G\left(x^{[i-1]}\right)^T\, \widetilde{\Pi}_G\Big|_{\sigma_i}}_{B_{x,n}}.$$

Because the process generated by G is ergodic, we have

$$\lim_{n\to\infty} A_{x,n} = \lim_{d\to\infty} D_d = \mathcal{D}_{\mathsf{KL}}\left(G \,\|\, H\right).$$

and $\lim_{n\to\infty} B_{x,n} = \mathcal{H}(G)$. ■

Next, we denote the log-likelihood of PFSA H generating a sequence x of length d which is actually generated by PFSA G as $L\left(x \overset{d}{\leftarrow} G, H\right)$. We show that the probability that sequences x, y generated by distinct processes cannot be distinguished by a random set of PFSA vanishes with enough data.

Theorem 2 (Approximate Metric)**.** *Let X and Y be two distinct PFSA in the sense of Eq.* (8)*, and x, y be of length at least d generated respectively by X, Y. If $\mathcal{G}$ is a randomly chosen set of k PFSA, then $Pr(\theta_{\mathcal{G}}(x, y) = 0) \to 0$, as $d, k \to \infty$.*

Proof: Because of Thm. 1, we start the proof by showing a fact about entropy and KL divergence: Let

$$D_{X,Y}(G) = \left|H(X) + \mathcal{D}_{\mathsf{kl}}\left(X \,\|\, G\right) - \left(H(Y) + \mathcal{D}_{\mathsf{kl}}\left(Y \,\|\, G\right)\right)\right|,$$

then either $D_{X,Y}(X) > 0$ or $D_{X,Y}(Y) > 0$. In fact, let us assume on the contrary that

$$D_{X,Y}(X) = \left|H(X) - \left(H(Y) + \mathcal{D}_{\mathsf{kl}}\left(Y \,\|\, X\right)\right)\right| = 0,$$
$$D_{X,Y}(Y) = \left|H(X) + \mathcal{D}_{\mathsf{kl}}\left(X \,\|\, Y\right) - \left(H(Y)\right)\right| = 0.$$

Since X and Y are not equivalent, we have

$$H(X) - H(Y) = \mathcal{D}_{\mathsf{kl}}\left(Y \,\|\, X\right) > 0, \text{ and}$$

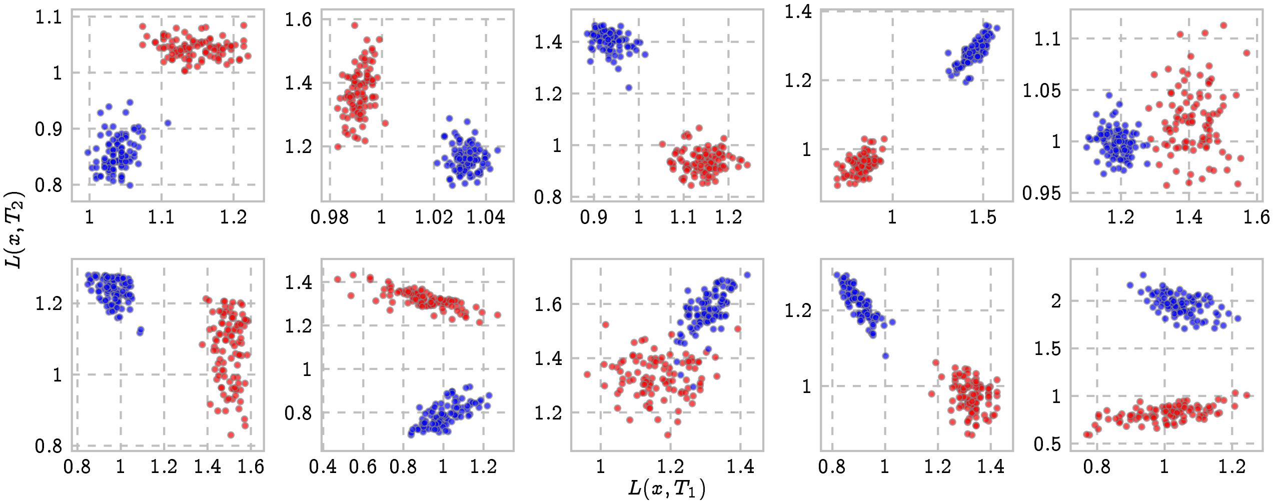

Fig. 1: For each of two PFSA G_1 and G_2 over binary alphabet, we generate 100 sequences of length 100. Then we generate random pairs of PFSA $\{T_1, T_2\}$ with binary alphabet and with size of state set ranging from 2 to 5. SLD between the sequences remain sufficiently positive for the examples, corroborating that a finite number of random PFSA discriminates between sample paths of distinct processes with high probability (Theorem 2).

$$H(X) - H(Y) = -\mathcal{D}_{\text{kl}}(X \,\|\, Y) < 0,$$

which is a contradiction.

Now, without loss of generality, let us assume that $D_{X,Y}(X) = c_{X,Y} > 0$ and let

$$\mathcal{A}_{X,Y} = \{G : D_{X,Y}(G) \geq c_{X,Y}/2\}.$$

Since $X \in \mathcal{A}_{X,Y}$, by continuity of entropy and KL divergence, we have $p_{X,Y} = Pr(\mathcal{A}_{X,Y}) > 0$.

Again by Thm. 1, we have

$$\left| L\left(x \overset{d}{\leftarrow} X, G\right) - L\left(y \overset{d}{\leftarrow} Y, G\right) \right| \to D_{X,Y}(G),$$

and hence, for $G \in \mathcal{A}_{X,Y}$,

$$Pr\left(L(x \overset{d}{\leftarrow} X, G) = L(y \overset{d}{\leftarrow} Y, G)\right) \to 0$$

as $d \to \infty$.

Let $\mathcal{G}$ be a set of k randomly chosen PFSA. From the analysis above, we see that as long as $\mathcal{G} \cap \mathcal{A}_{X,Y} \neq \emptyset$, the probability that $\mathcal{G}$ cannot distinquish sequences generate by X and Y vanishes as sequence length approaches infinity. However, since a randomly chosen set $\mathcal{G}$ of k PFSA satisfies

$$Pr(\mathcal{G} \cap \mathcal{A}_{X,Y} \neq \emptyset) = 1 - (1 - p_{X,Y})^k \to 1$$

as $k \to \infty$, we conclude the proof. ■

We illustrate the claim in Theorem 1 in Fig. 1 with a pair of randomly generated PFSA project sequences generated by two distinct PFSA. With two fixed PFSA G_1 and G_2 over the binary alphabet, we generate from each model 100 sequences, each of length 100. Then, we randomly choose two PFSAs $\mathcal{G} = \{T_1, T_2\}$, and compute SLD using Eq. (1), showing that the sequences remain well-separated.

Algorithm 1: PFSA Log-likelihood

```
Data: A PFSA $G = (Q, \Sigma, \delta, \widetilde{\pi})$ and a sequence $x$
      of length $n$.
Result: Log-likelihood of $G$ generating $x$
Get the stationary distribution $\mathbf{p}_G$ as the left
  eigenvector of $\Pi_G$ of eigenvalue 1;
Let $\mathbf{p}$ be the current distribution on states, and
  initialize it with $\mathbf{p}_G$;
Let $L$ be the log-likelihood of $G$ generating $x$ and
  initialize it with 0;
for each symbol $\sigma$ in $x$ do
    Get the current distribution on symbols
      $\phi = \mathbf{p}_G^T \widetilde{\Pi}_G$;
    Update $L = L - \log \phi(\sigma)$;
    Let $\mathbf{p}_{\text{new}}$ be the new distribution on states, and
      initialize all its entries with 0;
    for each state $q \in Q$ do
        Let the next state $q_{\text{new}} = \delta(q, \sigma)$;
        Let $\mathbf{p}_{\text{new}}(q_{\text{new}}) = \mathbf{p}_{\text{new}}(q_{\text{new}}) + \mathbf{p}(q)\widetilde{\pi}(q, \sigma)$;
    Update $\mathbf{p}$ with $\mathbf{p}_{\text{new}} / \|\mathbf{p}_{\text{new}}\|_1$;
Let $L = L/n$;
return $L$;
```

V. Implementation Considerations for Sequence Likelihood Divergence

The algorithm for evaluating the log-likelihood of a PFSA generating a given sequence is given in Alg. 1. It is immediate that the time complexity of log-likelihood evaluation is $O(d \times |Q|) + A$ with d is the input length and $|Q|$ is the number of states in the PFSA being considered, and A is the complexity of computing the stationary eigenvector in step 1. We note that the complexity for likelihood scoring of HMMs with the forward algorithm has time complexity $O\left(d \times |Q|^2\right)$, where Q is the number of the hidden states [22]. Nothwithstanding asymptotic time complexities, Alg. 1 is clearly significantly simpler to the dynamic programmimg involved in the forward algorithm of HMM likelihood scoring.

A. SLD with fixed base sets

For the performance and run time comparison on a synthetic dataset in Sec. VI and applications in Sec. VII, we use $\mathcal{G}$ composed the four simple PFSA in Fig. 2a-d. While better results may be obtained by random set of base models, using a fixed set yields sufficiently good performance when compared with the state of art. In contrast to using a fixed set of base models, we can also infer good base models in a classification problem, by selecting as the base models the class-specific PFSA inferred from the training set. This approach is further described in Sec. VI.

B. SLD with continuous data

Since PFSA model sequences on finite alphabet, continuous-valued input should first be quantized to discrete ones. The simplest approach of discretization is to choose $k-1$ cut-off points $p_1 < p_2 < \cdots < p_{k-1}$ and replace a value $< p_1$ by 0, in $[p_i, p_{i+1})$ by i, and $\geq p_{k-1}$ by k. We call the set of cut-off points a *partition*. In our implementation, we use the entropy maximization principle to obtain bins in which data points are evenly distributed. If there are clear trends in the data stream, we carry out partitioning after detrending.

VI. Performance Comparison with Baselines

A. Performance and Run Time Comparison

We compare dynamic time warping (DTW) [23], Smash [7], and SLD on a synthetic dataset of binaty time series samples. All three algorithms are implemented in C++ to eliminate slowdowns from compiled vs interpreted software as much as possible (for DTW we use a publicly available C++ implementation [24], [2] from the original authors). While all three algorithms admit parallelization, we use sequential implementations for a straightforward run time comparison.

Our synthetic dataset comprises 200 random binary classificiation problems, each consisting of two classes, with each class represented by 25 binary sequences of length 500. All sequences for a given class are sample paths from a randomly generated hidden Markov model with binary output.

For evaluating classification performance, we use *separtaion ratio* defined as follows. Let D be the matrix with $D_{i,j}$ being the distance between the i-th sequence and j-th sequence and l_i be the class label of the i-th sequence, we define the *mean inter-class distance* $s(D)$ and *mean intra-class distance* $d(D)$ by

$$s(D) = \frac{\sum_{i,j} \delta_{l_i l_j} D_{i,j}}{\sum_{i,j} \delta_{l_i l_j}}, \quad d(D) = \frac{\sum_{i,j} \left(1 - \delta_{l_i l_j}\right) D_{i,j}}{\sum_{i,j} \left(1 - \delta_{l_i l_j}\right)},$$

respectively, where $\delta_{ab} = 1$ if $a = b$ and 0 if otherwise. The separation ratio $r(D)$ is defined to be $d(D)/s(D)$. For DTW, we use window sizes $5, 10, 20, 30, 40, 50$, and 100, and for data smashing, we use the consider of 2 reruns because of its probabilistic nature. The average run time of SLD is $.042$ second. We note that (See Fig. 2e) that SLD achieves an average separation ratio that is comparable to DTW of window size 30 but with half the run time. Fig. 2f compares the DTW and SLD run times as the input length is increased, which shows significant advantage for the latter. Both data sets are included in the supplementary material (`synthetic.zip`).

B. SLD Example with Inferred Base Set

To compare the performance of SLD with DTW with the base set $\mathcal{G} = \{G_i : i = 1, \ldots, k\}$, where G_i is inferred from class i, we use the FordA data set downloadable from the UCR time series classification archive [25]. The FordA is a binary classification problem with continuous-valued sensor signals of length 500, which we detrend and partition (at levels $-.199174$ and $.198883$) to get trinary sequences. Since we have two classes, we infer two base models, *i.e.*, $|\mathcal{G}| = 2$, and the signals are mapped by Alg. 1 into points in $\mathbb{R}^2$, which are plotted in the first row of Fig. 2g. On the second row of Fig. 2g we show the 2D embedding using multidimensional scaling (MDS) of the distance matrix produced by DTW5. We normalize the points in both case and show the decision boundaries of the embeddings produced by four classification algorithms: k-nearest neighbors with $k = 3$, support vector machine, random forest, and multi-layer perceptron. Clearly SLD achieves better class separationcompared to DTW5. Note that inference of the two base models is carried our using the algorithm GenESeSS [26].

C. SLD-based Classifier Implementation

We have implemented a general time-series classifier (ClusteredHMMClassifier) applicable to both continuous valued and categorical data, using SLD as the core principle [27], which is publicly available at https://pypi.org/project/timesmash/. ClusteredHMMClassifier (Alg. 2) operates by first finding clusters within each training class based on SLD distance, followed by the inference of a PFSA model corresponding to each cluster in each class. Classification is finally carried out using these inferred

```
Algorithm 2: ClusteredHMMClassifier
Data:
  • Dataset $\left(X^{\text{train}}, X^{\text{test}}\right)$;
  • Labels $L^{\text{train}} = (l_1, \dots, l_n)$;
  • Optional quantization parameters;
  • A set of basis PFSA $\mathcal{G}$ for SLD;
  • A clustering algorithm clu;
  • A classification algorithm clf.
Result: Predicted labels for $X^{\text{test}}$.
Let quantization schemes
  $\mathcal{Q}_1, \dots, \mathcal{Q}_m = \text{Quantizer}\left(X^{\text{train}}, L^{\text{train}}\right)$;
Let $\mathcal{L}$ be the set of unique labels;
for $i = 1, \dots, m$ do
    /* Cluster and get subclass labels */
    Let $X_i^{\text{train}}, X_i^{\text{test}} = \mathcal{Q}_i\left(X^{\text{train}}, X^{\text{test}}\right)$;
    for each $l \in \mathcal{L}$ do
        Let $X_{i,l}^{\text{train}}$ be the subset of $X_i^{\text{train}}$ with label $l$;
        Let $D = \text{SLD}\left(X_{i,l}^{\text{train}}, \mathcal{G}\right)$;
        Let $\left\{X_{i,l,c}^{\text{train}} : c = 1, \dots, C\right\} = \mathbf{clu}(D)$;
        for $c = 1, \dots, C$ do
            Assign a new label $l_c$ to sequences in
              $X_{i,l,c}^{\text{train}}$;
    Let $L_i^{\text{new}}$ be the new labels;
    /* Infer PFSA and featurize using PFSA
       log-likelihood. */
    Let $\mathcal{L}$ be the set of unique labels;
    Let $\mathcal{G} = \emptyset$ be the set of class PFSA;
    for each $l$ in $\mathcal{L}$ do
        Let $X_l = \{x_i : y_i = l\}$;
        Add PFSA $G_l = \text{genESeSS}(X_l)$ to $\mathcal{G}$;
    Let $F^{\text{train}} = \text{Log-likelihood}\left(X^{\text{train}}, \mathcal{G}\right)$ (See
      Alg. 1);
    Let $F^{\text{test}} = \text{Log-likelihood}\left(X^{\text{test}}, \mathcal{G}\right)$;
Let $F^{\text{train}} = \left(F_1^{\text{train}}, \dots, F_m^{\text{train}}\right)$;
Let $F^{\text{test}} = \left(F_1^{\text{test}}, \dots, F_m^{\text{test}}\right)$;
/* Classification with the featurization. */
Train clf with $F^{\text{train}}$;
return prediction clf $\left(F^{\text{test}}\right)$;
```

models as the base set. Instead of simply summing up the log-likelihoods with respect to the base models as shown in Eq. (1), ClusteredHMMClassifier uses the set of log-likelihoods as a feature vector, and a standard classifier from those available in the scikit-learn [28] package, *e.g.*, RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier, or SVC.

We compare the error rates (1-accuracy) of the DTW baseline and ClusteredHMMClassifier on datasets from the UCR time series classification archive [29] and demonstrated result in Tab. I. We compare datasets with at least 50 time series per class for the comparison since the inference algorithm genESeSS needs a moderate sample size to work optimally. On 26 out of the 44 of the datasets, ClusteredHMMClassifier outperforms the DTW baseline . For PFSA inference we use the algorithm GenESeSS described in [26].

D. A Challenge Dataset Breaking DTW

To conclude this section we demonstrate a challenge dataset for DTW. The dataset has two classes with 25 binary sequences in each class. The sequence are generated by two PFSA that have the same transition map as the one in Fig. 2a but with different transition probabilities

$$\widetilde{\pi}_1(q_0) = (.6, .4)\,, \qquad \widetilde{\pi}_1(q_1) = (.4, .6)\,,$$
$$\widetilde{\pi}_2(q_0) = (.4, .6)\,, \qquad \widetilde{\pi}_2(q_1) = (.3, .7)\,.$$

After the sequences are generated, we flip 20% of the symbols independently at random, and get distance matrices of DTW with window sizes $0, 2, 5, 10$, and SLD with inferred base set. This dataset is designed to be especially challenging for DTW with bigger window size. We can see from Fig. 2h that the faint separation produced by DTW with window sizes 0 and 2 disappeared completely from window size 5 and 10, while the performance of SLD remains robust even with 20% added noise. The dataset is included in the supplementary material `challenge.zip`.

VII. Applications with Real-world Data

A. Dataset 1: Motor Movement Imagery Dataset

This dataset is an excerpt from PysioNet [30], containing 64-channel 160Hz EEG from participants performing specific tasks with images, namely: 1) TM: A target appears on either the left or the right side of the screen. The participant opens and closes the corresponding fist until the target disappears. Then the participant relaxes. 2) TI: The same as the first task, except the participant *imagines* opening and closing the corresponding fist but doesn't really move. During each recording, an object appears on the screen for 4 seconds and disappear for 4 seconds. A participant moves or imagines to move their fists for 4 seconds, then relaxes for 4 seconds, and the cycle repeats. Fig. 3a,b,e,f illustrates the raw EEG recordings, colored to distinguish the relaxation periods from the movement/imaginary movement periods.

For each participant, we select a dataset of 56 sequences for each task from two 2-minute EEG recordings, and order the sequences as demonstrated by Tab. II. We drop the first and last 4 seconds from each recording since they tend to be noisy. The heatmaps of distance matrices of two channels from two participants are shown in Fig. 3cd and Fig. 3g-h. From Fig. 3c-d. Note that participant S004 has different patterns in EEG between

TABLE I: Performance Comparison on UCR Datasets (Error Rates)

Dataset	Baseline	CH	Dataset	Baseline	CH
GunPointOldVersusYoung	0.035	0.006	ScreenType	0.589	0.576
FreezerRegularTrain	0.093	0.020	SemgHandSubjectCh2	0.200	0.196
StarLightCurves	0.093	0.021	ProximalPhalanxOutlineCorrect	0.192	0.189
Wafer	0.004	0.002	EthanolLevel	0.718	0.714
FordA	0.309	0.151	GunPointAgeSpan	0.035	0.035
SmallKitchenAppliances	0.328	0.229	ChlorineConcentration	0.350	0.352
ProximalPhalanxOutlineAgeGroup	0.195	0.141	PhalangesOutlinesCorrect	0.239	0.254
FordB	0.380	0.283	ECG5000	0.075	0.080
MixedShapesRegularTrain	0.091	0.068	UWaveGestureLibraryZ	0.322	0.347
WormsTwoClass	0.377	0.286	MiddlePhalanxOutlineCorrect	0.234	0.268
SemgHandMovementCh2	0.362	0.278	ElectricDevices	0.381	0.489
MiddlePhalanxOutlineAgeGroup	0.429	0.357	Yoga	0.156	0.203
DistalPhalanxTW	0.367	0.309	UWaveGestureLibraryY	0.301	0.401
DistalPhalanxOutlineAgeGroup	0.230	0.194	UWaveGestureLibraryX	0.227	0.310
SemgHandGenderCh2	0.155	0.132	Strawberry	0.054	0.076
DistalPhalanxOutlineCorrect	0.275	0.239	Crop	0.288	0.443
Ham	0.400	0.352	LargeKitchenAppliances	0.205	0.352
Computers	0.300	0.276	PowerCons	0.067	0.128
RefrigerationDevices	0.536	0.501	MelbournePedestrian	0.152	0.292
ProximalPhalanxTW	0.244	0.229	HandOutlines	0.119	0.243
Earthquakes	0.273	0.259	UWaveGestureLibraryAll	0.034	0.299
MiddlePhalanxTW	0.487	0.468	GunPointMaleVersusFemale	0.003	0.029

TABLE II: The composition of the TM and TI datasets

seq. No.	TM	TI
0 to 13	rest, rec. 1	rest, rec. 1
14 to 27	rest, rec. 2	rest, rec. 2
28 to 34	left hand, rec. 1	*imaginary* left hand, rec. 1
35 to 41	left hand, rec. 2	*imaginary* left hand, rec. 2
42 to 48	right hand, rec. 1	*imaginary* right hand, rec. 1
49 to 55	right hand, rec. 2	*imaginary* right hand, rec. 2

relaxation and (imaginary) movement, both from the wave form and from the heatmaps of the distance matrices. The relaxation to movement difference is persistent across recordings for this participant. in contrast, participant S001 does not have significant difference in the EEG between rest and (imaginary) movement sections (See Fig. 3gh). Instead, this participant seem to be in very different brain states for the two recordings.

B. Dataset 2: User Identification from Walking Activity

We consider a dataset of accelerometer measurements in x, y, z directions of human participants walking along a predefined trail in the wild (University of California, Irvine machine learning repository [31]). The challenge is to identify a participant from the inferred pattern of motion. We select 10 sequences from each of the 5 participants from the beginning of the walk. Each sequence is of 500 time steps long (each time step is about 0.03 second) and with 250 time steps overlapping between the two consecutive sequences.

We consider multiple quantization schemes, and demonstrate the distances for the best two schemes for x, y, and z directions in Fig. 3i, 3j, and 3k, respectively. Note that while the distance in the x direction between the first two participants (sequences 0-9, and sequences 10-19) is small, their difference in the z direction is significant.

VIII. Conclusion

In this paper, we introduce sequence likelihood divergence as an efficiently computable measure of deviation between time series data. We compare SLD against state of the art algorithms demonstrating at par or better performance, and significant improvement in runtime. The recent explosion of data driven applications call for faster, better tools to compare and contrast data, and we hope that SLD offers a new addition to the toolbox of the modern data science revolution.

References

[1] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, vol. 44, no. 3, pp. 678–693, 2011.

[2] A. M. Q. Z. J. Z. E. K. G. B. Thanawin Rakthanmanon, Bilson Campana and B. Westover, "Ucr suite for time series subsequence search," https://www.cs.ucr.edu/~eamonn/UCRsuite.html, (Accessed on 01/20/2021).

[3] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery.* ACM, 2003, pp. 2–11.

[4] C. S. Möller-Levet, F. Klawonn, K.-H. Cho, and O. Wolkenhauer, "Fuzzy clustering of short time-series and unevenly distributed sampling points," in *International Symposium on Intelligent Data Analysis.* Springer, 2003, pp. 330–340.

[5] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.

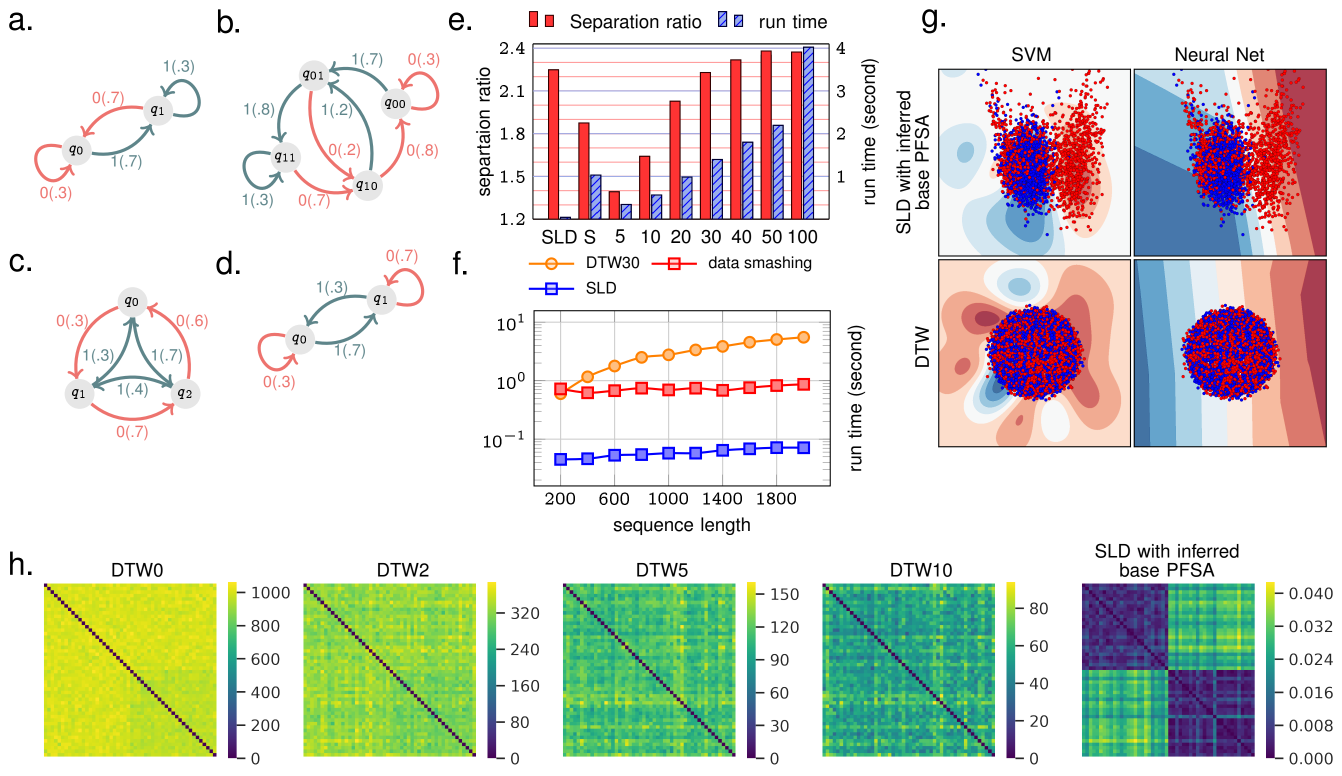

Fig. 2: Panel a-d: Four basis PFSA used for the algorithm comparison in Sec. VI and applications in Sec. VII. An edge connecting state q to q' is labeled as $\sigma\left(\widetilde{\pi}(q,\sigma)\right)$ if $\delta(q,\sigma) = q'$ (See Defn. 2). Panel e: performance and run time comparisons of SLD, data smashing [7], and DTW on a synthetic symbolic dataset. We denote data smashing by S and DTW by their window size. The average run time of of SLD is $.042$ second. Panel f: run time v.s. sequence length comparison between DTW30 and SLD. Panel g: $2D$ embeddings produced by Alg. 1 and DTW5 on the FordA dataset with decision boundaries of SVM and neural network. Panel h: Distance matrices produced by DTW with window sizes $0, 2, 5, 10$ and SLD with inferred basis PFSA on a dataset desigend especially challenging for DTW.

[6] L. Chen, M. T. Özsu, and V. Oria, "Robust and fast similarity search for moving object trajectories," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, 2005, pp. 491–502.

[7] I. Chattopadhyay and H. Lipson, "Data smashing: uncovering lurking order in data," *Journal of The Royal Society Interface*, vol. 11, no. 101, p. 20140826, 2014.

[8] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[9] A. Rényi, "On the foundations of information theory," *Revue de l'Institut International de Statistique*, pp. 1–14, 1965.

[10] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE mobile computing and communications review*, vol. 5, no. 1, pp. 3–55, 2001.

[11] J. P. Crutchfield, "The calculi of emergence: computation, dynamics and induction," *Physica D: Nonlinear Phenomena*, vol. 75, no. 1-3, pp. 11–54, 1994.

[12] P. Dupont, F. Denis, and Y. Esposito, "Links between probabilistic automata and hidden markov models: probability distributions, learning models and induction algorithms," *Pattern recognition*, vol. 38, no. 9, pp. 1349–1371, 2005.

[13] I. Chattopadhyay, "Causality networks," *arXiv preprint arXiv:1406.6651*, 2014.

[14] W. Ching and M. Ng, *Markov Chains: Models, Algorithms and Applications*, ser. International Series in Operations Research & Management Science. Springer US, 2006. [Online]. Available: https://books.google.com/books?id=IjMLAFZKBzYC

[15] C. W. Helstrom, *Probability and stochastic processes for engineers*. Macmillan Coll Division, 1991.

[16] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, and L. E. Meester, *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer Science & Business Media, 2005.

[17] J. Bondy and U. Murty, "Graph theory (2008)," *Grad. Texts in Math*, 2008.

[18] M. Vidyasagar, *Hidden markov processes: Theory and applications to biology*. Princeton University Press, 2014, vol. 44.

[19] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[20] M. Vidyasagar, "Bounds on the kullback-leibler divergence rate between hidden markov models," in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 6160–6165.

[21] G. Hardy, "Divergent series, with a preface by je littlewood and a note by ls bosanquet, reprint of the revised (1963) edition," *Éditions Jacques Gabay, Sceaux*, 1992.

[22] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[23] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[24] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proceedings of the 18th ACM SIGKDD international conference*

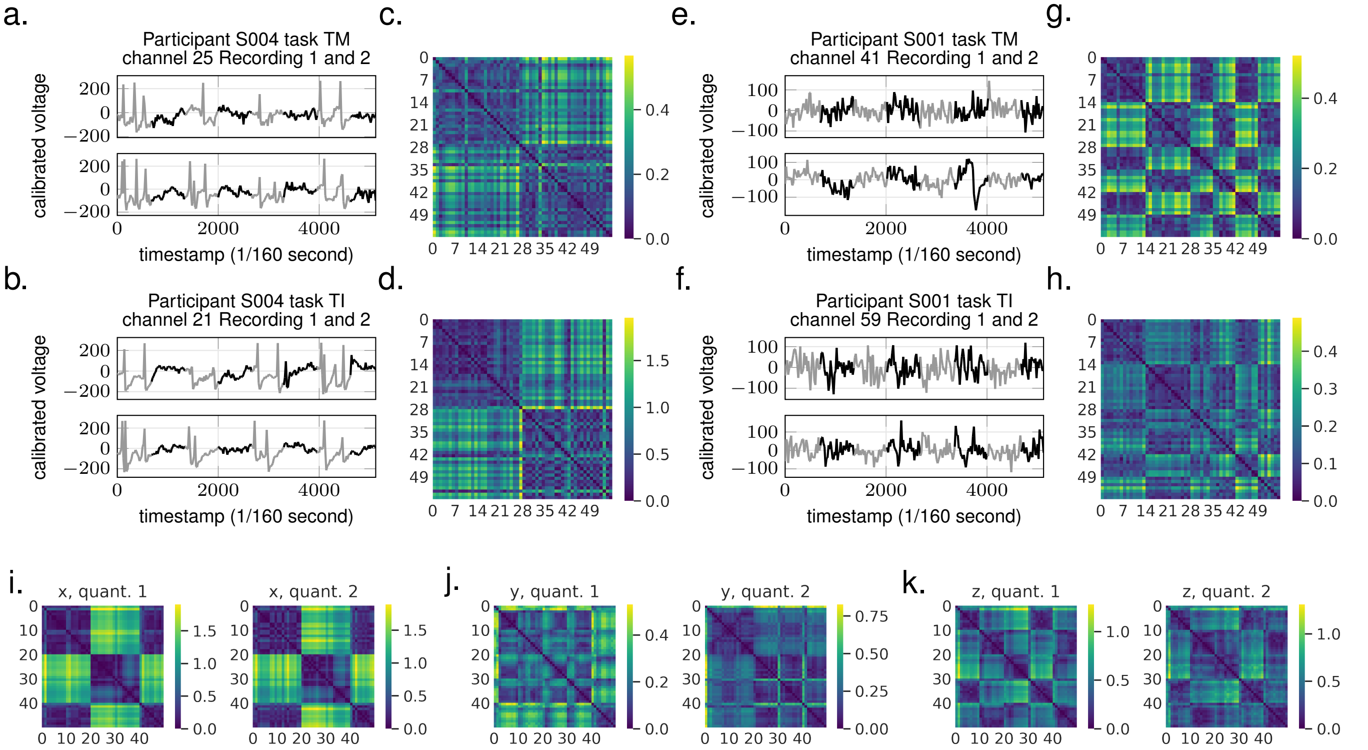

Fig. 3: Panel a-h: Multi-channel EEG recordings and distance matrices for two participants from the Motor Movement Imagery Datasets (see Sec. VII-A). For each participant, the dataset contains two EEG recordings of two tasks, alternating rest and movement (TM) and alternating rest and *imaginary* movement (TI). Rest sections are colored gray while (imaginary) movement sections, black. From the arrangement of the sequences (see Tab. II), we can see that SLD clearly distinguishes rest from (imaginary) movements for participant S004 while distinguishes two recordings for participant S001. Panel i-j: Distance matrices of accelerometer measurements in the x, y, and z directions from 5 users in the User Identification from Walking Activity Dataset (see Sec. VII-B). For each direction, we collect 10 sequences of 500 time steps from 5 users and show distance matrices resulted from two quantizations (see Sec. V). From the heatmap, we see that measurements from different directions can be combined in user recognition. For example, although SLD doesn't separate user 1 and 2 well in the x-direction, it picks enough separation in the z-direction.

on Knowledge discovery and data mining, 2012, pp. 262–270.

[25]

[26] I. Chattopadhyay and H. Lipson, "Abductive learning of quantized stochastic processes with probabilistic finite automata," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, p. 20110543, 2013.

[27] V. Rotaru, "Timesmash," https://pypi.org/project/timesmash/#description, 2020.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion,

[30] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals,"

O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The ucr time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, 2019.
Circulation, vol. 101, no. 23, pp. e215–e220, 2000.

[31] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml