Online Clustering of Processes

Azadeh Khaleghi Azadeh.Khaleghi@inria.fr
Daniil Ryabko Daniil.Ryabko@inria.fr
Jérémie Mary Jeremie.Mary@inria.fr
Philippe Preux Philippe.Preux@inria.fr

SequeL-INRIA/LIFL-CNRS, Université de Lille, France

Abstract

The problem of online clustering is considered in the case where each data point is a sequence generated by a stationary ergodic process. Data arrive in an online fashion so that the sample received at every time-step is either a continuation of some previously received sequence or a new sequence. The dependence between the sequences can be arbitrary. No parametric or independence assumptions are made; the only assumption is that the marginal distribution of each sequence is stationary and ergodic. A novel, computationally efficient algorithm is proposed and is shown to be asymptotically consistent (under a natural notion of consistency). The performance of the proposed algorithm is evaluated on simulated data, as well as on real datasets (motion classification).

1 Introduction

The focus of this work is on online clustering of time-series data. This involves the clustering of a growing body of sequences of data, when each sequence is generated by some discrete-time stochastic process. The clustering is done "online," meaning that at every time-step we receive some new samples that either form a new sequence or are a continuation of some previously observed sequence. Therefore at each time step t a total of $N(t)$ sequences $\mathbf{x}_1, \ldots, \mathbf{x}_{N(t)}$ are to be clustered, where each sequence $\mathbf{x}_i$ is of length $n_i(t) \in \mathbb{N}$ for $i = 1..N(t)$. The total number of observed sequences $N(t)$ as well as the individual sequence-lengths $n_i(t)$ grow with time.

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX.

We assume that each sequence is generated by one of k *unknown* stationary ergodic distributions, for some prescribed k. This is a very general assumption, subsuming most of the modeling and independence assumptions traditionally used in time-series clustering. We require (following the approach of [11]) that those and only those sequences generated by the same distribution be placed into the same cluster. A clustering function that, for each fixed portion of sequences, achieves this objective in asymptotic is *asymptotically consistent*.

Motivation. The analysis of time-series data in general, and the particular problem of time-series clustering is motivated by many research problems from a variety of disciplines, such as marketing and finance, biological and medical research, video/audio analysis, etc. In many applications data arrive dynamically, with both new sources being added and previously available sources generating more data. It is important for a clustering algorithm to cluster recently observed data points as soon as possible, without changing its decision about those that have already been clustered correctly.

The main challenge in online time-series clustering can be identified with "bad" sequences: recently-observed sequences for which sufficient information has not yet been collected to allow for their distinction based on their generating distribution. Thus (as will be thoroughly discussed in the paper), using a batch algorithm in this setting results in not only misclustering such "bad" sequences, but also in clustering incorrectly those for which sufficient data is already available. That is, such "bad" sequences could render the entire batch clustering useless, leading the algorithm to miscluster even the "good" sequences. Since new (bad) sequences may arrive in a data-dependent fashion, any batch algorithm will fail in this scenario.

Results. We present a novel non-parametric online clustering algorithm for time-series data, and evaluate it both theoretically and empirically.

Theoretically, we demonstrate that our algorithm is consistent provided solely that the marginal distribution of each sequence is stationary ergodic; there can be any (adversarial) dependence between the samples. We further show that our algorithm is computationally efficient: it is at most quadratic in each argument.

To test the empirical performance of the proposed algorithm, we first optimized and implemented the offline method of [11], and evaluated the studied methods on both synthetic and real data. In the batch setting, the error-rates of both methods go to zero with sequence-length. In the online setting with new samples arriving at every time-step, the error-rate of the offline algorithm remains consistently high, whereas that of the online algorithm converges to zero. This demonstrates that unlike the offline algorithm, the online algorithm is robust to "bad" sequences.

To reflect the generality of the suggested framework in the experimental setup, we had our synthetic data generated by processes that, while being stationary ergodic, do not belong to any "simpler" class of processes, and in particular cannot be modeled as a hidden Markov process with a countable set of states.

To demonstrate the applicability of the studied framework to *real data,* we chose the problem of clustering motion-capture sequences of human locomotion. This application area has also been studied in recent works [9] and [6], which (to the best of our knowledge) constitute the state-of-the-art performance on the datasets they consider, and against which we compare the performance of our methods. We obtained *consistently better* performance on the datasets involving motion that can be considered ergodic (walking, running), and competitive performance on those involving non-ergodic motions (single jumps).

Related work. Even though clustering is a classical problem in machine learning and statistics, the existing literature on non-parametric time-series clustering approaches as well as on the theoretical analysis of their consistency results is rather scarce. This is partially due to the fact that in most cases even defining what it means for a clustering result to be *correct*, can be notoriously difficult (if not impossible) [7, 17]. For this reason the most common approaches to time-series clustering is to study specific algorithms (like k-means) or specific models, for example, assuming that the data is generated by a specific family of hidden Markov chains, or using some other parametric families [2, 3, 8, 16, 18].

Recently, [11] proposed a natural notion of consistency along with a methodology to obtain consistent algorithms for the particular case of the (offline) clustering of stationary ergodic time-series. The proposed approach, which we extend to the online setting, is based on estimating the so-called distributional distance between process distributions. This approach has been previously used in [14, 13] to solve several other statistical problems about time series.

The main advantage of this framework is that it allows for the development of simple non-parametric algorithms that are consistent and are not limited by any modeling assumptions on the data.

Outline The rest of this paper is organized as follows. In Section 2 we introduce some notation and definitions, and formalize the online clustering problem considered. We present our theoretical and experimental results in Sections 3 and 4 respectively. Finally, we provide some concluding remarks in Section 5.

2 Preliminaries

Notation. Let $\mathcal{A}$ be an alphabet. In this work we consider the case where $\mathcal{A} = \mathbb{R}$; extensions to more general spaces are straightforward. Consider the Borel σ-algebra $\mathcal{B}$ generated by $\{B \times \mathcal{A}^\infty : B \in B^{m,l}, m, l \in \mathbb{N}\}$ on $\mathcal{A}^\infty$ where the sets $B^{m,l}, m, l \in \mathbb{N}$ are obtained via the partitioning of $\mathcal{A}^m$ into cubes of dimension m and volume 2^{-ml} (starting at the origin). Let also $B^m := \cup_{l\in\mathbb{N}} B^{m,l}$. Processes are probability measures on the space $(\mathcal{A}^\infty, \mathcal{B})$. Similarly, we can define distributions on the space $((\mathcal{A}^\infty)^2, \mathcal{B}_2)$ of infinite matrices where the Borel sigma algebra $\mathcal{B}_2$ is generated by cylinders $(B_1 \times \mathcal{A}^\infty) \times \cdots \times (B_r \times \mathcal{A}^\infty) \times (\mathcal{A}^\infty)^2$, $B_i \in B^m, i = 1..r$ where $m, r \in \mathbb{N}$. For a sequence $X_1, \ldots, X_n$ we use the abbreviation $X_{1..n}$. For $\mathbf{x} = X_{1..n} \in \mathcal{A}^n$ and $B \in B^m$ let $\nu(\mathbf{x}, B)$ denote the *frequency* with which $\mathbf{x}$ falls in the set B, i.e. $\nu(\mathbf{x}, B) := \frac{\mathbb{I}\{n\geq m\}}{n-m+1} \sum_{i=1}^{n-m+1} \mathbb{I}\{X_{i..i+m-1} \in B\}$.

A process ρ is *stationary* if for a sequence $\mathbf{x} = X_{1..n}$ and any $i, j \in 1..n$ and $B \in B^m$, $m \in \mathbb{N}$, we have $\rho(X_{1..j} \in B) = \rho(X_{i..i+j-1} \in B)$. A stationary process ρ is called *(stationary) ergodic* if $\rho(\lim_{n\to\infty} \nu(X_{1..n}, B) = \rho(B)) = 1$ for all $B \in \mathcal{B}$.

Online Clustering Protocol. We consider infinitely many one-way infinite sequences, each of which is generated by one out of k *unknown* stationary ergodic distributions. At time-step 1 initial segments of some of the first sequences are available to the learner. At each subsequent time step, new data is revealed, either as a subsequent segment of a previously observed sequence, or as a new sequence. Although it is known that the *eventual* number of different time-series distributions producing the sequences is k, the number of observed distributions at each individual time-step is unknown.

More formally, consider the matrix of random variables

$$\mathbf{X} := \begin{bmatrix} X_1^1 & X_2^1 & X_3^1 & \cdots \\ X_1^2 & X_2^2 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix} \in (\mathcal{A}^\infty)^2, \qquad (1)$$

(with an infinite number of rows and columns), generated by some (unknown) probability distribution P on $((A^\infty)^2, \mathcal{B}_2)$. We assume that the marginal distribution of P on each row of $\mathbf{X}$ is one of k *unknown stationary ergodic* processes $\{\rho_1, \rho_2, \cdots, \rho_k\}$. Here k is assumed known. Besides the assumption that ρ_i, $i = 1..k$ are stationary ergodic, we do not make any further assumptions on the distribution P that generates $\mathbf{X}$. This means that the samples in $\mathbf{X}$ are allowed to be dependent, and the dependence can be arbitrary; one can even think of the dependence between samples as "adversarial". For convenience of notation we assume that the distributions $\rho_i, i = 1..k$ are ordered in the order of appearance of their first samples in $\mathbf{X}$.

At every time step $t \in \mathbb{N}$, a part $S(t)$ of $\mathbf{X}$ is observed corresponding to the first $N(t) \in \mathbb{N}$ rows of $\mathbf{X}$, each of length $n_i(t), i \in 1..N(t)$, i.e. $S(t) = \{\mathbf{x}_1^t, \cdots \mathbf{x}_{N(t)}^t\}$ where $\mathbf{x}_i^t := X_{1..n_i(t)}^i$. Once revealed, data is never taken away: $N(t)$ is non-decreasing in t, as are $n_i(t)$ for each $i \in \mathbb{N}$. In our theoretical results we assume that the number of samples, as well as the individual sample-lengths tend to infinity with time; that is, $\lim_{t\to\infty} n_i(t) \to \infty$ for all $i \in 1..N(t)$.

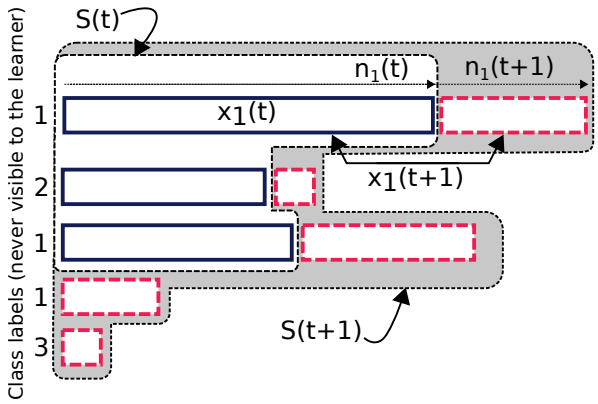

Figure 1: Online Protocol: solid rectangles correspond to sequences observed at time t, dashed rectangles correspond to segments arrived at time $t+1$.

An *online clustering function* is a mapping $f(S(t), k) \mapsto C(t) = \{C_1(t), \cdots, C_k(t)\}$ that for each $t \in \mathbb{N}$ gives a *partitioning* of the index-set $1..N(t)$ into k disjoint subsets $C_i(t)$, $i = 1..k$.

Of the many ways a set of k disjoint subsets of $S(t)$ may be produced, the most natural partitioning in this scenario is to *put into the same cluster those and only those sequences that were generated by the same distribution.* With this observation, following the approach, set in [11] we define the ground-truth clustering of $\mathbf{X}$ as follows:

Definition 2.1 (Ground-Truth Clustering) *Define the ground-truth clustering of $\mathbf{X}$ as the partitioning $\mathcal{I} = \{\mathcal{I}_1, \cdots, \mathcal{I}_k\}$ of $\mathbb{N}$ such that $j \in \mathcal{I}_i \Leftrightarrow \mathbf{x}_j \sim \rho_i$ (where $\mathbf{x}_j$ denotes the j^{th} row of $\mathbf{X}$).*

A clustering function is *(asymptotically) consistent* if, with probability 1, for each $N \in \mathbb{N}$ from some time on the first N sequences are clustered correctly (with respect to the ground-truth clustering). More formally we have the following definition:

Definition 2.2 (Consistency) *A clustering function is said to be (strongly) asymptotically consistent, if with probability 1 for every $N \in \mathbb{N}$ there exists some time T such that for all $t \geq T$ we have,*

$$\{C_i(t) \cap 1..N : i = 1..k\} = \{\mathcal{I}_i \cap 1..N : i = 1..k\}.$$

For every pair of processes ρ_1 and ρ_2 the *distributional distance* between them is defined as follows

$$d(\rho_1, \rho_2) := \sum_{m,l=1}^{\infty} w_{m,l} \sum_{B \in B^{m,l}} |\rho_1(B) - \rho_2(B)|,$$

where $w_{m,l} := w_m w_l$ and $w_i = 2^{-i}$, $i \in \mathbb{N}$. [1] It is easy to see that $d(\cdot,\cdot)$ is a metric. For more on the distributional distance and its properties see [5].

The algorithms presented in this work are based on *empirical estimates* of this distance:

$$\hat{d}(\mathbf{x}_1, \mathbf{x}_2) := \sum_{m,l=1}^{\infty} w_{m,l} \sum_{B \in B^{m,l}} |\nu(\mathbf{x}_1, B) - \nu(\mathbf{x}_2, B)|,$$

where $\mathbf{x}_i = X_{1..n_i}^i \in \mathcal{A}^{n_i}$, $n_i, \in \mathbb{N}$, $i = 1, 2$. Similarly, the empirical estimate of the distance between a sequence $\mathbf{x} \in \mathcal{A}^n, n \in \mathbb{N}$ and a process ρ is defined as:

$$\hat{d}(\mathbf{x}, \rho) := \sum_{m,l=1}^{\infty} w_{m,l} \sum_{B \in B^{m,l}} |\nu(\mathbf{x}, B) - \rho(B)|.$$

As shown in [11] $\hat{d}(\cdot,\cdot)$ is asymptotically consistent: for every pair of sequences $\mathbf{x}_1 = X_{1..n_1}^1$ and $\mathbf{x}_2 = X_{1..n_2}^2$, each generated by a stationary ergodic distribution ρ_i, $i = 1, 2$ with a stationary ergodic joint distribution ρ, we have

$$\lim_{n_1,n_2\to\infty} \hat{d}(X_{1..n_1}^1, X_{1..n_2}^2) = d(\rho_1, \rho_2), \ \rho - \text{a.s., and} \quad (2)$$

$$\lim_{n_i\to\infty} \hat{d}(X_{1..n_i}^i, \rho_j) = d(\rho_i, \rho_j), \ i, j = 1, 2, \ \rho - \text{a.s.} \quad (3)$$

[1] Any summable sequence of positive weights also works.

Moreover a more general estimate of the distributional distance may be obtained as

$$\check{d}(\mathbf{x}_1,\mathbf{x}_2) := \sum_{m=1}^{m_n}\sum_{l=1}^{l_n} w_{m,l} \sum_{B\in B^{m,l}} |\nu(\mathbf{x},B)-\rho(B)|, \quad (4)$$

where m_n and l_n are any sequences of integers that go to infinity with n. As shown in [11] the consistency results for $\hat{d}(\cdot,\cdot)$ (Equations 2 and 3) equally hold for $\check{d}(\cdot,\cdot)$ as long as $m_n, l_n \to \infty$ with $n \to \infty$.

Algorithm 1 Offline Clustering Subroutine [11]

INPUT: sequences $\{\mathbf{x}_1,\cdots,\mathbf{x}_N\}$, # of clusters k
* *Initialize k-farthest points as cluster-centers:*
$C_1 \leftarrow \{1\}$
$C_j \leftarrow \{\operatorname{argmax}_{i=1..N} \min_{i'\in\bigcup_{j'=1}^{j-1} C_{j'}} \hat{d}(\mathbf{x}_i,\mathbf{x}_{i'})\},\ j=2..k$
* *Assign the remaining points to closest centers:*
for $i=1..N$ **do**
  $j \leftarrow \operatorname{argmin}_{i'\in\bigcup_{j'=1}^{k} C_{j'}} \hat{d}(\mathbf{x}_i,\mathbf{x}_{i'})$
  $C_j \leftarrow C_j \cup \{i\}$
end for
* *Search for the lowest index in each cluster:*
for $i=1..k$ **do**
  $r_i \leftarrow \min_{r\in C_i} r$
end for
* *Output the lowest index in each cluster:*
OUTPUT: $(r_1, r_2, \cdots, r_k)$

Offline Clustering Algorithm. Alg 1 is a consistent batch method of [11], slightly modified to serve as a subroutine in our online algorithm. It is essentially one iteration of k-means with farthest-point initialization, using $\hat{d}(\cdot,\cdot)$ as the distance between sequences.

3 Theoretical Results

We present via Alg 2 an online clustering procedure which, as the following theorem shows, is consistent under the most general assumptions.

Theorem 3.1 ***i.*** *Alg 2 is asymptotically consistent, provided that the marginal distribution of each sequence is stationary ergodic, and that the correct number of clusters k is supplied to the algorithm. The same statement holds if $\check{d}(\cdot,\cdot)$ is used instead of $\hat{d}(\cdot,\cdot)$ with any pairs of sequences m_n, l_n s.t. $m_n, l_n \to \infty$.*
ii. *The per symbol resource (space and time) complexity of Alg 2 at time-step t is of order*

$$\mathcal{O}(kN(t)^2 + N(t)n_{\max}\log s_{\min}^{-1}),$$

where $s_{\min} := \min_{\substack{u,v\in 1..N(t)\\ i=1..n_u, j=1..n_v, X_i^u\neq X_j^v}} |X_i^u - X_j^v|$ *and* $n_{\max} := \max_{i=1..N(t)} n_i(t)$. *If $\check{d}(\cdot,\cdot)$ is used the complexity becomes $\mathcal{O}(\{kN(t)^2 + N(t)m_{n_{\max}}\log s_{\min}^{-1})$*

Algorithm 2 Online Clustering

INPUT: # of clusters k
for $t=1..\infty$ **do**
  Obtain new sequences $S(t)=\{\mathbf{x}_1^t,\cdots,\mathbf{x}_{N(t)}^t\}$
  Initialize the normalization factor: $\eta \leftarrow 0$
  for $j=k..N(t)$ **do**
  * *Use Alg 1 to select k indices from $1..j$ to index cluster-representatives within $\mathbf{x}_1^t..\mathbf{x}_j^t$ & store the received k-tuple as the j^{th} column of the representative matrix R. (so R_{ij} indexes the representative of cluster i selected at iteration j):*
    $R_{1..k,j} \leftarrow \text{Alg1}(\{\mathbf{x}_1^t,\cdots,\mathbf{x}_j^t\},k)$
  * *Calculate the min inter-cluster distance γ_j:*
    $\gamma_j \leftarrow \min_{a\neq b\in 1..k} \hat{d}(\mathbf{x}^t_{R_{a,j}}, \mathbf{x}^t_{R_{b,j}})$
  * *Calculate the weight α_j (corresponding to the cluster-representatives $\mathbf{x}^t_{R_{1j}}..\mathbf{x}^t_{R_{kj}}$):*
    $w_j \leftarrow j^{-2}$;$\alpha_j \leftarrow w_j\gamma_j$
  * *Update the normalization factor:*
    $\eta \leftarrow \eta + \alpha_j$
  end for
  * *Form clusters: For every sequence $\mathbf{x}_l^t$, $l\in 1..N(t)$ find some $c\in 1..k$ that minimizes the weighted sum of the distances between $\mathbf{x}_l^t$ & the sequences $\mathbf{x}^t_{R_{ij}}$, $i=1..k, j=1..N(t)$; let l join $C_c(t)$.*
  for $l=1..N(t)$ **do**
    $c \leftarrow \operatorname{argmin}_{i\in 1..k} \frac{1}{\eta}\sum_{j=1}^{N} \alpha_j \hat{d}(\mathbf{x}_l^t, \mathbf{x}^t_{R_{i,j}})$
    $C_c(t) \leftarrow C_c(t) \cup \{l\}$
  end for
  OUTPUT: $\{C_1(t),\cdots,C_k(t)\}$
end for

The proof is deferred to Section 3.1. Here we informally explain how and why the algorithm works.

The proposed algorithm is based on combining several clusterings, each obtained by running the offline algorithm on different portions of data; more specifically, the batch algorithm is run on each subset of $\{\mathbf{x}_1^t,\cdots,\mathbf{x}_j^t\}$ $j=k..N(t)$ of the sequences $S(t)$ observed at time-step t. These clusterings are combined with weights that depend on j and on the performance of each clustering as reflected by the minimum inter-cluster distance.

To see the intuition behind this approach, first note that $\hat{d}(\cdot,\cdot)$ is consistent, meaning that the empirical distributional distance between a given pair of sequences converges to the distributional distance between their generating processes. As shown in [11] this is the key reason why, Alg 1 is asymptotically consistent in a batch setting. Knowing that the batch algorithm is consistent, it can be tempting to view the solution to the online clustering problem as the direct

application of this algorithm to the batch of sequences $S(t)$ observed at every time-step t. However, with new sequences arriving at every time step, there are always some sequences for which the distance estimate is far from being correct. This leads to producing incorrect clusterings of (potentially) all sequences at every time step. An alternative solution would be to take some fixed portion of the samples, run the batch algorithm on it, then simply assign every remaining sequence to the nearest cluster. This procedure would be asymptotically consistent, if we were guaranteed that the selected portion of the sequences contains at least one sequence sampled from each and every one of the k distributions. However, we cannot know this. In other words, there is no way to select a portion of data that would have sequences long enough to result in a correct clustering and at the same time contain a representative of each distribution.

A key observation we make is that any k-partitioning of a batch containing sequences generated by *at most* $k-1$ processes results in a minimum inter-cluster distance γ that, as follows from the asymptotic consistency of $\hat{d}(\cdot,\cdot)$, converges to 0. On the other hand, if the observed batch of sequences $\{\mathbf{x}_1,\cdots,\mathbf{x}_N\}$ contains sequences generated by all k processes, then the estimated minimal inter-cluster distance converges to a non-zero value: to the minimal distance between distributions generating the data.

Therefore, given the $N = N(t)$ sequences in $S(t)$ observed at time t, we use Alg 1 to generate N clusterings, each based on the first j sequences $\mathbf{x}_1,\cdots,\mathbf{x}_j$ for $j = k...N$. These clusterings are then combined with two sets of weights: **1.** γ_j to penalize for small inter-cluster distance, annulling those clusterings produced based on sets of sequences generated by less than k distributions. **2.** w_j to give precedence to chronologically earlier clusterings, protecting the clustering decisions from the presence of the (potentially "bad") newly formed sequences, whose corresponding distance estimates may still be far from accurate.

This approach may be reminiscent of prediction with expert advice [4], where experts are combined based on their past performance. A key difference is that we cannot measure the performance of each clustering directly.

3.1 Proofs

Proposition 3.1 *Let the sequence* $\mathbf{x}_1 = X^1_{1..n_1}$ *be obtained when* $\mathbf{x}'_1 = X^1_{1..n_1-1}$ *is extended by a single element. Given* $\hat{d}(\mathbf{x}'_1,\mathbf{x}_2)$ *the computational complexity of obtaining* $\hat{d}(\mathbf{x}_1,\mathbf{x}_2)$ *is of order* $\mathcal{O}(\max\{n_1,n_2\}\log s^{-1})$, *where* $s := \min_{\substack{X^u_i\neq X^v_j \\ u,v\in1,2,i=1..n_u,j=1..n_v}} |X^u_i - X^v_j|$.

Proof Define $T^{m,l} \triangleq \sum_{B\in B^{m,l}} |\nu(\mathbf{x}_1,B) - \nu(\mathbf{x}_2,B)|$. For all $l \geq \log s^{-1}$ the cubes $B \in B^{m,l}$ contain *at most* a single m-tuple of the form $X^u_{i..i+m}, i = 1..n_u, u = 1,2$. Thus,

$$\sum_{l\in\mathbb{N}} w_{m,l}T^{m,l} = w_m(1 - \sum_{l=1}^{\log s^{-1}} w_l)T^{m,\log s^{-1}} + \sum_{l=1}^{\log s^{-1}} w_{m,l}T^{m,l}.$$

Moreover, for fixed $m,l \in \mathbb{N}$ every cube in $B^{m,l}$ can be partitioned into 2^m cubes in $B^{m,l+1}$ so that, $\nu(\mathbf{x},B)$ can be obtained recursively for all $B \in B^{m,l}$, i.e.

$$\nu(\mathbf{x},B) = \sum_{B'\in B^{m,l+1}:B'\subset B} \nu(\mathbf{x},B'). \tag{5}$$

The frequency values $\nu(\mathbf{x}_i,B), i = 1,2$ may be stored in a 2^m-ary tree of depth $\log s^{-1}$, whose nodes at each level $l \in 1..\log s^{-1}$ hold a tuple $v_{j,l} = (\nu(\mathbf{x}_1,B_j),\nu(\mathbf{x}_2,B_j)), j = 1..2^{ml}$ where $B_j \in B^{m,l}$; inter-node connections are based on containment so that the node corresponding to $v_{j,l}, j = 1..2^{ml}$ is connected to a node $v_{j',l+1}, j' = 2^{m(l+1)}$ if and only if $B_{j'} \subset B_j$. For a fixed m, let $B^* \in B^{m,\log s^{-1}}$ be the cube containing $X^1_{n_1-m+1..n_1}$. Let $v'_{jl} = (\nu(\mathbf{x}'_1,B),\nu(\mathbf{x}_2,B)), j = 1..2^{ml}, l = 1..\log s_0^{-1}$, where $s_0 := \max\{s, \min_{\substack{X^u_i\neq X^1_{n_1} \\ u=1,2,i=1..n_u}} |X^u_i - X^1_{n_1}|\}$. To obtain the 2^m-ary tree corresponding to $v_{j,l}, j = 1..2^{ml}, l = 1..\log s^{-1}$ we can first extend that already generated for $v'_{j,l}, j = 1..2^{ml}, l = 1..\log s_0^{-1}$ to include a path to B^*, and next traverse the path bottom-up to update the frequency values. This entails $\log s^{-1}$ computations as this path is of length $\log s^{-1}$. Moreover, by definition we have that $\nu(\mathbf{x},B) = 0$, for all $B \in B^{m,l}$, with $m \geq \max\{n_1,n_2\}$ and $l \in \mathbb{N}$, therefore a total of $\max\{n_1,n_2\}\log s^{-1}$ frequency updates are required to obtain $\hat{d}(\mathbf{x}_1,\mathbf{x}_2)$ from $\hat{d}(\mathbf{x}'_1,\mathbf{x}_2)$.

Proof of Theorem 3.1 i. Consistency: To show the consistency of Alg 2 we show that for every fixed number $N \in \mathbb{N}$ of sequences, there exists some time T, such that for all $t \geq T$ and all $r \in \mathcal{I}_i \cap 1..N$, $i \in 1..k$, we have that $\operatorname{argmin}_{i'\in1..k} \frac{1}{\eta}\sum_{j=1}^{N(t)} \alpha_j \hat{d}(\mathbf{x}^t_r, \mathbf{x}^t_{R_{i'j}}) = i$, so that from T on, r is mapped to the correct target cluster, (i.e. $\{C_i(t) \cap 1..N\} = \{\mathcal{I}_i \cap 1..N\}$, for all $i = 1..k$).

Fix an $\varepsilon > 0$. We can find an index J such that $\sum_{j=J}^{\infty} w_j \leq \varepsilon$. Denote by $S(t)|_j = \{\mathbf{x}^t_1,\cdots,\mathbf{x}^t_j\}$, the subset of $S(t)$ consisting of the first j sequences for $j \in 1..N(t)$. For $i = 1..k$ let $s_i := \min_{\mathbf{x}^t_j\sim\rho_i} j$ index the first sequence in $S(t)$ that is generated by ρ_i, and denote by

$$m := \max_{i\in1..k} s_i, \tag{6}$$

the maximum such index. Let $t(j,\varepsilon)$ be the time-step such that for all $t \geq t(j,\varepsilon)$ we have

$$|\hat{d}(\mathbf{x},\rho_y) - d(\rho_x,\rho_y)| \leq \varepsilon, \tag{7}$$

$$|\hat{d}(\mathbf{x},\mathbf{y}) - d(\rho_x,\rho_y)| \leq \varepsilon \tag{8}$$

for all pairs of sequences $\mathbf{x} \sim \rho_x$, and $\mathbf{y} \sim \rho_y$ in $S(t)|_j$, and that Alg1$(S(t)|_j, k)$ is consistent unless ofcourse if $j < m$. Such time-step always exists (with probability 1), due to the consistency of $\hat{d}(\cdot,\cdot)$ (Equations 2 and 3), the consistency of Alg 1 [11], and the assumption that the sequence lengths grow indefinitely with time.

Let $\delta := \frac{1}{2}\min_{i\neq j\in 1..k} d(\rho_i,\rho_j)$ and define

$$t(j) := \max\{t(m,\delta), \max_{j=1..J} t(j,\varepsilon)\}.$$

By (6) and (8) for all $t \geq t(j)$ we have,

$$|\gamma_m^t - \min_{i\neq j\in\{1,\cdots,k\}} d(\rho_i,\rho_j)| \leq \delta.$$

Hence, recalling that (as specified in Alg 2) $\eta = \sum_{j=1}^{N(t)} w_j\gamma_j^t$ we have $\eta \geq w_m\delta$, and since $\hat{d}(\cdot,\cdot) \leq 1$ we obtain,

$$\frac{1}{\eta}\sum_{j=1}^{N(t)} \alpha_j \hat{d}(\mathbf{x}^t_{R_{ij}},\rho_i) \leq \frac{1}{\eta}\sum_{j=1}^{J} \alpha_j \hat{d}(\mathbf{x}^t_{R_{ij}},\rho_i) + \frac{\varepsilon}{w_m\delta}. \tag{9}$$

The sequences in $S(t)|_j$ for $j = 1..m-1$ are generated by *at most* $k-1$ out of the k generating distributions. Therefore, $\gamma_j \leq \varepsilon$ for $j = 1..m-1$, since there exists at least one pair of distinct cluster representatives generated by *the same* distribution. We have,

$$\frac{1}{\eta}\sum_{j=1}^{m-1} w_j\gamma_j \hat{d}(\mathbf{x}^t_{R_{ij}},\rho_i) \leq \frac{1}{\eta}\sum_{j=1}^{m-1} w_j\gamma_j^t \leq \frac{\varepsilon}{w_m\delta}. \tag{10}$$

Noting that the clusters are ordered in the order of appearance of the distributions, we have $\mathbf{x}^t_{R_{ij}} = \mathbf{x}_{s_i}$ for all $j = m..J$ and $i = 1..k$. Therefore,

$$\frac{1}{\eta}\sum_{j=m}^{J} \alpha_j \hat{d}(\mathbf{x}^t_{R_{ij}},\rho_i) = \frac{1}{\eta}\hat{d}(\mathbf{x}^t_{s_i},\rho_i)\sum_{j=m}^{J}\alpha_j \leq \varepsilon, \tag{11}$$

for every $i = 1..k$. Combining (9), (10), and (11) we obtain

$$\frac{1}{\eta}\sum_{j=1}^{N(t)} \alpha_j \hat{d}(\mathbf{x}^t_{R_{ij}},\rho_i) \leq 2\frac{\varepsilon}{w_m\delta} + \varepsilon \tag{12}$$

for every $i = 1..k$.

Consider an index $r \in \mathcal{I}_i \cap 1..N$ for some $N \in 1..|S(t)|$. Let $T := t(N)$. For all $t \geq T$ and all $i' \neq i \in 1..k$ we have,

$$\begin{aligned}
&\frac{1}{\eta}\sum_{j=1}^{N(t)} \alpha_j \hat{d}(\mathbf{x}^t_r, \mathbf{x}^t_{R_{i'j}}) \\
&\geq \frac{1}{\eta}\sum_{j=1}^{N(t)} \alpha_j \hat{d}(\mathbf{x}^t_r,\rho_{i'}) - \frac{1}{\eta}\sum_{j=1}^{N(t)} \alpha_j \hat{d}(\mathbf{x}^t_{R_{i'j}},\rho_{i'}) \\
&\geq d(\rho_i,\rho_{i'}) - \varepsilon - \frac{1}{\eta}\sum_{j=1}^{N(t)} \alpha_j \hat{d}(\mathbf{x}^t_{R_{i'j}},\rho_{i'}) \\
&\geq\ 2\delta - 2\varepsilon(1 + \frac{1}{w_m\delta}),
\end{aligned} \tag{13}$$

where the first inequality follows from applying the triangle inequality to $\hat{d}(\cdot,\cdot)$, the second inequality follows from (7) and the last inequality follows from (12) and the definition of δ. Since the choice of ε is arbitrary, from (7) and (13) we obtain,

$$\operatorname*{argmin}_{i'\in 1..k} \frac{1}{\eta}\sum_{j=1}^{N(t)} \alpha_j \hat{d}(\mathbf{x}^t_r, \mathbf{x}^t_{R_{i'j}}) = i,$$

implying the consistency statement. Moreover, by Lemma 2 of [11] the same consistency result holds if we replace $\hat{d}(\cdot,\cdot)$ in the algorithm by any corresponding consistent estimate $\check{d}(\cdot,\cdot)$ defined by Equation 4.

ii. Computational Complexity: Let $N := N(t)+1$ and denote by D the $(N-1)\times(N-1)$ (symmetric) matrix of pairwise distances between the sequences in $\{\mathbf{x}_1,\cdots,\mathbf{x}_{N-1}\}$, i.e. $D_{ij} := \hat{d}(\mathbf{x}_i,\mathbf{x}_j)$, $i,j \in N-1$. Assume that a new symbol $X \in \mathcal{A}$ arrives along with an indicator of where it belongs, i.e. whether it is a continuation of a previous sequence $\mathbf{x}_i$ for some $i \in N-1$ or that it is to start a new sequence, $\mathbf{x}_N$. In the former case, the i^{th} row and column of D need updating whereas in the latter, a new row and column are to be added to D so that for all $i = 1..N-1$, $D_{Ni} = D_{iN} = \hat{d}(\mathbf{x}_N,\mathbf{x}_i)$. In both cases a total of $N-1$ distance updates are required to update D, which by Proposition 3.1 yield a computational complexity of order $\mathcal{O}(N(t)n_{\max}\log s_{\min}^{-1}\})$. Apart from the distance calculations, the rest of the computations, namely those corresponding to updating **1.** the cluster-representative matrix R, **2.** the weighted intercluster distances $\alpha_j = 2^{-j}\gamma_j$, $j = 1..N$ and **3.** the clusters $C_i(t)$, $i = 1..k$, are of order $\mathcal{O}(kN(t)^2)$. Thus, the per-symbol resource complexity of Alg 2 at timestep t is of order $\mathcal{O}(kN(t)^2 + N(t)n_{\max}\log s_{\min}^{-1})$. The statement regarding $\check{d}(\cdot,\cdot)$ can be derived analogously.

4 Experimental Results

We present empirical evaluations of the considered framework. In our experiments $\check{d}(\cdot,\cdot)$ is used with

$m_n = \log n$, in order to optimize running time. The choice of this parameter value is justified as follows: The expected waiting time before a word $\mathbf{y} = Y_{1..m}$ is repeated within a sequence $\mathbf{x} = X_{1..n}$ is inversely proportional to its probability of occurrence, $P(\mathbf{y})$. For a sequence generated by an ergodic source with entropy rate h $P(\mathbf{y})$ is asymptotically of order 2^{-mh}. Therefore, counting the frequencies of words $\mathbf{y} = Y_{1..m}$ in $\mathbf{x}$ for $m > \log n$ does not result in a consistent estimate of $P(\mathbf{y})$. While the consistency of the distance estimates and thus the algorithm is not affected by this choice, it helps reduce the computational complexity (cf. Theorem 3.1). Apart from this choice of the parameters m_n, no parameter tuning was used to achieve the empirical results (the values of the other parameters, such as the weights w_j, were set to defaults described in Section 3).

4.1 Synthetic Data

In this section we present empirical evaluations of our algorithms on synthetically generated data. To put the generality of our approach to a test, we have selected time-series distributions that, while being stationary ergodic, do not belong to any "simpler" general class of time-series, and are difficult to approximate by finite-state models. The considered processes are taken from [15], where they are used as an example of stationary ergodic processes that are not B-processes. Such time-series cannot be modeled by a hidden Markov model with a finite or countably infinite set of states. Moreover, k-order Markov or hidden Markov approximations of this process do not converge to it in $\bar{d}$ distance, a distance that is stronger than d, and whose empirical approximations are often used to study general (non-Markovian) processes (see, e.g. [10]).

Time-series generation. To generate a sequence $\mathbf{x} = X_{1..n}$ we proceed as follows: Fix some parameter $\alpha \in (0, 1)$. Select $r_0 \in [0, 1]$; then, for each $i = 1..n$ obtain r_i by shifting r_{i-1} by α to the right, and removing the integer part, i.e. $r_i := r_{i-1} + \alpha - \lfloor r_{i-1} + \alpha \rfloor$. The sequence $\mathbf{x} = (X_1, X_2, \cdots)$ is then obtained from r_i by thresholding at 0.5, that is $X_i := \mathbb{I}\{r_i > 0.5\}$. We call this procedure $DAS(\alpha)$. If α is irrational then $\mathbf{x}$ forms a stationary ergodic time-series. [2]

For the purpose of our experiments, first we use five process distributions different processes $DAS(\alpha_i)$, $i = 1..5$, with $\alpha_1 = 0.31...$, $\alpha_2 = 0.33...$, $\alpha_3 = 0.35...$, $\alpha_4 = 0.37...$, $\alpha_5 = 0.39$. Next we generate an $N \times M$ data matrix $\mathbf{X}$, each row of which is a sequence generated by one of the five $DAS(\alpha_i)$, $i = 1..5$ processes. Our task in both the online and the batch setting is to cluster the rows of $\mathbf{X}$ into $k = 5$ clusters. The selected α_i are intentionally chosen to be close, making the processes harder to distinguish.

Experiment 1. (Batch Setting) In this experiment we demonstrate that in the batch setting, the clustering errors corresponding to both the online and the offline algorithms converge to 0 as the sequence-lengths grow. To this end, at every time-step t we take an $N \times n(t)$ sub-matrix $\mathbf{X}|_{\mathbf{n(t)}}$ of $\mathbf{X}$ composed of the rows of $\mathbf{X}$ terminated at length $n(t)$, where $n(t) = 5t$. Then at each iteration we let each of the algorithms, (online and offline) cluster the rows of $\mathbf{X}|_{\mathbf{n(t)}}$ into five clusters, and calculate the clustering error-rate of each algorithm. As shown in Figure 2 (top) the error-rate of both algorithms decrease with sequence-length.

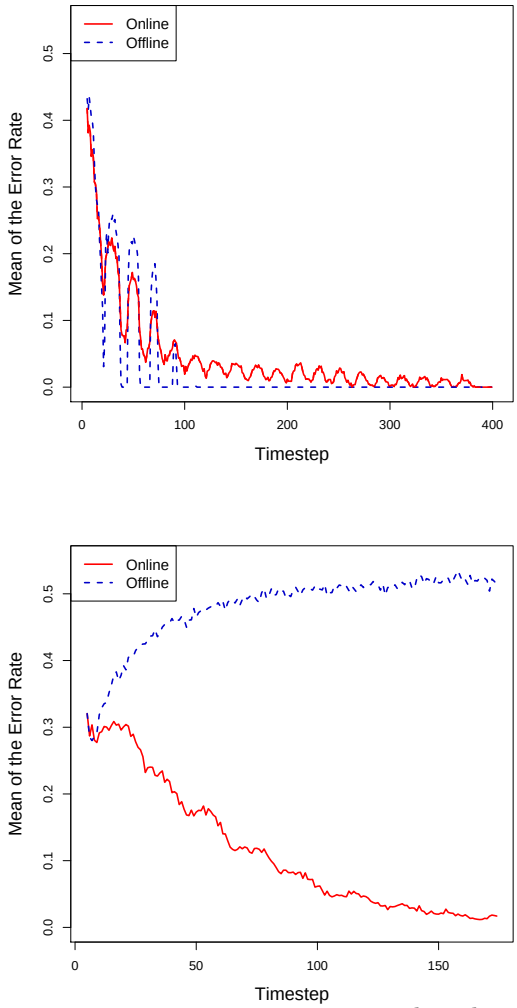

Figure 2: Top: error-rate vs. sequence length in batch setting, Bottom: error-rate vs. # of observed samples in online setting. (error-rates averaged over 100 runs.)

Experiment 2. (Online Setting) In this experiment we demonstrate that, unlike the online algorithm, the offline algorithm is consistently confused by the new sequences arriving at each time step in

[2] We simulate α by a `longdouble` with a long mantissa.

an online setting. To simulate an online setting, we proceed as follows: At every time-step t, a triangular window is used to reveal the first $1..n_i(t),\ i = 1..t$ elements of the first t rows of the data-matrix $\mathbf{X}$, with $n_i(t) := 5(t-i)+1,\ i = 1..t$. This gives a total of t sequences, each of length $n_i(t)$, for $i = 1..t$, where the i^{th} sequence for $i = 1..t$ corresponds to the i^{th} row of $\mathbf{X}$ terminated at length $n_i(t)$. At every time-step t the online and offline algorithms are each used to in turn cluster the observed t sequences into five clusters. As shown in Figure 2 (bottom), in this setting the clustering error-rate of the offline algorithm remains consistently high, whereas that of the online algorithm converges to zero.

4.2 Real Data

As a real application we consider the problem of clustering motion capture sequences, where groups of sequences with similar dynamics are to be identified. Data is taken from the Motion Capture database (MOCAP) [1] which consists of time-series data representing human locomotion. The sequences are composed of marker positions on human body which are tracked spatially through time for various activities.

We compare our results against two other methods, namely those of [9] and [6], which (to the best of our knowledge) constitute the state-of-the-art performance on these datasets. Note that we have not implemented these reference methods, rather we have taken their numerical results directly from their corresponding articles. In order to have common grounds for each comparison we use the same sets of sequences,[3] and the same means of evaluation as those used in [9, 6].

In [9] two MOCAP datasets[4] are used, where the sequences in each dataset are labeled with either running or walking as annotated in the database. Performance is evaluated via the conditional entropy S of the true labeling with respect to the prediction, i.e. $S = -\sum_{i,j} \frac{\mathcal{M}_{ij}}{\sum_{i',j'} \mathcal{M}_{i'j'}} \log \frac{\mathcal{M}_{ij}}{\sum_{j'} \mathcal{M}_{ij'}}$ where $\mathcal{M}$ denotes the clustering confusion matrix. The motion sequences used in [9] are reportedly trimmed to equal duration. However, we use the original sequences as our method is not limited by variation in sequence lengths. Table 1 lists performance of Alg 1 as well as that reported for the method of [9]; Alg 1 performs consistently better.

In [6] four MOCAP datasets[5] are used, corresponding to four motions: run, walk, jump and forward jump. Table 2 lists performance in terms of accuracy. The datasets in Table 2 constitute two types of motions: **1.** motions that can be considered ergodic (walk, run, run/jog; displayed above the double line), and **2.** non-ergodic motions (single jumps; displayed below the double line). As shown in Table 2, Alg 1 achieves consistently better performance on the first group of datasets, while being competitive (better on one and worse on another) on the non-ergodic motions. The time taken to complete each task is in the order of few minutes on a standard laptop computer.

	Dataset	**[9]**	Alg 1
1.	Walk vs. Run (#35)	0.1015	**0**
2.	Walk vs.Run (#16)	0.3786	**0.2109**

Table 1: Comparison with [9]: Performance in terms of entropy; data-sets concern ergodic motion captures.

	Dataset	**[6]**	Alg 1
1.	Run(#9) vs. Run/Jog(#35)	**100%**	**100%**
2.	Walk(#7) vs. Run/Jog(#35)	95%	**100%**
3.	Jump vs. Jump fwd.(#13)	87%	**100%**
4.	Jump vs. Jump fwd.(#13, 16)	**66%**	60%

Table 2: Comparison with [6]: Performance in terms of accuracy; Rows 1 & 2 concern ergodic, Rows 3 & 4 concern non-ergodic motion captures.

5 Outlook

The framework for clustering time-series considered in this work is fairly new, giving rise to many open problems and exciting directions for further research. While in this work we have assumed that the number of clusters k is known, in practice this may not be the case. Therefore, an interesting extension would be to consider the case of unknown number of clusters. As shown in [12] in general it is impossible to decide whether a pair of observed sequences are generated by the same or by two different stationary ergodic distributions. One mitigation is to make stronger assumptions on the data. Another approach often considered in the clustering literature is to construct a hierarchy of clustering for different k, in such a way that the true clustering is a pruning of this hierarchy.

Acknowledgements

This work is supported by the French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council and FEDER through CPER 2007-2013, ANR projects EXPLO-RA (ANR-08-COSI-004) and Lampada (ANR-09-EMER-007), by an INRIA Ph.D. grant to Azadeh Khaleghi, by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 231495 (project CompLACS), and by Pascal-2.

[3] marker position: the subject's right foot.
[4] subjects #16 and #35.
[5] subjects #7, #9, #13, #16 and #35.

References

[1] CMU graphics lab motion capture database. `http://mocap.cs.cmu.edu/`, 2009.

[2] F.R. Bach and M.I. Jordan. Learning graphical models for stationary time series. *IEEE Transactions on Signal Processing*, 52(8):2189 – 2199, aug. 2004.

[3] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(7):719–725, 2000.

[4] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[5] R. Gray. *Probability, Random Processes, and Ergodic Properties*. Springer Verlag, 1988.

[6] T. Jebara, Y. Song, and K. Thadani. Spectral clustering and embedding with hidden Markov models. *Machine Learning: ECML 2007*, pages 164–175, 2007.

[7] J. Kleinberg. An impossibility theorem for clustering. In *15th Conf. Neural Information Processing Systems (NIPS'02)*, pages 446–453, Montreal, Canada, 2002. MIT Press.

[8] M. Kumar, N.R. Patel, and J. Woo. Clustering seasonality patterns in the presence of errors. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge Discovery and Data mining*, pages 557–563. ACM, 2002.

[9] Lei Li and B. Aditya Prakash. Time series clustering: Complex is simpler! In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 185–192, New York, NY, USA, June 2011. ACM.

[10] D.S. Ornstein and B. Weiss. How sampling reveals a process. *Annals of Probability*, 18(3):905–930, 1990.

[11] D. Ryabko. Clustering processes. In *Proc. the 27th International Conference on Machine Learning (ICML 2010)*, pages 919–926, Haifa, Israel, 2010.

[12] D. Ryabko. Discrimination between B-processes is impossible. *Journal of Theoretical Probability*, 23(2):565–575, 2010.

[13] D. Ryabko. Testing composite hypotheses about discrete ergodic processes. *Test*, 2012 (to appear).

[14] D. Ryabko and B. Ryabko. Nonparametric statistical inference for ergodic processes. *IEEE Transactions on Information Theory*, 56(3):1430–1435, 2010.

[15] P. Shields. *The Ergodic Theory of Discrete Sample Paths*. AMS Bookstore, 1996.

[16] P. Smyth. Clustering sequences with hidden Markov models. In *Advances in Neural Information Processing Systems*, pages 648–654. MIT Press, 1997.

[17] R. Zadeh and S. Ben-David. A uniqueness theorem for clustering. In A. Ng J. Bilmes, editor, *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI'09)*, Montreal, Canada, 2009.

[18] Shi Zhong and Joydeep Ghosh. A unified framework for model-based clustering. *Journal of Machine Learning Research*, 4:1001–1037, 2003.