Consistent algorithms for clustering time series

Azadeh Khaleghi AZADEH.KHALEGHI@INRIA.FR
Daniil Ryabko DANIIL@RYABKO.NET
Jérémie Mary JEREMIE.MARY@INRIA.FR
Philippe Preux PHILIPPE.PREUX@INRIA.FR
SequeL-INRIA/LIFL-CNRS,
Université de Lille, France

Editor: ?

Abstract

[1] The problem of clustering is considered for the case where every point is a time series. The time series are either given in batch (offline setting), or they are allowed to grow with time and new time series can be added along the way (online setting). We propose a natural notion of consistency for this problem, and show that there are simple, computationally efficient algorithms that are asymptotically consistent under extremely weak assumptions on the distributions that generate the data. The notion of consistency is as follows: a clustering algorithm is called consistent if it puts two time series into the same cluster if and only if the distribution that generates them is the same. In the considered framework the time series are allowed to be highly dependent, and the dependence can have arbitrary form. For the case of a known number of clusters, the only assumption we make is that the (marginal) distribution of each time series is stationary ergodic. No parametric, memory or mixing assumptions are made. For the case of unknown number of clusters, stronger assumptions are provably necessary, but non-parametric algorithms consistent under very general conditions are still possible. The theoretical findings of this work are illustrated with experiments on both synthetic and real data.

Keywords: Clustering, Time-Series, Ergodicity, Unsupervised Learning

1. Introduction

Clustering is a well-explored problem in machine learning, and is key to many applications in almost all fields of science. The goal is to partition a given dataset into a set of non-overlapping *clusters* in some natural way, thus hopefully revealing an underlying structure in the data. In particular, this problem for time series data is motivated by many research problems from a variety of disciplines, such as marketing and finance, biological and medical research, video/audio analysis, etc., with the common feature that the data are abundant while little is known about the nature of the processes that generate them.

While intuitively appealing, the problem of clustering is notoriously difficult to formalize. First, an intrinsic part of the problem is a similarity measure: the points in each cluster are supposed to be close to each other in some sense. Most formulations simply assume the similarity measure to be given: it is either some fixed metric, or just a matrix of similarities

1. This is an extended version of two conference papers: Ryabko (2010b) and Khaleghi et al. (2012).

between the points. However, even if the similarity measure is fixed, it is still unclear how to define good clustering. Thus, Kleinberg (2002) presents a set of fairly natural properties that a good clustering function should have, and then proceeds to demonstrate that there is no clustering function with these properties. A common approach is thus to fix not only the similarity metric, but also some specific objective function — typically along with the number of clusters — and to construct algorithms to maximize this objective. Even this approach has some fundamental difficulties, albeit of a different, this time, computational, nature: already in the case where the number of clusters is known, and the distance between the points is set to be the Euclidean distance, optimizing some fairly natural objectives (such as k-means) turns out to be NP hard (Mahajan et al., 2009).

In this paper we consider a subset of the clustering problem, namely, clustering time series. That is, we consider the situation in which each data point is a sample drawn from some time series distribution. The time series distributions in question are assumed to be completely unknown. While at first glance this does not appear to be a simplification, we make two observations that allow us to define consistency of clustering for this problem, and to show that it is achievable under minimal assumptions by computationally efficient algorithms. First, note that time series present a different dimension of asymptotic: with respect to the length of the time series, rather than with respect to the number of points to cluster. Second, note that certain types of asymptotic statistical inference are possible for a very large class of time series distributions. Specifically, a growing segment of time series generated by a stationary ergodic distribution allows one to estimate any finite-dimensional characteristic of this distribution with arbitrary precision. The assumption that a given time series is stationary ergodic is one of the most general assumptions used in statistics; in particular, it allows for arbitrary long-range serial dependence, and subsumes most of the non-parametric as well as modelling assumptions used in the literature on clustering time series, such as i.i.d., (Hidden) Markov, or mixing time series.

This allows us to define the following clustering objective: *put two time series into the same cluster if and only if the distribution generating them is the same.*

1.1 Problem setup

We consider two variants of the clustering problem in this setting: offline (batch) and online, defined as follows.

In the *offline (batch)* setting a finite number N of sequences $\mathbf{x}_1 = (X_1^1, \dots, X_{n_1}^1), \dots, \mathbf{x}_N = (X_1^N, \dots, X_{n_N}^N)$ is given. Each sequence is generated by one of k different *unknown* stationary ergodic process distributions. No restrictions are placed on the dependence between the sequences (this dependence may be arbitrary). The target clustering is the partitioning of $\mathbf{x}_1, \dots, \mathbf{x}_N$ into k clusters, putting together those and only those sequences that were generated by the same time series distribution. We call a batch clustering algorithm *asymptotically consistent* if, with probability 1, it stabilizes on the target clustering in the limit as the lengths $n_1, \dots, n_N$ of all of the N samples tend to infinity. The number k of clusters may be either known or unknown. Note that the asymptotic is not with respect to the number of sequences, but with respect to the individual sequence lengths.

In the *online* setting we have a growing body of sequences of data. Both the number of sequences and sequences themselves grow with time. The manner of this evolution can be arbitrary; we only require that the length of each individual sequence tends to infinity. Similarly to the batch setting, the joint distribution generating the data is unknown; there can be any (adversarial) dependence between the samples. At time-step 1 initial segments of some of the first sequences are available to the learner. At each subsequent time-step, new data are revealed, either as an extension of previously observed sequence, or as a new sequence. Thus, at each time-step t a total of $N(t)$ sequences $\mathbf{x}_1, \dots, \mathbf{x}_{N(t)}$ 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. In the online setting, a clustering algorithm is called asymptotically consistent, if for each fixed batch of sequences $\mathbf{x}_1, \dots, \mathbf{x}_N$, the clustering restricted to this sequences coincides with the target clustering from some time on.

At the first glance it may seem that one can use the offline algorithm in the online case, simply applying it to the entire data observed at every time-step. However, this naive approach does not result in a consistent algorithm. The main challenge in the online setting can be identified with what we regard as "bad" sequences: recently-observed sequences for which sufficient information has not yet been collected, and that as such are not possible to be distinguished based on the process distributions that generate them. In this setting, using a batch algorithm at every time-step, results in not only mis-clustering such "bad" sequences, but also in clustering incorrectly those for which sufficient data are already available. That is, such "bad" sequences can render the entire batch clustering useless, leading the algorithm to incorrectly cluster even the "good" sequences. Since new sequences may arrive in an uncontrollable (even data-dependent, adversarial) fashion, any batch algorithm will fail in this scenario.

1.2 Results

In this paper we first provide an offline clustering algorithm that we show to be strongly asymptotically consistent provided that the number k of clusters is known. The consistency is established under the only assumption that the distribution generating each sequence is stationary ergodic. Next, we use the offline algorithm as a building block to construct an algorithm for the online setting, and show its asymptotic consistency for known k under the general online setup described, again assuming only stationarity and ergodicity of each time series distribution.

As follows from the theoretical impossibility results of (Ryabko, 2010c) (discussed in Section 1.4), under these assumptions only, it is impossible to find the correct number of clusters k, that is, the number of different time series distributions that generate the data. However, this is possible under additional conditions on the distributions; in particular, we show that if k is unknown, consistent algorithms are possible to obtain provided that an upper-bound α_n on the α-mixing rates of the joint distribution of the processes is known, and $\alpha_n \to 0$. Under this relaxation, besides the asymptotic result, finite-time bounds on the probability of incorrect clustering can be obtained. Again, these are provably impossible to obtain if we only assume that the distributions are stationary ergodic (see, e.g., Shields (1996)).

However, the main focus of this paper remains on the general framework where no additional assumptions on the unknown process distributions are made (other than that they are stationary ergodic). As such, the main theoretical and experimental results concern the case of known k.

Furthermore, we show that our methods can be implemented efficiently: they are at most quadratic in each of their arguments, and are linear (up to log terms) in some formulations.

To test the empirical performance of our algorithms we evaluated them on both synthetic and real data. To reflect the generality of the suggested framework in the experimental setup, we had our synthetic data generated by stationary ergodic process distributions that do not belong to any simpler class of distributions, and in particular cannot be modeled as hidden Markov processes with countable sets of states. 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 demonstrate the applicability of our work to real-world scenarios, we chose the problem of clustering motion-capture sequences of human locomotion. This application area has also been studied by (Li and Prakash, 2011) and (Jebara et al., 2007) that (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).

1.3 Methodology and algorithms

A crucial point of any clustering method is the similarity measure. Since in our formulation the objective is to cluster the time series based on the distribution that generates them, the similarity measure must reflect the difference between the underlying distributions. Moreover, our goal is to tackle the case of highly-dependent time series. Thus, we chose to *estimate* the distance between the *distributions* that generate the time series, based on the data given.

It turns out that a suitable distance for this purpose is the so-called distributional distance. The distributional distance $d(\rho_1, \rho_2)$ between a pair of process distributions ρ_1, ρ_2 is defined (Gray, 1988) as $\sum_{i \in \mathbb{N}} w_i |\rho_1(B_i) - \rho_2(B_i)|$ where, w_i are positive summable real weights, e.g. $w_i = 1/i(i+1)$, and B_i range over a countable field that generates the sigma-algebra of the underlying probability space. For example, consider finite-alphabet processes with the binary alphabet $\mathcal{X} = \{0, 1\}$. In this case $B_i,\ i \in \mathbb{N}$ would range over the set $\mathcal{X}^* = \cup_{m \in \mathbb{N}} \mathcal{X}^m$; that is, over all tuples $0, 1, 00, 01, 10, 11, 000, 001, \dots$; therefore, the distributional distance in this case is the weighted sum of the differences of the probability values (calculated with respect to ρ_1 and ρ_2) of all possible tuples. In this work we consider real-valued processes so that the sets B_k range over a suitable sequence of intervals, all pairs of such intervals, triples, and so on (see the formal definitions in Section 2). Although this distance involves infinite summations, we show that its empirical approximations can be easily calculated. Asymptotically consistent estimates of this distance can be obtained by replacing unknown probabilities with the corresponding frequencies (Ryabko and Ryabko, 2010); these estimator have proved useful in various statistical problems concerning ergodic processes (Ryabko and Ryabko, 2010; Ryabko, 2012; Khaleghi and Ryabko, 2012).

Armed with an estimator of the distributional distance, it is relatively easy to construct a consistent clustering algorithm for the batch setting. In particular, we show that the following algorithm is asymptotically consistent. First a set of k cluster centers are identified using k farthest point initialization (using $\hat{d}$). In particular the first sequence is assigned as the first cluster centre. Iterating over $2..k$, at every iteration a sequence is sought which has the largest minimum distance from the already chosen cluster centers. Next, the remaining sequences are assigned to the closest clusters.

The online algorithm is based on a *weighted combination* of several clusterings, each obtained by running the offline procedure on different portions of data. The partitions are combined with weights that depend on the batch size and on an appropriate performance measure for each individual partition. The performance measure of each clustering is the minimum inter-cluster distance.

1.4 Related Work

Some probabilistic formulations of the time series clustering problem can be related to ours. Perhaps the closest one is mixture models (Bach and Jordan, 2004; Biernacki et al., 2000; Kumar et al., 2002; Smyth, 1997; Zhong and Ghosh, 2003): it is assumed that the data is generated by a mixture of k different distributions, that have a particular known form (such as Gaussian, Hidden Markov models, or graphical models) Thus, each of the N samples is generated independently according to one of these k distributions (with some fixed probability). Since the model of the data is specified quite well, one can use likelihood-based distances (and then, for example, the k-means algorithm), or Bayesian inference, to cluster the data. Another typical objective is to estimate the parameters of the distributions in the mixture (e.g., Gaussians), rather than actually clustering the data points. Clearly, the main difference from our setting is in that we do not assume any known model of the data; not even between-sample independence is assumed.

The problem of clustering in our formulations generalizes two classical problems of mathematical statistics namely, homogeneity testing (or the two-sample problem) and process classification (or the three-sample problem). In the two-sample problem, given two sequences $\mathbf{x}_1 = (x_1^1, \dots, x_{n_1}^1)$ and $\mathbf{x}_2 = (x_1^2, \dots, x_{n_2}^2)$ it is required to test whether they are generated by the same or by different process distributions. This corresponds to the special case of clustering only $N = 2$ data points with the number k of clusters unknown: where k could be either 1 or 2. In the three-sample problem. Three sequences $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are given, it is known that two of them are generated by the same process distribution, while one is generated by a different process distribution. It is required to find the two sequences that were generated by the same process distribution. This corresponds to clustering $N = 3$ data points, with a known number $k = 2$ of clusters. The classical approach is of course to consider Gaussian i.i.d. data, but general non-parametric solutions exist not only for i.i.d. data (Lehmann, 1986), but also for Markov chains (Gutman, 1989), and under certain mixing rates conditions. Observe that the two-sample problem is more difficult to solve than the three-sample problem, as the number k of clusters is unknown in the former while it is given in the latter. Indeed, as shown by (Ryabko, 2010c) in general for stationary ergodic (binary-valued) processes there is no solution to the two-sample problem, even in the weakest asymptotic sense. However, a solution to the three-sample problem, for (real-valued) stationary ergodic processes was given in (Ryabko and Ryabko, 2010). This means that in the framework that we consider it is impossible to consistently estimate the number k of clusters. Moreover, rates of convergence are provably impossible to obtain, so the asymptotic results of this work cannot be strengthened.

1.5 Organization

The remainder of the paper is organized as follows. We start with some preliminary notations and definitions in Section 2. In Section 3 we define the considered clustering protocol. Our main theoretical results are presented in Section 4, where we present our methods, state and prove their asymptotic consistency, and disucuss their computational complexity. In Section 6 we provide some experimental evaluations on both synthetic and real data. Finally in Section 7 we provide some concluding remarks and open directions.

2. Preliminaries

Let $\mathcal{X}$ be a measurable space (the domain); in this work we let $\mathcal{X} = \mathbb{R}$ but extensions to more general spaces are straightforward. For a sequence $X_1, \dots, X_n$ we use the abbreviation $X_{1..n}$. Consider the Borel σ-algebra $\mathcal{B}$ on $\mathcal{X}^\infty$ generated by the cylinders $\{B \times \mathcal{X}^\infty : B \in B^{m,l}, m, l \in \mathbb{N}\}$ where, the sets $B^{m,l}, m, l \in \mathbb{N}$ are obtained via the partitioning of $\mathcal{X}^m$ into cubes of dimension m and volume 2^{-ml} (starting at the origin). Let also $B^m := \cup_{l \in \mathbb{N}} B^{m,l}$. We could choose any other way to generate the Borel sigma algebra $\mathcal{B}$ on $\mathcal{X}$, but for the sake of computation we need to fix a specific choice. Process distributions are probability measures on the space $(\mathcal{X}^\infty, \mathcal{B})$. For a sequence $X_1, \dots, X_n$ we use the abbreviation $X_{1..n}$. Similarly, one can define distributions over the space $((\mathcal{X}^\infty)^\infty, \mathcal{B}_2)$ of infinite matrices with the Borel σ-algebra $\mathcal{B}_2$ generated by the cylinders $\{(\mathcal{X}^\infty)^k \times (B \times \mathcal{X}^\infty) \times (\mathcal{X}^\infty)^\infty : B \in B^{m,l}, k, m, l \in \mathbb{N}\}$. For $\mathbf{x} = X_{1..n} \in \mathcal{X}^n$ and $B \in B^m$ let $\nu(\mathbf{x}, B)$ denote the *frequency* with which $\mathbf{x}$ falls in B, i.e.

$$\nu(\mathbf{x}, B) := \frac{\mathbb{I}\{n \geq m\}}{n - m + 1} \sum_{i=1}^{n-m+1} \mathbb{I}\{X_{i..i+m-1} \in B\} \tag{1}$$

A process ρ is *stationary* if for any $i, j \in 1..n$ and $B \in \mathcal{B}$, we have

$$\rho(X_{1..j} \in B) = \rho(X_{i..i+j-1} \in B).$$

A stationary process ρ is called *ergodic* if for all $B \in \mathcal{B}$ with probability 1 we have

$$\lim_{n \to \infty} \nu(X_{1..n}, B) = \rho(B).$$

By virtue of the ergodic theorem (e.g. Billingsley, 1961), this definition can be shown to be equivalent to the standard definition of stationary ergodic processes (every shift-invariant set has measure 0 or 1; see, e.g., Gray, 1988).

Definition 1 (Distributional Distance) *The distributional distance between a pair of process distributions ρ_1, ρ_2 is defined as follows (Gray, 1988).*

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

where we set $w_j := 1/j(j+1)$, but any summable sequence of positive weights may be used.

In words, this involves partitioning the sets $\mathcal{X}^m$, $m \in \mathbb{N}$ into cubes of decreasing volume (indexed by l) and then taking a sum over the differences in probabilities of all the cubes in these partitions. The differences in probabilities are weighted: smaller weights are given to larger m and finer partitions. We use *empirical estimates* of this distance defined as follows.

Remark 2 *The distance is more generally defined as $d(\rho_1, \rho_2) := \sum_{i\in\mathbb{N}} w_i|\rho_1(A_i) - \rho_2(A_i)|$ where A_i range over a countable field that generates the σ-algebra of the underlying probability space (Gray, 1988). We deliberately use Definition 1 above for practical purposes. By this choice, the methods based on the distributional distance and its empirical estimates defined below, not only possess theoretical guarantees, but are also easily implementable.*

Definition 3 (Empirical estimates of $d(\cdot,\cdot)$) *Define the empirical estimate of the distributional distance between a sequence $\mathbf{x} = X_{1..n} \in \mathcal{X}^n, n \in \mathbb{N}$ and a process distribution ρ by*

$$\hat{d}(\mathbf{x}, \rho) := \sum_{m=1}^{m_n}\sum_{l=1}^{l_n} w_m w_l \sum_{B\in B^{m,l}} |\nu(\mathbf{x}, B) - \rho(B)| \tag{2}$$

and that between a pair of sequences $\mathbf{x}_i \in \mathcal{X}^{n_i}$ $n_i \in \mathbb{N}$, $i = 1, 2$ as

$$\hat{d}(\mathbf{x}_1, \mathbf{x}_2) := \sum_{m=1}^{m_n}\sum_{l=1}^{l_n} w_m w_l \sum_{B\in B^{m,l}} |\nu(\mathbf{x}_1, B) - \nu(\mathbf{x}_2, B)| \tag{3}$$

where m_n and l_n are any sequences that go to infinity with n and $w_j := 1/j(j+1)$, although any summable set of weights may also be used.

Remark 4 (constants m_n, l_n) *The sequences of constants $m_n, l_n (n \in \mathbb{N})$ in the definition of $\hat{d}$ are intended to introduce some computational flexibility in the estimate: while already the choice $m_n, l_n \equiv \infty$ is computationally feasible, other choices give better computational complexity while not sacrificing the quality of the estimate; see Section 5. For the asymptotic consistency results this choice is irrelevant. Thus, in the proofs we tacitly assume $m_n, l_n \equiv n = \infty$; the extension to the general case is straightforward.*

Lemma 5 ($\hat{d}$ is asymptotically consistent Ryabko and Ryabko (2010)) *For every pair of sequences $\mathbf{x}_1 \in \mathcal{X}^{n_1}$ and $\mathbf{x}_2 \in \mathcal{X}^{n_2}$ with* (an arbitrary) *joint distribution ρ and stationary ergodic marginals ρ_i, $i = 1, 2$ we have*

$$\lim_{n_1,n_2\to\infty} \hat{d}(\mathbf{x}_1, \mathbf{x}_2) = d(\rho_1, \rho_2),\ \ \rho - a.s.,\ \ and \tag{4}$$

$$\lim_{n_i\to\infty} \hat{d}(\mathbf{x}_i, \rho_j) = d(\rho_i, \rho_j),\ \ i, j \in 1, 2,\ \ \rho - a.s. \tag{5}$$

The proof of this lemma can be found in (Ryabko and Ryabko, 2010); since it is important for further development but rather simple, we reproduce it here.

Proof The idea of the proof is simple: for each set $B \in \mathcal{B}$, the frequency with which the sample $\mathbf{x}_i$, $i = 1, 2$ falls into B converges to the probability $\rho_i(B)$, $i = 1, 2$. When the sample sizes grow, there will be more and more sets $B \in \mathcal{B}$ whose frequencies have already converged to the probabilities, so that the cumulative weight of those sets whose frequencies have not converged yet, will tend to 0.

Fix $\varepsilon > 0$. We can find an index J such that

$$\sum_{m,l=J}^{\infty} w_m w_l \leq \varepsilon/3.$$

Moreover, for each $m, l \in 1..J$ we can find a finite subset $S^{m,l}$ of $B^{m,l}$ such that

$$\rho_i(S^{m,l}) \geq 1 - \frac{\varepsilon}{6Jw_m w_l}$$

There exists some N (which depends on the realization $X_1^i, \dots, X_{n_i}^i,\ i = 1,2$) such that for all $n_i \geq N,\ i = 1,2$ we have,

$$\sup_{B \in S^{m,l}} |\nu(\mathbf{x}_i, B) - \rho_i(B)| \leq \frac{\varepsilon \rho_i(B)}{6Jw_m w_l},\ i = 1,2 \tag{6}$$

For all $n_i \geq N,\ i = 1,2$ we have

$$|\hat{d}(\mathbf{x}_1, \mathbf{x}_2) - d(\rho_1, \rho_2)| = \left| \sum_{m,l=1}^{\infty} w_m w_l \sum_{B \in B^{m,l}} \big(|\nu(\mathbf{x}_1, B) - \nu(\mathbf{x}_2, B)| - |\rho_1(B) - \rho_2(B)| \big) \right|$$
$$\leq \sum_{m,l=1}^{\infty} w_m w_l \sum_{B \in B^{m,l}} |\nu(\mathbf{x}_1, B) - \rho_1(B)| + |\nu(\mathbf{x}_2, B) - \rho_2(B)|$$
$$\leq \sum_{m,l=1}^{J} w_m w_l \sum_{B \in S^{m,l}} \big(|\nu(\mathbf{x}_1, B) - \rho_1(B)| + |\nu(\mathbf{x}_2, B) - \rho_2(B)| \big) + 2\varepsilon/3$$
$$\leq \sum_{m,l=1}^{J} w_m w_l \sum_{B \in S^{m,l}} \frac{\rho_1(B)\varepsilon}{6Jw_m w_l} + \frac{\rho_2(B)\varepsilon}{6Jw_m w_l} + 2\varepsilon/3 \leq \varepsilon,$$

which proves (4). The statement (5) can be proven analogously. ■

Remark 6 *The triangle inequality holds for the distributional distance $d(\cdot,\cdot)$ and its empirical estimates $\hat{d}(\cdot,\cdot)$ so that for all distributions $\rho_i,\ i = 1..3$ and all sequences $\mathbf{x}_i \in \mathcal{X}^{n_i}$ $n_i \in \mathbb{N},\ i = 1..3$ we have,*

$$\begin{aligned} d(\rho_1, \rho_2) &\leq d(\rho_1, \rho_3) + d(\rho_2, \rho_3) \\ \hat{d}(\mathbf{x}_1, \mathbf{x}_2) &\leq \hat{d}(\mathbf{x}_1, \mathbf{x}_3) + \hat{d}(\mathbf{x}_2, \mathbf{x}_3) \\ \hat{d}(\mathbf{x}_1, \rho_1) &\leq \hat{d}(\mathbf{x}_1, \rho_2) + d(\rho_1, \rho_2). \end{aligned}$$

3. Clustering protocol

We start with a common general probabilistic model for both problems, which we use to formulate each of the two settings (offline and online) separately.

Consider the matrix $\mathbf{X} \in (\mathcal{X}^\infty)^\infty$ of random variables

$$\mathbf{X} := \begin{bmatrix} X_1^1 & X_2^1 & X_3^1 & \dots \\ X_1^2 & X_2^2 & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \in (\mathcal{X}^\infty)^\infty \tag{7}$$

generated by some probability distribution P on $((\mathcal{X}^\infty)^\infty, \mathcal{B}_2)$. Assume that the marginal distribution of P on each row of $\mathbf{X}$ is one of κ *unknown stationary ergodic* process distributions $\rho_1, \rho_2, \dots, \rho_\kappa$. Thus, the matrix $\mathbf{X}$ corresponds to infinitely many one-way infinite sequences, each of which is generated by a stationary ergodic distribution. Aside from this assumption, 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 $\rho_k, k = 1..\kappa$ are ordered in the order of appearance of their first samples in $\mathbf{X}$.

Of the many ways a set of κ disjoint subsets of the rows of $\mathbf{X}$ may be produced, the most natural partitioning in this formulation is to group together those and only those rows of $\mathbf{X}$ for which the marginal distribution is the same. More precisely, we define the ground-truth partitioning of $\mathbf{X}$ as follows.

Definition 7 (Ground-truth $\mathcal{G}$) *Let*

$$\mathcal{G} = \{\mathcal{G}_1, \dots, \mathcal{G}_\kappa\}$$

be a partitioning of $\mathbb{N}$ into κ disjoint subsets $\mathcal{G}_k,\ k = 1..\kappa$, such that $\mathbf{x}_i,\ i \in \mathbb{N}$ is generated by ρ_k for some $k \in 1..\kappa$ if and only if $i \in \mathcal{G}_k$. Call $\mathcal{G}$ the ground-truth clustering. Introduce also the notation $\mathcal{G}|_N$ for the restriction of $\mathcal{G}$ to the first N sequences:

$$\mathcal{G}|_N := \{\mathcal{G}_k \cap \{1..N\} : k = 1..\kappa\}.$$

Offline Setting. The problem may be formulated as follows. We are given a finite set $S := \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ of N samples, for some fixed $N \in \mathbb{N}$. Each sample is generated by one of κ unknown stationary ergodic process distributions $\rho_1, \dots, \rho_\kappa$. More specifically, the set S is obtained from $\mathbf{X}$ as follows. Some (arbitrary) lengths $n_i \in \mathbb{N},\ i \in 1..N$ are fixed, and $\mathbf{x}_i$ for each $i = 1..N$ is defined as $\mathbf{x}_i := X^i_{1..n_i}$.

A clustering function f takes a finite set $S := \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ of samples and an optional parameter κ (the number of target clusters) to produce a partition $f(S, \kappa) := \{C_1, \dots, C_\kappa\}$ of the index-set $\{1..N\}$. The goal is to partition $\{1..N\}$ so as to recover the ground-truth clustering $\mathcal{G}$. We call a clustering algorithm asymptotically consistent if it achieves this goal for long enough sequences $\mathbf{x}_i,\ i = 1..N$ in S:

Definition 8 (Asymptotic consistency: offline setting) *A clustering function f is strongly asymptotically consistent in the offline sense if with probability* 1 *we have*

$$\lim_{n\to\infty} f(S, \kappa) = \mathcal{G}|_N,$$

where $n := \min\{n_1, \dots, n_N\}$. It is weakly asymptotically consistent if

$$\lim_{n\to\infty} P(f(S,\kappa) = \mathcal{G}|_N) = 1.$$

Note that the consistency is asymptotic with respect to the minimum sample length n, and not with respect to the number N of samples.

When considering the offline setting with known number of clusters κ we assume that N is such that $\{1..N\} \cap G_k \neq \varnothing$ for every $k = 1..\kappa$ (that is, all κ clusters are represented); otherwise we could just take a smaller κ).

Online Setting. The online problem may be formulated analogously as follows. Consider the infinite matrix $\mathbf{X}$ given by (7). 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.

$$S(t) = \{\mathbf{x}_1^t, \cdots \mathbf{x}_{N(t)}^t\} \text{ where } \mathbf{x}_i^t := X^i_{1..n_i(t)}.$$

This setting is depicted in Figure 3. We assume that the number of samples, as well as the individual sample-lengths grow with time. That is, the length $n_i(t)$ of each sequence $\mathbf{x}_i$ is non-decreasing and grows to infinity (as a function of time t). The number of sequences $N(t)$ also grows to infinity. Aside from these assumptions, the functions $N(t)$ and $n_i(t)$ are completely arbitrary.

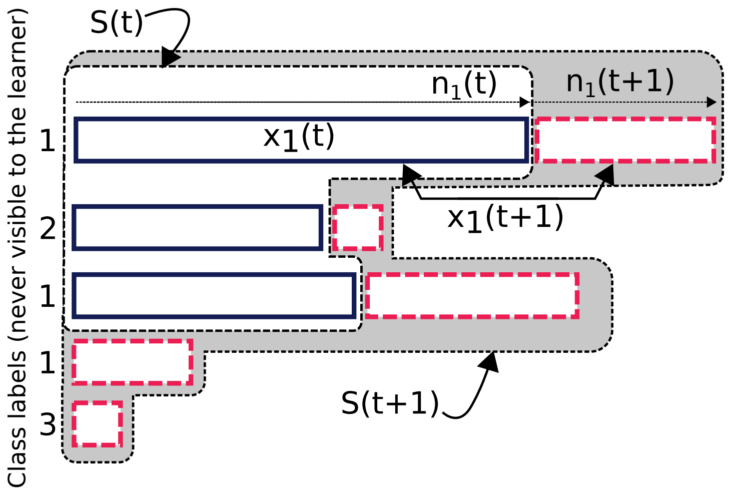

Figure 1: Online Protocol: solid rectangles correspond to sequences observed at time t, dashed rectangles correspond to segments arrived at time $t+1$.

An online clustering function is strongly 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 given by Definition 7).

Definition 9 (Asymptotic consistency: online setting) *A clustering function is strongly (weakly) asymptotically consistent in the online sense, if for every $N \in \mathbb{N}$ the clustering*

$f(S(t),\kappa)|_N$ is (weakly) asymptotically consistent in the offline sense, where $f(S(t),\kappa)|_N$ is the clustering $f(S(t),\kappa)$ restricted to the first N sequences:

$$f(S(t),\kappa)|_N := \{f(S(t),\kappa)\cap\{1..N\}, k=1..\kappa\}.$$

Remark 10 *Note that even if the* eventual *number of different time series distributions producing the sequences is κ (that is, the number of clusters in the ground-truth clustering $\mathcal{G}$) is known, the number of observed distributions at each individual time-step is unknown. In particular, it may be possible that at a given time-step t we have $\{1..N(t)\}\cap\mathcal{G}_k=\varnothing$ for some $k\in 1..\kappa$.*

Known and unknown κ. Under the general framework (for both the offline and the online problems) described above, consistent clustering with unknown number of clusters is impossible. This follows from an impossibility result by (Ryabko, 2010c) which states that when we have only two (binary-valued) samples, generated (independently) by two stationary ergodic distributions, it is impossible to decide whether they have been generated by the same or by different distributions, even in the sense of weak asymptotic consistency. (This holds even if the distributions come from a smaller class: the set of all B-processes.) Therefore, if the number of clusters is unknown, we are bound to make stronger assumptions. Since our main interest in this paper is to develop consistent clustering algorithms under the general framework described above, for the most part of this paper we assume that the correct number κ of clusters is known. However, in Section 4.3 we also show that under certain mixing conditions on the process distributions that generate the data, it is possible to have consistent algorithms that address the case where κ is unknown.

4. Main Theoretical Results

In this section we present our clustering methods for both the offline and the online settings. The main results, presented in Sections 4.1 (offline) and 4.2 (online), concern the case where the number of clusters, κ, is known. In Section 4.3 we show that, given the mixing rates of the process distributions that generate the data, it is possible to find the correct number of clusters κ in the offline setting. The extension of this result to the online setting is left as future work.

4.1 Offline Setting

Given that we have asymptotically consistent estimates $\hat{d}$ of the distributional distance d, it is relatively simple to construct asymptotically consistent clustering algorithms for the offline setting. This follows from the fact that, since $\hat{d}(\cdot,\cdot)$ converges to $d(\cdot,\cdot)$, for large enough sequence lengths n, the points $\mathbf{x}_1,\dots,\mathbf{x}_N$ have the so-called strict separation property: the points within each target cluster are closer to each other than to points in any other cluster (on strict separation in clustering see, for example, Balcan et al., 2008). This makes many simple algorithms, such as, single or average linkage, or the k-means algorithm with certain initializations, provably consistent.

We present below one specific algorithm that we show to be asymptotically consistent in the general framework introduced. What makes this simple algorithm interesting is that it requires only κN distance calculations. In short, Algorithm 1 initializes the clusters using farthest-point initialization, and then assigns each remaining point to the nearest cluster. More precisely, the sample $\mathbf{x}_1$ is assigned as the first cluster centre. Then a sample is found that is farthest away from $\mathbf{x}_1$ in the empirical distributional distance $\hat{d}$ and is assigned as the second cluster centre. For each $k = 2..\kappa$ the k^{th} cluster center is sought as the sequence with the largest minimum distance from the already assigned cluster centers for $1..k-1$. By the last iteration we have κ cluster centers. Next the remaining samples are each assigned to the closest cluster.

Algorithm 1 Offline clustering method of (Ryabko, 2010b)

```
1: INPUT: sequences $S := \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, Number $\kappa$ of clusters
2:
3: Initialize $\kappa$-farthest points as cluster-centres:
4: $c_1 \leftarrow 1$
5: $C_1 \leftarrow \{c_1\}$
6: for $k = 2..\kappa$ do
7:     $c_k \leftarrow \operatorname{argmax}_{i=1..N} \min_{j=1..k-1} \hat{d}(\mathbf{x}_i, \mathbf{x}_{c_j})$, where ties are broken arbitrarily
8:     $C_k \leftarrow \{c_k\}$
9: end for
10: Assign the remaining points to closest centres:
11: for $i = 1..N$ do
12:     $k \leftarrow \operatorname{argmin}_{j \in \bigcup_{k=1}^{\kappa} C_k} \hat{d}(\mathbf{x}_i, \mathbf{x}_j)$
13:     $C_k \leftarrow C_k \cup \{i\}$
14: end for
15: OUTPUT: clusters $C_1, C_2, \cdots, C_\kappa$
```

Theorem 11 *Algorithm 1 is strongly asymptotically consistent (in the offline sense of Definition 8), provided the correct number κ of clusters is known, and the marginal distribution of each sequence $\mathbf{x}_i, i = 1..N$ is stationary ergodic.*

Proof To prove the consistency statement we use Lemma 5 to show that if the samples in S are long enough, the samples that are generated by the same process distribution are closer to each other than to the rest of the samples. Therefore, the samples chosen as cluster centers are each generated by a different process distribution, and since the algorithm assigns the rest of the samples to the closest clusters, the statement follows. More formally, let n denote the shortest sample length in S, i.e.

$$n_{\min} := \min_{i \in 1..N} n_i.$$

Denote by δ the minimum non-zero distance between the process distributions i.e.,

$$\delta := \min_{k \neq k' \in 1..\kappa} \hat{d}(\rho_k, \rho_{k'}).$$

Fix $\varepsilon \in (0, \delta/4)$. Since there are a finite number N of samples, by Lemma 5 for all large enough $n_{\min}$ we have

$$\sup_{\substack{k\in 1..\kappa \\ i\in \mathcal{G}_k\cap\{1..N\}}} \hat{d}(\mathbf{x}_i, \rho_k) \leq \varepsilon. \tag{8}$$

where $\mathcal{G}_k,\ k = 1..\kappa$ denote the ground-truth partitions given by Definition 7. By (8) and applying the triangle inequality we obtain

$$\sup_{\substack{k\in 1..\kappa \\ i,j\in \mathcal{G}_k\cap\{1..N\}}} \hat{d}(\mathbf{x}_i, \mathbf{x}_j) \leq 2\varepsilon. \tag{9}$$

Thus, for all large enough $n_{\min}$ we have

$$\begin{aligned} \inf_{\substack{i\in\mathcal{G}_k\cap\{1..N\} \\ j\in\mathcal{G}_{k'}\cap\{1..N\} \\ k\neq k'\in 1..\kappa}} \hat{d}(\mathbf{x}_i, \mathbf{x}_j) &\geq \inf_{\substack{i\in\mathcal{G}_k\cap\{1..N\} \\ j\in\mathcal{G}_{k'}\cap\{1..N\} \\ k\neq k'\in 1..\kappa}} d(\rho_k, \rho_{k'}) - \hat{d}(\mathbf{x}_i, \rho_k) - \hat{d}(\mathbf{x}_j, \rho_{k'}) \\ &\geq \delta - 2\varepsilon \end{aligned} \tag{10}$$

where the first inequality follows from the triangle inequality, and the second inequality follows from (8) and the definition of δ. In words, (9) and (10) mean that the samples in $\mathcal{S}$ that are generated by the same process distribution are closer to each other than to the rest of the samples. Finally, for all $n_{\min}$ large enough to have (9) and (10) we obtain

$$\max_{i=1..N} \min_{k=1..\kappa-1} \hat{d}(\mathbf{x}_i, \mathbf{x}_{c_k}) \geq \delta - 2\varepsilon$$

where as specified by Algorithm 1, $c_1 := 1$ and $c_k := \operatorname*{argmax}_{i=1..N} \min_{j=1..k-1} \hat{d}(\mathbf{x}_i, \mathbf{x}_{c_j}),\ k = 2..\kappa$. Hence, the indices $c_1, \dots, c_\kappa$ will be chosen to index sequences generated by different process distributions, and the consistency statement follows. ■

4.2 Online setting

The online version of the problem turns out to be more complicated than the offline formulation. The problem is that, since new sequences arrive (potentially) at every time-step, we can never rely on the distance estimates corresponding to all of the observed samples to be correct. Thus, as mentioned in the introduction, the main challenge can be identified with what we regard as "bad" sequences: recently-observed sequences, for which sufficient information has not yet been collected, and for which the estimates of the distance (with respect to any other sequence) are bound to be misleading. Thus, in particular, farthest-point initialization would not work in this case. More generally, using any batch algorithm on all available data at every time-step results in not only mis-clustering "bad" sequences, but also in clustering incorrectly those for which sufficient data are already available.

The solution, realized in Algorithm 2, is based on a *weighted combination* of several clusterings, each obtained by running the offline algorithm (Algorithm 1) on different portions of data. The partitions are combined with weights that depend on the batch size and on the minimum inter-cluster distance. This last step of combining multiple clusterings with weights may be reminiscent of prediction with expert advice (e.g., Cesa-Bianchi and Lugosi, 2006), where experts are combined based on their past performance. However, the difference here is that the performance of each clustering cannot be measured directly in our setting.

Algorithm 2 Online Clustering

```
 1: INPUT: Number $\kappa$ of target clusters
 2: for $t = 1..\infty$ do
 3:     Obtain new sequences $S(t) = \{\mathbf{x}_1^t, \cdots, \mathbf{x}_{N(t)}^t\}$
 4:     Initialize the normalization factor: $\eta \leftarrow 0$
 5:     Initialize the final clusters: $C_k(t) \leftarrow \varnothing,\ k = 1..\kappa$
 6:     Generate $N(t) - \kappa + 1$ candidate cluster centers:
 7:     for $j = \kappa..N(t)$ do
 8:         $\{C_1^j, \ldots, C_\kappa^j\} \leftarrow \text{Alg1}(\{\mathbf{x}_1^t, \cdots, \mathbf{x}_j^t\}, \kappa)$
 9:         $\mu_k \leftarrow \min\{i \in C_k^j\}, k = 1..\kappa$        ▷ Set the smallest index as cluster center.
10:         $(c_1^j, \ldots, c_\kappa^j) \leftarrow \text{sort}(\mu_1, \ldots, \mu_\kappa)$        ▷ Sort the cluster centers increasingly.
11:         $\gamma_j \leftarrow \min_{k \neq k' \in 1..\kappa} \hat{d}(\mathbf{x}_{c_k^j}^t, \mathbf{x}_{c_{k'}^j}^t)$        ▷ Calculate performance score.
12:         $w_j \leftarrow j^{-2}$
13:         $\eta \leftarrow \eta + w_j \gamma_j$
14:     end for
15:     Assign points to clusters:
16:     for $i = 1..N(t)$ do
17:         $k \leftarrow \text{argmin}_{k' \in 1..\kappa} \frac{1}{\eta} \sum_{j=1}^{N(t)} w_j \gamma_j \hat{d}(\mathbf{x}_i^t, \mathbf{x}_{c_{k'}^j}^t)$
18:         $C_k(t) \leftarrow C_k(t) \cup \{i\}$
19:     end for
20:     OUTPUT: $\{C_1(t), \cdots, C_k(t)\}$
21: end for
```

More precisely, Algorithm 2 works as follows. Given a set $S(t)$ of $N(t)$ samples, the algorithm iterates over $j := \kappa, \ldots, N(t)$ where at each iteration Algorithm 1 is used to cluster the first j sequences $\{\mathbf{x}_1^t, \ldots, \mathbf{x}_j^t\}$ into κ clusters. In each cluster the sequence with the smallest index is assigned as the candidate cluster center. A performance score γ_j is calculated as the minimum distance $\hat{d}$ between the κ candidate cluster centers obtained at iteration j. Thus, γ_j is an estimate of the minimum inter-cluster distance. At this point we have $N(t) - \kappa + 1$ sets of κ cluster centers $c_1^j, \ldots, c_\kappa^j,\ j = 1..N(t) - \kappa + 1$. Next, every sample $\mathbf{x}_i^t,\ i = 1..N(t)$ is assigned to the *closest* cluster. This is determined by minimizing the weighted combination of the distances between $\mathbf{x}_i^t$ and the candidate cluster centers obtained at each iteration on j. More specifically, for each $i = 1..N(t)$ the sequence $\mathbf{x}_i^t$ is assigned to the cluster k, where k is defined as

$$k := \text{argmin}_{k=1..\kappa} \sum_{j=\kappa}^{N(t)} w_j \gamma_j \hat{d}(\mathbf{x}_i^t, \mathbf{x}_{c_k^j}^t).$$

Theorem 12 *Algorithm 2 is strongly asymptotically consistent (in the online sense of Definition 9), provided the correct number of clusters κ of clusters is known, and the marginal distribution of each sequence $\mathbf{x}_i, i \in \mathbb{N}$ is stationary ergodic.*

Before giving the proof of Theorem 12, we provide an intuitive explanation as to how Algorithm 2 works. First consider the following simple solution. Take some fixed (reference) portion of the samples, run the batch algorithm on it, and then simply assign every remaining sequence to the nearest cluster. Since the offline algorithm is asymptotically consistent, this procedure would be asymptotically consistent as well, but only if we were knew that the selected portion of the sequences contains at least one sequence sampled from each and every one of the κ distributions. However, there is no way to find a fixed (not growing with time) portion of data that would be guaranteed to contain a representative of each cluster (that is, of each time series distribution). Allowing such a reference set of sequences to grow with time would guarantee that eventually it contains representatives of all clusters, but it would break the consistency guarantee for the reference set, since it effectively returns us back to the original online clustering problem.

A key observation we make to circumvent this problem is the following. If, for some $j \in \{\kappa, \dots, N(t)\}$, each sample in the batch $\{\mathbf{x}_1^t, \dots, \mathbf{x}_j^t\}$ is generated by *at most* $\kappa - 1$ process distributions, any partitioning of this batch into κ sets results in a minimum inter-cluster distance γ_j that, as follows from the asymptotic consistency of $\hat{d}$, converges to 0. On the other hand, if the set of samples $\{\mathbf{x}_1^t, \dots, \mathbf{x}_j^t\}$ contains sequences generated by all κ process distributions, γ_j converges to a non-zero constant, namely, the minimum distance between the distinct process distributions $\rho_1, \dots, \rho_\kappa$. In this case, as follows from the consistency of Algorithm 1, from some time on, the batch $\{\mathbf{x}_1^t, \dots, \mathbf{x}_j^t\}$ will be clustered correctly. Thus, instead of selecting one reference batch of sequences and constructing a set of clusters based on those, we consider all batches of sequences for $j = \kappa..N(t)$, and combine them with weights. Two sets of weights are involved in this step: γ_j and w_j, where

1. γ_j is used to penalize for small inter-cluster distance, canceling the clustering results produced based on sets of sequences generated by less than κ distributions;

2. w_j is used 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.

It remains to use the consistency of $\hat{d}$ and that of Algorithm 1 to see that every finite number $N \in \mathbb{N}$ of points are from some time on assigned to the correct target cluster
Proof [of Theorem 12] First, we show that for every $k \in 1..\kappa$ we have

$$\frac{1}{\eta}\sum_{j=1}^{N(t)} w_j \gamma_j^t \hat{d}(\mathbf{x}_{c_k^j}^t, \rho_k) \to 0 \text{ a.s.} \tag{11}$$

Denote by δ the minimum non-zero distance between the process distributions i.e.,

$$\delta := \min_{k \neq k' \in 1..\kappa} \hat{d}(\rho_k, \rho_{k'}). \tag{12}$$

Fix $\varepsilon \in (0, \delta/4)$. We can find an index J such that $\sum_{j=J}^{\infty} w_j \leq \varepsilon$. Let $S(t)|_j = \{\mathbf{x}_1^t, \cdots, \mathbf{x}_j^t\}$ denote the subset of $S(t)$ consisting of the first j sequences for $j \in 1..N(t)$. For $k = 1..\kappa$ let

$$s_k := \min\{i \in \mathcal{G}_k \cap 1..N(t)\} \tag{13}$$

index the first sequence in $S(t)$ that is generated by ρ_k where $\mathcal{G}_k,\ k = 1..\kappa$ are the ground-truth partitions given by Definition 7. Define

$$m := \max_{k \in 1..\kappa} s_k. \tag{14}$$

Recall that the sequence lengths $n_i(t)$ grow with time. Therefore, by the consistency of $\hat{d}$, i.e. Lemma 5 for every fixed $j \in 1..J$ there exists some $T_1(j)$ such that for all $t \geq T_1(j)$ we have

$$\sup_{\substack{k \in 1..\kappa \\ i \in \mathcal{G}_k \cap \{1..j\}}} \hat{d}(\mathbf{x}_i^t, \rho_k) \leq \varepsilon. \tag{15}$$

Moreover, by Theorem 11 for every $j \in m..J$ there exists some $T_2(j)$ such that $\text{Alg1}(S(t)|_j, \kappa)$ is consistent for all $t \geq T_2(j)$. Let

$$T := \max_{\substack{i=1,2 \\ j \in 1..J}} T_i(j)$$

Recall that by definition (i.e. 14) $S(t)|_m$ contains samples from all κ distributions. Therefore, for all $t \geq T$

$$\begin{aligned}\inf_{k \neq k' \in 1..\kappa} \hat{d}(\mathbf{x}_{c_k^m}^t, \mathbf{x}_{c_{k'}^m}^t) &\geq \inf_{k \neq k' \in 1..\kappa} d(\rho_k, \rho_{k'}) - \sup_{k \neq k' \in 1..\kappa} (\hat{d}(\mathbf{x}_{c_k^m}^t, \rho_k) + \hat{d}(\mathbf{x}_{c_{k'}^m}^t, \rho_{k'})) \\ &\geq \delta - 2\varepsilon \geq \delta/2. \end{aligned} \tag{16}$$

where the first inequality follows from the triangle inequality and the second inequality follows from the consistency of $\text{Alg1}(S(t)|_m, \kappa)$ for $t \geq T$, the definition of δ given by (12) and the assumption that $\varepsilon \in (0, \delta/4)$. Recall that (as specified in Algorithm 2) we have $\eta = \sum_{j=1}^{N(t)} w_j \gamma_j^t$ Hence, by (16) for all $t \geq T$ we have

$$\eta \geq w_m \delta/2. \tag{17}$$

By (17) and noting that by definition $\hat{d}(\cdot, \cdot) \leq 1$ for every $k \in 1..\kappa$ we obtain,

$$\frac{1}{\eta} \sum_{j=1}^{N(t)} w_j \gamma_j^t \hat{d}(\mathbf{x}_{c_k^j}^t, \rho_k) \leq \frac{1}{\eta} \sum_{j=1}^{J} w_j \gamma_j^t \hat{d}(\mathbf{x}_{c_k^j}^t, \rho_k) + \frac{2\varepsilon}{w_m \delta}. \tag{18}$$

On the other hand by definition (i.e. (14)) the sequences in $S(t)|_j$ for $j = 1..m-1$ are generated by *at most* $\kappa - 1$ out of the κ process distributions. Therefore, at every iteration on $j \in 1..m-1$ there exists at least one pair of distinct cluster centers that are generated by *the same* process distribution. Therefore, by (15) and (17), for all $t \geq T$ and every $k \in 1..\kappa$ we have,

$$\frac{1}{\eta} \sum_{j=1}^{m-1} w_j \gamma_j^t \hat{d}(\mathbf{x}_{c_k^j}^t, \rho_i) \leq \frac{1}{\eta} \sum_{j=1}^{m-1} w_j \gamma_j^t \leq \frac{2\varepsilon}{w_m \delta}. \tag{19}$$

Noting that the clusters are ordered in the order of appearance of the distributions, we have $\mathbf{x}^t_{c^j_k} = \mathbf{x}^t_{s_k}$ for all $j = m..J$ and $k = 1..\kappa$, where the index s_k is defined by (13). Therefore, by (15) for all $t \geq T$ and every $k = 1..\kappa$ we have

$$\frac{1}{\eta}\sum_{j=m}^{J} w_j \gamma^t_j \hat{d}(\mathbf{x}^t_{c^j_k}, \rho_k) = \frac{1}{\eta}\hat{d}(\mathbf{x}^t_{s_k}, \rho_k)\sum_{j=m}^{J} w_j \gamma^t_j \leq \varepsilon. \tag{20}$$

Combining (18), (19), and (20) we obtain

$$\frac{1}{\eta}\sum_{j=1}^{N(t)} w_j \gamma^t_j \hat{d}(\mathbf{x}^t_{c^j_k}, \rho_k) \leq \varepsilon(1 + \frac{4}{w_m \delta}). \tag{21}$$

for all $k = 1..\kappa$ and all $t \geq T$ (establishing 11).

To finish the proof of the consistency consider an index $i \in \mathcal{G}_r$ for some $r \in 1..\kappa$. By Lemma 5, increasing T if necessary, for all $t \geq T$ we have

$$\sup_{\substack{k \in 1..\kappa \\ i \in \mathcal{G}_k \cap 1..N}} \hat{d}(\mathbf{x}^t_i, \rho_k) \leq \varepsilon. \tag{22}$$

For all $t \geq T$ and all $k \neq r \in 1..\kappa$ we have,

$$\begin{aligned}\frac{1}{\eta}\sum_{j=1}^{N(t)} w_j \gamma^t_j \hat{d}(\mathbf{x}^t_i, \mathbf{x}^t_{c^j_k}) &\geq \frac{1}{\eta}\sum_{j=1}^{N(t)} w_j \gamma^t_j \hat{d}(\mathbf{x}^t_i, \rho_k) - \frac{1}{\eta}\sum_{j=1}^{N(t)} w_j \gamma^t_j \hat{d}(\mathbf{x}^t_{c^j_k}, \rho_k)\\ &\geq \frac{1}{\eta}\sum_{j=1}^{N(t)} w_j \gamma^t_j (\hat{d}(\rho_k, \rho_r) - \hat{d}(\mathbf{x}^t_i, \rho_r)) - \frac{1}{\eta}\sum_{j=1}^{N(t)} w_j \gamma^t_j \hat{d}(\mathbf{x}^t_{c^j_k}, \rho_k)\\ &\geq \delta - 2\varepsilon(1 + \frac{2}{w_m \delta}),\end{aligned} \tag{23}$$

where the first and second inequalities follow from subsequent application of the triangle inequality, and the last inequality follows from (22), (21) and the definition of δ. Since the choice of ε is arbitrary, from (22) and (23) we obtain

$$\operatorname*{argmin}_{k \in 1..\kappa} \frac{1}{\eta}\sum_{j=1}^{N(t)} w_j \gamma^t_j \hat{d}(\mathbf{x}^t_i, \mathbf{x}^t_{c^j_k}) = r. \tag{24}$$

It remains to note that for any fixed $N \in \mathbb{N}$ from some t on (24) holds for all $i = 1..N$, and the consistency statement follows. ■

4.3 Unknown number κ of clusters

So far we have shown that when κ is known in advance, consistent clustering is possible under the only assumption that the samples are generated by unknown stationary ergodic process distributions, while their joint distribution can be arbitrary. However, as follows from the theoretical impossibility results of (Ryabko, 2010c) discussed in Section 3, the correct number κ of clusters is not possible to be estimated with no further assumptions or additional constraints. One way to overcome this obstacle is to assume known rates of convergence of frequencies to the corresponding probabilities. Such rates are provided by assumptions on the mixing rates of the process distributions that generate the data. Here we will show that under rather mild assumptions on the mixing rates (and still without any modeling or independence assumptions), consistent clustering is possible when the number of clusters is unknown.

In this section we assume that the samples are $[0,1]$-valued; extension to arbitrary bounded (multidimensional) ranges is straightforward. We introduce *mixing coefficients*, mainly following (Bosq, 1996) in formulations. Informally, mixing coefficients of a stochastic process measure how fast the process forgets about its past. Any one-way infinite stationary process $X_1, X_2, \dots$ can be extended backwards to make a two-way infinite process $\dots, X_{-1}, X_0, X_1, \dots$ with the same distribution. In the definition below we assume such an extension. Define the α mixing coefficients as

$$\alpha(n) = \sup_{A \in \sigma(\dots, X_{-1}, X_0), B \in \sigma(X_n, X_{n+1}, \dots))} |P(A \cap B) - P(A)P(B)|, \tag{25}$$

where $\sigma(..)$ stands for the sigma-algebra generated by random variables in brackets. These coefficients are non-increasing. A process is called strongly *α-mixing* if $\alpha(n) \to 0$. Many important classes of processes satisfy the mixing conditions. For example, if a process is a stationary irreducible aperiodic Hidden Markov process, then it is α-mixing. If the underlying Markov chain is finite-state, then the coefficients decrease exponentially fast. Other probabilistic assumptions can be used to obtain bounds on the mixing coefficients, see, e.g., (Bradley, 2005) and references therein.

We consider the clustering problem in the offline setting. Extension to the online problem is left as future work. The method that we propose, namely Algorithm 3 is very simple. Its inputs are: samples $S := \{\mathbf{x}_1, \dots, x_N\}$, the threshold level $\delta \in (0,1)$ and the parameters $m, l \in \mathbb{N}$, $B^{m,l,n}$. The algorithm assigns to the same cluster all samples which are at most δ-far from each other, as measured by $\hat{d}$. We do not give a pseudo code implementation of this algorithm, since it is rather clear.

The idea is that the threshold level δ is selected according to the minimum length of a sample and the (known bounds on) mixing rates of the process ρ generating the samples (see Theorem 13). As we show in Theorem 13, if the distribution of the samples satisfies $\alpha(n) \le \alpha_n \to 0$, where α_n are known, then one can select (based on α_n only) the parameters of Algorithm 3 in such a way that it is weakly asymptotically consistent. Moreover, a bound on the probability of error before asymptotic is provided.

Theorem 13 (Algorithm 3 is consistent for unknown k) *Fix sequences $\alpha_n \in (0,1)$, $m_n, l_n, b_n \in \mathbb{N}$, and let, for each $m, l \in \mathbb{N}$, $B^{m,l,n} \subset B^{m,l}$ be an increasing sequence of finite sets such that $\cup_{n \in \mathbb{N}} B^{m,l,n} = B^{m,l}$. Set $b_n := \max_{l \le l_n, m \le m_n} |B^{m,l,n}|$. Let also $\delta_n \in (0,1)$. Let $N \in \mathbb{N}$ and suppose that the samples $\mathbf{x}_1, \dots, \mathbf{x}_N$ are generated in such a way that the (unknown marginal) distributions ρ_i, $i = 1..\kappa$ are stationary ergodic and satisfy $\alpha_n(\rho) \le \alpha_n$, for all $n \in \mathbb{N}$. Then for every sequence $q_n \in [0..n/2]$, Algorithm 3 satisfies*

$$P(T \ne \mathcal{G}|_N) \le 2N(N+1)(m_n l_n b_n \gamma_n(\delta_n) + \gamma_n(\varepsilon_\rho)) \tag{26}$$

where

$$\gamma(\delta) := (2e^{-q_n\delta^2/32} + 11(1+4/\delta)^{1/2} q_n \alpha_{(n-2m_n)/2q_n}),$$

T is the partition output by the algorithm, $G|_N$ is the ground-truth clustering, ε_ρ is a constant that depends only on ρ, and $n = \min_{i=1..N} n_i$. In particular, if $\alpha_n = o(1)$, then, selecting the parameters in such a way that $\delta_n = o(1)$, $q_n, m_n, l_n, b_n = o(n)$, $q_n, m_n, l_n \to \infty$, $\cup_{k\in\mathbb{N}} B^{m,l,k} = B^{m,l}$, $b_n^{m,l} \to \infty$, for all $m, l \in \mathbb{N}$, and, finally, $m_n l_n b_n \gamma(\delta) = o(1)$, as is always possible, Algorithm 3 is weakly asymptotically consistent *(with the number of clusters k unknown).*

Proof We use the following bound from (Bosq, 1996): for any zero-mean random process $Y_1, Y_2, \dots$, every $n \in \mathbb{N}$ and every $q \in [1..n/2]$ we have

$$P\left(|\sum_{i=1}^{n} Y_i| > n\varepsilon\right) \le 4\exp(-q\varepsilon^2/8) + 22(1+4/\varepsilon)^{1/2} q\alpha(n/2q).$$

For every $j = 1..N$, every $m < n$, $l \in \mathbb{N}$, and $B \in B^{m,l}$, define the processes $Y_1^j, Y_2^j, \dots$, where

$$Y_t^j := \mathbb{I}\{(X_t^j, \dots, X_{t+m-1}^j) \in B\} - \rho_k(X_{1..m}^j \in B)$$

and ρ_k is the marginal distribution of X^j (that is, k is such that $j \in \mathcal{G}_k$). It is easy to see that α-mixing coefficients for this process satisfy $\alpha(n) \le \alpha_{n-2m}$. Thus,

$$P(|\nu(X_{1..n_j}^j, B) - \rho_k(X_{1..m}^j \in B)| > \varepsilon/2) \le \gamma_n(\varepsilon) \tag{27}$$

Then for every $i, j \in \mathcal{G}_k \cap 1..N$ for some $k \in 1..\kappa$ (that is, $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same ground-truth cluster)

$$P(|\nu(X_{1..n_i}^i, B) - \nu(X_{1..n_j}^j, B)| > \varepsilon) \le 2\gamma_n(\varepsilon).$$

Using the union bound, summing over m, l, and B, we obtain

$$P(\hat{d}(\mathbf{x}_i, \mathbf{x}_j) > \varepsilon) \le 2m_n l_n b_n \gamma_n(\varepsilon). \tag{28}$$

Next let $i \in \mathcal{G}_k \cap 1..N$ and $j \in \mathcal{G}_{k'} \cap 1..N$ for $k \ne k' \in 1..\kappa$ (i.e., $\mathbf{x}_i, \mathbf{x}_j$ are in two different target clusters). Then, for some $m_{i,j}, l_{i,j} \in \mathbb{N}$ there is $B_{i,j} \in B^{m_{i,j}, l_{i,j}}$ such that for some $\tau_{i,j} > 0$ we have

$$|\rho_k(X_{1..|B_{i,j}|}^i \in B_{i,j}) - \rho_{k'}(X_{1..|B_{i,j}|}^j \in B_{i,j})| > 2\tau_{i,j}.$$

Then for every $\varepsilon < \tau_{i,j}/2$ we have

$$P(|\nu(X_{1..n_i}^i, B_{i,j}) - \nu(X_{1..n_j}^j, B_{i,j})| < \varepsilon) \le 2\gamma_n(\tau_{i,j}). \tag{29}$$

Moreover, for $\varepsilon < w_{m_{i,j}} w_{l_{i,j}} \tau_{i,j}/2$ we have

$$\rho(\hat{d}(\mathbf{x}_i, \mathbf{x}_j) > \varepsilon) \le 2\gamma_n(w_{m_{i,j}} w_{l_{i,j}} \tau_{i,j}). \tag{30}$$

Define

$$\varepsilon_\rho := \min_{\substack{k\neq k'\in 1..\kappa\\ i\in\mathcal{G}_k\cap 1..N\\ j\in\mathcal{G}_{k'}\cap 1..N}} w_{m_{i,j}} w_{l_{i,j}} \tau_{i,j}/2.$$

Clearly, from this and (29), for every $\varepsilon < 2\varepsilon_\rho$ we obtain

$$\rho(\hat{d}(\mathbf{x}_i,\mathbf{x}_j) > \varepsilon) \leq 2\gamma_n(\varepsilon_\rho). \tag{31}$$

Algorithm 3 produces correct results if for every pair i, j we have $\hat{d}(\mathbf{x}_i,\mathbf{x}_j) < \delta_n$ if and only if $i, j \in \mathcal{G}_k \cap 1..N$ for some $k \in 1..\kappa$. Therefore, taking the bounds (28) and (31) together for each of the $N(N+1)/2$ pairs of samples, we obtain (26). ■

The purpose of the theorem above is to demonstrate that asymptotic consistency is achievable when the number of clusters is unknown, under some non-parametric assumptions on the time series distribution. While it provides bound on the error of the algorithm, we do not attempt to establish tight bounds for the set of assumptions made.

5. Computational considerations

In this section we show that all the proposed methods are efficiently computable. This claim is further illustrated by the experimental results in the next section. Note, however, that since the results presented are asymptotic, the question of what is the best achievable computational complexity of an algorithm that still has the same asymptotic performance guarantees is meaningless: for example, an algorithm could through away most of the data can still be asymptotically consistent. This is why we do not attempt to find a resource-optimal way of computing the methods presented.

First, we show that **calculating $\hat{d}$ is at most quadratic (up to log terms), and quasilinear if we use $m_n = \log n$.** Let us first show that calculating $\hat{d}$ is fully tractable with $m_n, l_n \equiv \infty$. First, observe that for fixed m and l, the sum

$$T^{m,l} := \sum_{B\in B^{m,l}} |\nu(X^1_{1..n_1}, B) - \nu(X^2_{1..n_2}, B)| \tag{32}$$

has not more than $n_1 + n_2 - 2m + 2$ non-zero terms (assuming $m \leq n_1, n_2$; the other case is obvious). Indeed, there are $n_i - m + 1$ tuples of size m in each sequence $\mathbf{x}_i,\ i = 1, 2$ namely, $X^i_{1..m}, X^i_{2..m+1}, \dots, X^i_{n_1-m+1..n_1}$. Therefore, $T^{m,l}$ can be obtained by a finite number of calculations.

Furthermore, let

$$s = \min_{\substack{X^1_i\neq X^2_j\\ i=1..n_1, j=1..n_2}} |X^1_i - X^2_j|. \tag{33}$$

and observe that $T^{m,l} = 0$ for all $m > n$ and for each m, for all $l > \log s^{-1}$ the term $T^{m,l}$ is constant. That is, for each fixed m we have

$$\sum_{l=1}^{\infty} w_m w_l T^{m,l} = w_m w_{\log s^{-1}} T^{m,\log s^{-1}} + \sum_{l=1}^{\log s^{-1}} w_m w_l T^{m,l}.$$

so that we simply double the weight of the last non-zero term. (Note also that s is bounded above by the length of the binary precision in representing the random variables X_j^i.) Thus, even with $m_n, l_n \equiv \infty$ one can calculate $\hat{d}$ precisely. Moreover, for a fixed $m \in 1..\log n$ and $l \in 1..\log s^{-1}$ for every sequence $\mathbf{x}_i,\ i = 1,2$ the frequencies $\nu(\mathbf{x}_i, B),\ B \in B^{m,l}$ may be calculated using suffix trees or suffix arrays, with $\mathcal{O}(n)$ worst case construction and search complexity (see, e.g., Ukkonen, 1995). The construction of the suffix tree is $\mathcal{O}(n)$, and searching all $z := n - m + 1$ occurrences of patterns of length m entails $\mathcal{O}(m + z) = \mathcal{O}(n)$ complexity. This brings the overall computational complexity of (3) to $\mathcal{O}(nm_n \log s^{-1})$; this can potentially be improved using specialized structures (Grossi and Vitter, 2005).

The following consideration can be used to set m_n. For a fixed l the frequencies $\nu(\mathbf{x}_i, B),\ i = 1,2$ of cells in $B \in B^{m,l}$ corresponding to values of $m > \log_n$ are not consistent estimates of their probabilities (and thus only add to the error of the estimate). More specifically for a pattern $X_{j..j+m}$ with $j = 1..n - m$ of length m the probability $\rho_i(X_{j..j+m} \in B),\ i = 1,2$ is of order $2^{-mh_i},\ i = 1,2$ where h_i denotes the entropy rate of $\rho_i,\ i = 1,2$. Therefore, the frequencies of patterns of length $m > \log n$ in $\mathbf{x}_i,\ i = 1,2$ are not consistent estimates of their probabilities. By the above argument, one can set $m_n := \log n$ To choose $l_n < \infty$ one can either fix some constant based on the bound on the precision in real computations, or choose it in such a way that each cell B^{m,l_n} contains no more than $\log n$ points for all $m = 1..\log n$ largest values of l_n.

Complexity of the algorithms. The computational complexity of the presented algorithms is dominated by the complexity of calculations of $\hat{d}$ between different pairs of sequences. Thus, it is enough to bound the number of these computations.

It is easy to see that the offline algorithm for the case of known κ (Algorithm 1) requires at most κN distance calculations, while for the case of unknown κ all N^2 distance calculations are necessary.

The per symbol computational complexity of the online algorithm can be computed as follows. Assume that the pairwise distance values are stored in a database D, and that for every sequence $\mathbf{x}_i^{t-1},\ i \in \mathbb{N}$ we have already constructed a suffix tree, using for example, the online algorithm of (Ukkonen, 1995). At time-step t, a new symbol X is received. Let us first calculate the required computations to update D. We have two cases, either X forms a new sequence, so that $N(t) = N(t-1) + 1$, or it is the subsequent element of a previously received segment, say, $\mathbf{x}_j^t$ for some $j \in 1..N(t)$, so that $n_j(t) = n_j(t-1) + 1$. In either case, let $\mathbf{x}_j^t$ denote the updated sequence. Note that for all $i \neq j \in 1..N(t)$ we have $n_i(t) = n_i(t-1)$. Recall the notation $\mathbf{x}_i^t := X_1^{(i)}, \dots X_{n_i(t)}^{(i)}$ for $i \in 1..N(t)$. In order to update D we need to update the distance between $\mathbf{x}_j^t$ and $\mathbf{x}_i^t$ for all $i \neq j \in N(t)$. Thus, we need to search for all m_n new patterns induced by the received symbol X, resulting in complexity at most $\mathcal{O}(N(t)m_n^2 l_n)$. Let $n(t) := \max\{n_1(t), \dots n_{N(t)}(t)\},\ t \in \mathbb{N}$. As discussed previously, we let $m_n := \log n(t)$; we also define $l_n := \log s(t)^{-1}$ where

$$s(t) := \min_{\substack{i,j \in 1..N(t) \\ u=1..n_i(t), v=1..n_j(t), X_u^{(i)} \neq X_v^{(j)}}} |X_u^{(i)} - X_v^{(j)}|,\ t \in \mathbb{N}.$$

Thus, the per symbol complexity of updating D is at most $\mathcal{O}(N(t)\log^3 n(t))$. However, note that if $s(t)$ decreases from one time-step to the next, updating D will have a complexity of order equivalent to its complete construction, resulting in a computational complexity at most $\mathcal{O}(N(t)n(t)\log^2 n(t))$. Therefore, we avoid calculating $s(t)$ at every time-step; instead, we update $s(t)$ at pre specified time-steps so that for every $n(t)$ symbols received, D is reconstructed at most $\log n(t)$ times. (This can be done, for example, by recalculating $s(t)$ at time-steps where $n(t)$ is a power of 2.) It is easy to see that with the database D of distance values at hand, the rest of the computations are of order at most $\mathcal{O}(N(t)^2)$. Thus, the per symbol computational complexity of Algorithm 2 is at most $\mathcal{O}(N(t)^2 + N(t)\log^3 n(t))$.

6. Experimental Results

In this section we present empirical evaluations of Algorithms 1 and 2 on both synthetically generated and real data.

6.1 Synthetic Data

We start with synthetic experiments. In order for the experiments to reflect the generality of our approach 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 (Shields, 1996), 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. (Ornstein and Weiss, 1990)).

6.1.1 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 $X_i := \mathbb{I}\{r_i > 0.5\}$. If α is irrational then $\mathbf{x}$ forms a stationary ergodic time series. (We simulate α by a `longdouble` with a long mantissa.)

For the purpose of our experiments, first we fix $\kappa := 5$ difference process distributions specified by $\alpha_1 = 0.31...$, $\alpha_2 = 0.33...$, $\alpha_3 = 0.35...$, $\alpha_4 = 0.37...$, $\alpha_5 = 0.39$. The parameters α_i are intentionally selected to be close, in order to make the process distributions harder to distinguish. Next we generate an $N \times M$ data matrix $\mathbf{X}$, each row of which is a sequence generated by one of the process distributions. Our task in both the online and the batch setting is to cluster the rows of $\mathbf{X}$ into $\kappa = 5$ clusters.

6.1.2 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.

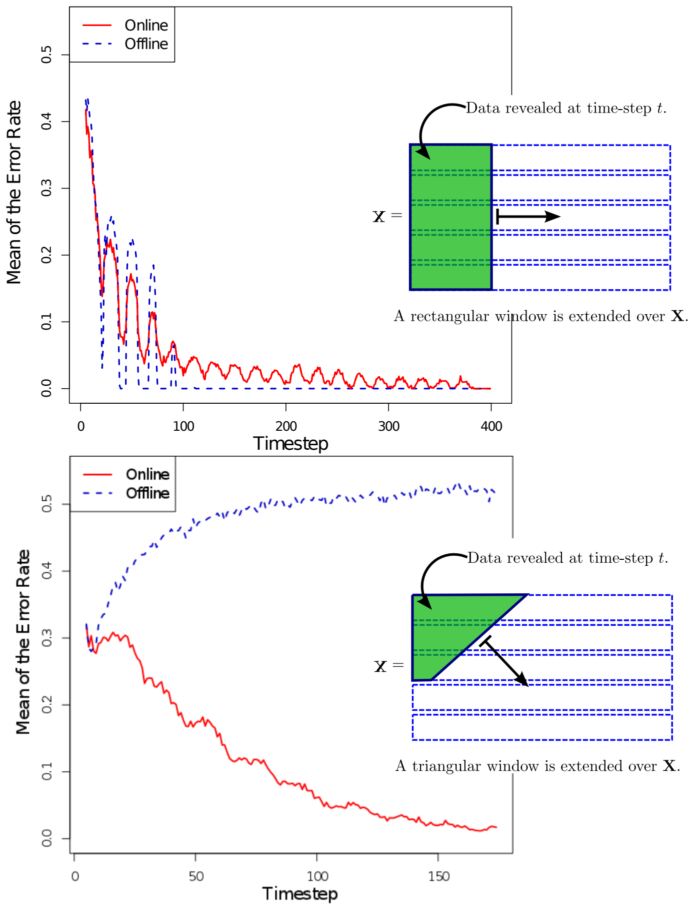

Figure 2: Top: error-rate vs. sequence length in batch setting. Bottom: error-rate vs. Number of observed samples in online setting. (error-rates averaged over 100 runs.)

6.1.3 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 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. Note that the performance of both algorithms is measured on all sequences available at a given time, not on a fixed batch of sequences. 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.

6.2 Real Data

As an 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) (cmu) 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 to those obtained with two other methods, namely those of (Li and Prakash, 2011) and (Jebara et al., 2007), 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 the numerical results directly from the corresponding articles. In order to have common grounds for each comparison we use the same sets of sequences,[2] and the same means of evaluation as those used by (Li and Prakash, 2011; Jebara et al., 2007).

In the paper by (Li and Prakash, 2011) two MOCAP datasets[3] 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 by (Li and Prakash, 2011) 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 Algorithm 1 as well as that reported for the method of (Li and Prakash, 2011); Algorithm 1 performs consistently better.

In the paper by (Jebara et al., 2007) four MOCAP datasets[4] 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, Algorithm 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.

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

	Dataset	(Li and Prakash, 2011)	Algorithm 1
1.	Walk vs. Run (#35)	0.1015	**0**
2.	Walk vs.Run (#16)	0.3786	**0.2109**

Table 1: Comparison with (Li and Prakash, 2011): Performance in terms of entropy; datasets concern ergodic motion captures.

	Dataset	(Jebara et al., 2007)	Algorithm 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 (Jebara et al., 2007): Performance in terms of accuracy; Rows 1 & 2 concern ergodic, Rows 3 & 4 concern non-ergodic motion captures.

7. Discussion

We have proposed a natural notion of consistency for clustering time series in both the online and the offline scenarios. While in this work we have taken some first steps in investigating the theoretical and algorithmic questions arising in the proposed framework, there are many open problems and exciting directions for future research remaining to be explored. Some of these are discussed in this section.

Rates, optimality. The main focus of this work is on the most general case of highly dependent time series. On the one hand, this captures best the spirit of the unsupervised learning problem in question: the nature of the data is completely unknown, and one tries to find some structure in it. On the other hand, as discussed above, in this generality rates of convergence and finite-sample performance guarantees are provably impossible to obtain, and thus one cannot argue about optimality. While we have provided some results on a more restrictive setting (time series with mixing, Section 4.3), the question of what are the optimal performance guarantees for different classes of time series remains open. In fact, the first interesting question in this direction is not about time series with mixing, but about i.i.d. series. What is the minimal achievable probability of clustering error in this setting, for finite sample sizes, and what algorithms attain it? One can also introduce **decisions** in this framework, in the **bandit-like** fashion: At each step of time, one can request a sample from one out of N sequences. The goal is, again, to cluster the sequences, which means to determine which of them are generated by the same distribution, requesting as few samples as possible. This question is interesting both for a known and unknown number of clusters k.

Relaxing the setting. In the proposed setting we have N (or $N(t)$) time series generated by k different time series distributions. As discussed in Section 4.1, in this formulation, from some time on, the sequences possess the so-called strict separation property: sequences in the same target cluster are closer to each other than to sequences in other clusters. One possible way to relax the considered setting is to impose the strict separation property on the distributions that generate the data. That is, each sequence $\mathbf{x}_i, i = 1..N$, may be generated by its own distribution ρ_i, but the distributions $\{\rho_i : i = 1..N\}$ can be clustered in such a way that the resulting clustering has the strict separation property with respect to the metric d. The goal would then be to recover this clustering based on the samples given. In fact, one can show that the offline Algorithm 1 of Section 4.1 is consistent in this setting as well. How this transfers to the online setting remains open.

Online setting: bad points. In the online setting of Section 4.2 we have assumed that the length of each sequence grows to infinity with time. While this is a good first approximation, this assumption may not be practical. It is interesting to consider the situation in which some sequences stop growing at some point; moreover, one can assume that such sequences are not representative of the corresponding distribution. While this clearly makes the problem much more difficult, already in the setting considered in this work we are dealing with "bad" sequences at each step of time: these are those sequences which are as yet too short to be informative. This hints at the possibility of obtaining consistent algorithm in the extended setting outlined.

Other metrics and non-metric-based methods. All the methods presented above are based on the distributional distance d. The main property of this distance that we exploit is that it can be estimated consistently. In principle, one can use other distances that have this property, to obtain consistent clustering algorithms. While there are not many known distances that can be consistently estimated for arbitrary stationary ergodic distributions, there is at least one, namely the telescope distance introduced recently in (Ryabko and Mary, 2012). Moreover, the definition of consistency proposed does not entail the need to use a distance between time series distributions. As an alternative, the use compression-based methods can be considered. Such methods have been used to solve various statistical problems concerning stationary ergodic time series, see, for example, (Ryabko and Astola, 2006; Ryabko, 2010a). Compression-based methods have also been used for clustering time series data before, albeit without asymptotic consistency analysis, in (Cilibrasi and Vitanyi, 2005). Combining our consistency framework with these compression-based methods is a promising direction for further research.

References

CMU graphics lab motion capture database. URL `http://mocap.cs.cmu.edu/`.

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.

M.F. Balcan, A. Blum, and S. Vempala. A discriminative framework for clustering via similarity functions. In *STOC*, 2008.

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.

P. Billingsley. Statistical inference about Markov chains. *Ann. Math. Stat.*, 32(1):12–40, 1961.

D. Bosq. *Nonparametric Statistics for Stochastic Processes.* Estimation and Prediction. Springer, 1996. ISBN 0-387-94713-2.

R.C. Bradley. Basic properties of strong mixing conditions. a survey and some open questions. *Probability Surveys*, 2:107–144, 2005.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006. ISBN 0521841089.

R. Cilibrasi and P.M.B. Vitanyi. Clustering by compression. *IEEE Transactions on Information Theory*, 51:1523–1545, 2005.

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

Roberto Grossi and Jeffrey Scott Vitter. Compressed suffix arrays and suffix trees with applications to text indexing and string matching. *SIAM Journal on Computing*, 35(2): 378–407, 2005.

M. Gutman. Asymptotically optimal classification for multiple tests with empirically observed statistics. *IEEE Transactions on Information Theory*, 35(2):402–408, 1989.

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

A. Khaleghi and D. Ryabko. Locating changes in highly-dependent data with unknown number of change points. In *Neural Information Processing Systems (NIPS)*, Lake Tahoe, Nevada, United States, 2012.

A. Khaleghi, D. Ryabko, J. Mary, and P. Preux. Online clustering of processes. In *AISTATS*, JMLR W&CP 22, pages 601–609, 2012.

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

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.

E. Lehmann. *Testing Statistical Hypotheses, 2nd edition.* Wiley, New York, 1986.

L. Li and B.A. Prakash. Time series clustering: Complex is simpler! 2011.

M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar k-means problem is np-hard. In *WALCOM '09: Proceedings of the 3rd International Workshop on Algorithms and Computation*, pages 274–285, Berlin, Heidelberg, 2009. Springer-Verlag.

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

B. Ryabko and J. Astola. Universal codes as a basis for time series testing. *Statistical Methodology*, 3:375–397, 2006.

Boris Ryabko. Applications of universal source coding to statistical analysis of time series. *Selected Topics in Information and Coding Theory, World Scientific Publishing*, pages 289–338, 2010a.

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

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

D. Ryabko. Testing composite hypotheses about discrete ergodic processes. *Test*, 21(2): 317–329, 2012.

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

Daniil Ryabko and Jeremie Mary. Reducing statistical time-series problems to binary classification. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2069–2077. 2012.

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

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

Esko Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14(3):249–260, 1995.

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