Data-driven abstractions via adaptive refinements and a Kantorovich metric [extended version]

Adrien Banse, Licio Romao, Alessandro Abate and Raphaël M. Jungers*

March 30, 2023

Abstract

We introduce an adaptive refinement procedure for smart, and scalable abstraction of dynamical systems. Our technique relies on partitioning the state space depending on the observation of future outputs. However, this knowledge is dynamically constructed in an adaptive, asymmetric way. In order to learn the optimal structure, we define a Kantorovich-inspired metric between Markov chains, and we use it as a loss function. Our technique is prone to data-driven frameworks, but not restricted to.

We also study properties of the above mentioned metric between Markov chains, which we believe could be of application for wider purpose. We propose an algorithm to approximate it, and we show that our method yields a much better computational complexity than using classical linear programming techniques.

1 Introduction

Feedback control of dynamical systems is at the core of several techniques that have caused tremendous impact in several industries, being essential to important advancements in e.g. aerospace and robotics. Traditionally, these control techniques were model-based, relying on a complete mathematical model to perform controller design. With recent technological advancements, however, where a vast amount of data can be collected online or offline, the interest within the control community to study methods that leverage available data for feedback controller design has been reignited [1, 2, 3, 4].

In this paper, we focus on data-driven techniques for building *abstractions of dynamical systems.* Abstractions methods create a symbolic model [5, 6] that approximates the behaviour of the original (the "concrete") dynamics in a way that controllers designed for such a symbolic representation can be refined to a valid controller for the original dynamics. The main advantage of abstraction methods with respect to standard control techniques is that one can transfer formal properties from the abstract system to the concrete one in a rigorous manner. Moreover, one can enforce complex temporal properties [7, 8], such as those described by LTL, STL, or PCTL temporal logics, using standard automata or MDP-based algorithms widely studied within the formal verification and control communities [9, 10, 11, 12, 13]. Classical abstraction methods, however, do not scale well with the state space dimension of the original dynamics, as they usually require a partitioning of the state whose complexity grows exponentially with the underlying dimension. Besides, most of the existing abstraction techniques have been designed for when a full mathematical representation of the dynamics is available.

*R. M. Jungers is a FNRS honorary Research Associate. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under grant agreement No 864017 - L2C. R. M. Jungers is also supported by the Walloon Region and the Innoviris Foundation. He is currently on sabbatical leave at Oxford University, Department of Computer Science, Oxford, UK. Adrien Banse is supported by the French Community of Belgium in the framework of a FNRS/FRIA grant. Adrien Banse and Raphaël M. Jungers are with ICTEAM, UCLouvain. E-mail adresses: {adrien.banse, raphael.jungers}@uclouvain.be. Licio Romao and Alessandro Abate are with the Department of Computer Science, Oxford University. E-mail adresses: {licio.romao, alessandro.abate}@cs.ox.ac.uk.

Several recent research efforts started exploring the possibility of generating data-driven abstractions for stochastic dynamical systems [14, 15, 16, 17]. In [14], we show that memory-based Markov models can be built from trajectory data. Memory has been classically used as a tool to mitigate non-Markovian behaviors of the original dynamics [11, 13], a feature also explored in recent papers [11, 16]. Increasing memory allows us to create more precise representations of the original dynamics using Markov decision processes or Markov chains. In [14], we also propose a heuristic to validate this observation, and prove a theoretic result showing convergence of the behaviour of the discrete model to the original dynamical when memory increases (under observability and ergodicity assumptions on the original dynamics). Despite promising results, paper [14] does not offer an adaptive mechanism to compute the generated abstraction, and thus it faces the curse of dimensionality, as the number of possible observations grows exponentially with the memory length.

We follow-up on our results in [14] and address their main shortcomings. At the core of the approach in this paper is the construction of a novel metric between two Markov chains; this metric is then exploited to adaptively increase memory in certain regions of the state space, in view of taming the complexity of the generated abstraction. As opposed to the approach we take in [14], where the states of the chain are built using past memory, the abstractions we construct in this paper are based on forward memory. In order to define a metric between two Markov models, we leverage the *Kantorovich* metric (also known as the Wasserstein or Earth's mover distance) between the induced probability on words of a fixed length and let the word length go to infinity. To define the Kantorovich metric, we equip the space of words with the *Baire distance* [18], turning it into a metric space. The Baire distance is classically used in symbolic dynamics [5], and we argue that it is a natural and relevant choice for control purposes, which is confirmed by numerical experiments. Not only we show that proposed metric between Markov models is well-defined, but we also present an efficient algorithm that avoids solving linear programs of increasing size, which would lead to a prohibitive computational burden.

Computing metrics between Markov models has been an active research topic within the computer science community [19]. Our construction on the metric between Markov chain resembles the one presented in [20], however with another metric, namely the Kantorovich metric induced by the Baire distance. In [21] computability and complexity results are shown for the total variation metric. Kantorovich metrics for Markov models have been studied in [22, 23, 24, 25], but their underlying distance is different from ours, which is crucial for both computational aspects, and relevant for our purposes.

Our approach is the first one to provide a data-driven, adaptive abstraction method for dynamical systems that leverages memory to alleviate non-Markorvian behaviors associated with the original dynamics. Overall, our main contributions are:

- We propose a new metric to measure distance between Markov models. This metric is the limit of the Kantorovich metric, defined using the Baire distance between words, when the word length tends to infinity.
- We develop an efficient algorithm that approximates arbitrarily well the proposed metric.
- We exploit the proposed metric to adaptively increase memory in specific regions of the state space. This allows us to generate less complex data-driven abstractions for dynamical systems in a smart and adaptive way.

Outline The rest of this paper is organized as follows. In Section 2, we introduce the Kantorovich metric between two Markov chains. We also propose an efficient algorithm to approximate it arbitrarily close. In Section 3, we apply this metric to data-driven construction of abstractions for dynamical systems in a greedy way. In particular, we use the Kantorovich metric as a tool to compare different abstractions, leading to an adaptive refinement procedure. We also demonstrate the quality of our procedure on an example.

Notations Let $\mathcal{A}$ be a finite alphabet, and $\mathcal{A}^n$ the set of n-long sequences of this alphabet. The set $\mathcal{A}^*$ is the set of countable cartesian product of $\mathcal{A}$. The symbol Λ stands for the empty sequence and, for any $w_1 \in \mathcal{A}^{n_1}$, $w_2 \in \mathcal{A}^{n_2}$, the sequence $w_1 w_2 \in \mathcal{A}^{n_1+n_2}$ is the concatenation of w_1 and w_2. In terms of

computational complexity, let $c(x)$ be a number of operations w.r.t. some attributes x. We say that an algorithm has a computational complexity $\mathcal{O}(f(x))$ if there exists $M > 0$, x_0 such that, for all $x \geq x_0$, $c(x) \leq Mf(x)$. Concerning the measure theoretical concepts, for any bounded set $X \subset \mathbb{R}^d$, let $\sigma(X)$ be the σ-algebra of X, and λ be the Lebesgue measure on $\mathbb{R}^d$, then $\lambda_X : \sigma(X) \to [0,1]$ is defined as $\lambda_X(A) = \lambda(A)/\lambda(X)$[1]. Finally, for any set X and function F, the set $F(X) = \{F(x) : x \in X\}$.

2 A Kantorovich metric between Markov chains

2.1 Preliminaries

Using a similar formalism as in [14], we define a labeled Markov chain as follows.

Definition 1 (Markov chain)**.** A Markov chain is the 5-tuple $\Sigma = (\mathcal{S}, \mathcal{A}, P, \mu, L)$, where

- $\mathcal{S}$ is a finite set of states,
- $\mathcal{A}$ is a finite alphabet,
- P is the transition matrix on $\mathcal{S} \times \mathcal{S}$,
- μ is the initial measure on $\mathcal{S}$,
- and $L : \mathcal{S} \to \mathcal{A}$ is a labelling function.

In Definition 1, the element of the transition matrix $P_{s,s'}$ is the probability $\mathbb{P}(X_{k+1} = s' | X_k = s)$. The labelling L is not set-valued, and therefore induces a partition of the states. Consider the equivalence relation on $\mathcal{S}$ defined as $s \sim s'$ if and only if $L(s) = L(s')$. For any $a \in \mathcal{A}$, the notion of equivalent classes is defined as

$$[a] = \{s \in \mathcal{S} : L(s) = a\}. \tag{1}$$

We also define the behaviour of a Markov chain $\mathcal{B}(\Sigma) \subseteq \mathcal{A}^*$ as follows. A sequence $w^* = (a_1, a_2, \dots) \in \mathcal{B}(\Sigma_{\mathcal{W}})$ if there exists $s_1, s_2, \dots \in \mathcal{S}$ such that $\mu_{s_1} > 0$, $P_{s_i, s_{i+1}} > 0$ and $L(s_i) = a_i$.

In the present work, we focus on a notion of metric between probabilities on label sequences. Let $w = (a_1, \dots, a_n)$ be a n-long sequence of labels, and $p^n : \mathcal{A}^n \to [0,1]$ be the probability associated to the sequences. It is defined as

$$p^n(w) = \sum_{s_1 \in [a_1]} \mu_{s_1} \sum_{s_2 \in [a_2]} P_{s_1,s_2} \cdots \sum_{s_n \in [a_n]} P_{s_{n-1},s_n}. \tag{2}$$

Remark 1. Classical procedures are well-known in the literature allowing to compute the probabilities p^n for increasing n, with a complexity proportional to $|\mathcal{S}|^2$ at every step [26].

We endow the set of n-long sequences of labels with the Baire's distance d_B defined as follows.

Definition 2 (Baire's distance, [18])**.** The Baire's distance $d_B : \mathcal{A}^n \to \mathbb{R}$ is defined as

$$d_B(w_1, w_2) = 2^{-l}, \tag{3}$$

where l is the length of the longest common prefix, that is $l = \inf\{k : a_k \neq b_k\}$ for $w_1 = (a_1, \dots, a_n)$ and $w_2 = (b_1, \dots, b_n)$.

Remark 2. It is well-known that the Baire's distance is an ultrametric [27]. It means that is satisfies the strong triangular inequality

$$d_B(w_1, w_3) \leq \max\{d_B(w_1, w_2), d_B(w_2, w_3)\}. \tag{4}$$

This property will be crucial in our developments.

[1]The measure λ_X well defined, since it is absolutely continuous with respect to λ with the Radon-Nikodym derivative being the indicator function $\mathbb{I}_X$

2.2 The Kantorovich metric

Consider two Markov chains $\Sigma_1 = (\mathcal{S}_1, \mathcal{A}, P_1, \mu_1, L_1)$ and $\Sigma_2 = (\mathcal{S}_2, \mathcal{A}, P_2, \mu_2, L_2)$ defined on the same alphabet $\mathcal{A}$. For a fixed n, they respectively generate the distributions p_1^n and p_2^n defined on the metric space $(\mathcal{A}^n, d_B)$ as described in (2). The Kantorovich metric between p_1^n and p_2^n is defined as follows.

Definition 3 (Kantorovich metric)**.** The Kantorovich metric between the probability distributions p_1^n and p_2^n is defined as

$$K(p_1^n, p_2^n) = \inf_{\pi^n \in \Pi(p_1^n, p_2^n)} \sum_{w_1, w_2 \in \mathcal{A}^n} d_B(w_1, w_2) \pi^n(w_1, w_2), \tag{5}$$

where $\Pi(p_1^n, p_2^n)$ is the set of couplings of p_1^n and p_2^n, that is the set of joint distributions $\pi^n : \mathcal{A}^n \times \mathcal{A}^n \to [0, 1]$ whose marginal distributions are p_1^n and p_2^n.

The constraint $\pi^n \in \Pi(p_1^n, p_2^n)$ can also be written as

$$\begin{aligned} &\forall w_1, w_2 \in \mathcal{A}^n : \pi^n(w_1, w_2) \geq 0, \\ &\forall w_1 \in \mathcal{A}^n : \sum_{w_2 \in \mathcal{A}^n} \pi^n(w_1, w_2) = p_1^n(w_1), \\ &\forall w_2 \in \mathcal{A}^n : \sum_{w_1 \in \mathcal{A}^n} \pi^n(w_1, w_2) = p_2^n(w_2). \end{aligned} \tag{6}$$

The Kantorovich metric is often interpreted as an optimal transport problem. Indeed one can see the problem (5) as the problem of finding the optimal way to satisfy "demands" p_2^n with "supplies" p_1^n, where the cost of moving $\pi^n(w_1, w_2)$ probability mass from w_1 to w_2 amounts to $\pi^n(w_1, w_2) d_B(w_1, w_2)$. An illustration is provided in Figure 1.

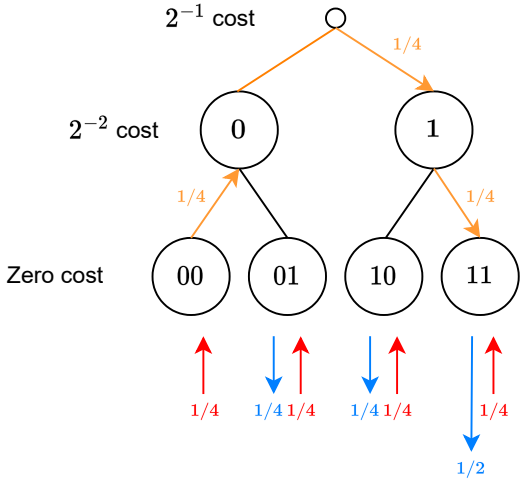

Figure 1: Interpretation of the Kantorovich distance as an optimal transport problem. In this example, $p_1^2(w) = 1/4$ for all $w \in \mathcal{A}^2$, and $p_2^2(00) = 0$, $p_2^2(01) = p_2^2(10) = 1/4$, and $p_2^2(11) = 1/2$. One can see that the optimal way to satisfy the demands p_2^n with the supplies p_1^n is to move $1/4$ of probability mass from 00 to 11, that is $\pi^2(00, 11) = 1/4$. Since $d_B(00, 11) = 1/2$, the Kantorovich distance is $K(p_1^2, p_2^2) = 1/8$.

A naïve computation of $K(p_1^n, p_2^n)$ in (5) is by linear programming. One can use techniques such as interior point methods, or network simplex to solve (5) which, in some cases, can be solved in $\mathcal{O}(n|\mathcal{A}|^{3n} \log(|\mathcal{A}|))$ computational complexity, and therefore scales very poorly with the number labels. In this section, we show that it is possible to compute $K(p_1^n, p_2^n)$ in $\mathcal{O}(|\mathcal{S}|^2 |\mathcal{A}|^{n+1})$ operations. We first present two lemmata that will be useful for our purpose.

Lemma 1. *For any $n \geq 1$, let π^n be the solution of (5). For all $w \in \mathcal{A}^n$,*

$$\pi^n(w,w) = \min\{p_1^n(w), p_2^n(w)\}. \tag{7}$$

Proof. We first prove that $\pi^n(w,w) \leq \min(p_1^n(w), p_2^n(w))$. Constraints (6) imply that, for all $w \in \mathcal{A}^n$,

$$\begin{aligned} p_1^n(w) &= \pi^n(w,w) + \sum_{\substack{w' \in \mathcal{A}^n \\ w' \neq w}} \pi^n(w,w') \geq \pi^n(w,w), \\ p_2^n(w) &= \pi^n(w,w) + \sum_{\substack{w' \in \mathcal{A}^n \\ w' \neq w}} \pi^n(w',w) \geq \pi^n(w,w), \end{aligned} \tag{8}$$

which imply that $\pi^n(w,w) \leq \min\{p_1^n(w), p_2^n(w)\}$. Now we prove that $\pi^n(w,w) \geq \min\{p_1^n(w), p_2^n(w)\}$. Consider an optimal solution π^n to the problem (5) such that

$$\pi^n(w,w) = \min\{p_1^n(w), p_2^n(w)\} - \varepsilon, \tag{9}$$

for some $w \in \mathcal{A}^n$ and $\varepsilon > 0$. Assume w.l.o.g. that $\min\{p_1(w), p_2(w)\} = p_1(w)$. Therefore, constraints (6) imply that

1. there exists $w' \neq w$, such that $\pi^n(w,w') = \varepsilon'$ for some $\varepsilon' \in (0, \varepsilon]$, and
2. there exists $w'' \neq w$ such that $\pi^n(w'',w) = \varepsilon''$ for some $\varepsilon'' \in (0, \varepsilon]$.

Let $K(p_1^n, p_2^n)$ denote the Kantorovich metric corresponding to such π^n. Now assume w.l.o.g. that $\varepsilon' \leq \varepsilon''$. Consider then $(\pi^n)'$ such that $(\pi^n)'(w_1,w_2) = \pi^n(w_1,w_2)$ for all $w_1, w_2 \in \mathcal{A}^n$ except

1. $(\pi^n)'(w,w) = \pi^n(w,w) + \varepsilon'$,
2. $(\pi^n)'(w,w') = \pi^n(w,w') - \varepsilon'$,
3. $(\pi^n)'(w'',w') = \pi^n(w'',w') + \varepsilon'$, and
4. $(\pi^n)'(w'',w) = \pi^n(w,w) - \varepsilon'$.

The joint distribution $(\pi^n)'$ is feasible since it still satisfies the constraints (6). Now let $K'(p_1^n, p_2^n)$ denote the solution corresponding to such $(\pi^n)'$, we have that

$$K'(p_1^n, p_2^n) = K(p_1^n, p_2^n) - \varepsilon' \left[d_B(w,w') + d_B(w',w'') - d_B(w',w'') \right]. \tag{10}$$

Since the Baire's distance d_B as defined in Definiton 2 satisfies triangular inequality, we have that $K'(p_1^n, p_2^n) < K(p_1^n, p_2^n)$, which is a contradiction. ■

Lemma 2. *For any $n \geq 1$, let π^{n+1} be the solution of (5). Then, for all $w \in \mathcal{A}^n$ such that $p_1^n(w) > p_2^n(w)$, then*

$$\begin{aligned} \sum_{\substack{w' \in \mathcal{A}^n \\ w' \neq w}} \sum_{a_1, a_2 \in \mathcal{A}} \pi^{n+1}(wa_1, w'a_2) &= p_1^n(w) - p_2^n(w), \\ \sum_{\substack{w' \in \mathcal{A}^n \\ w' \neq w}} \sum_{a_1, a_2 \in \mathcal{A}} \pi^{n+1}(w'a_1, wa_2) &= 0, \end{aligned} \tag{11}$$

and for all $w_1 \in \mathcal{A}^n$ such that $p_1^n(w_1) \leq p_2^n(w_1)$,

$$\begin{aligned} \sum_{\substack{w' \in \mathcal{A}^n \\ w' \neq w}} \sum_{a_1, a_2 \in \mathcal{A}} \pi^{n+1}(wa_1, w'a_2) &= 0, \\ \sum_{\substack{w' \in \mathcal{A}^n \\ w' \neq w}} \sum_{a_1, a_2 \in \mathcal{A}} \pi^{n+1}(w'a_1, wa_2) &= p_2^n(w) - p_1^n(w). \end{aligned} \tag{12}$$

Proof. For some $w \in \mathcal{A}^n$, assume w.l.o.g. that $p_1^n(w) > p_2^n(w)$. First, by feasibility conditions,

$$\sum_{\substack{w' \in \mathcal{A}^n \\ w' \neq w}} \sum_{a_1, a_2 \in \mathcal{A}} \pi^{n+1}(wa_1, w'a_2) \geq p_1^n(w) - p_2^n(w). \tag{13}$$

Now, we proceed similarly as for Lemma 1. Suppose by contradiction that

$$\sum_{\substack{w' \in \mathcal{A}^n \\ w' \neq w}} \sum_{a_1, a_2 \in \mathcal{A}} \pi^{n+1}(wa_1, w'a_2) > p_1^n(w) - p_2^n(w). \tag{14}$$

Then there exists $w' \neq w \in \mathcal{A}^n$, and $a_1, a_2 \in \mathcal{A}$ such that $\pi^{n+1}(w'a_1, wa_2) = \varepsilon' > 0$. There also exists $w'' \in \mathcal{A}^n$ such that $w'' \neq w$ and $w'' \neq w'$, and $a_3, a_4 \in \mathcal{A}$ such that $\pi^{n+1}(wa_3, w''a_4) = \varepsilon'' > 0$. Asumme w.l.o.g. that $\varepsilon' \leq \varepsilon''$, and consider a solution $(\pi^{n+1})'$ such that $(\pi^{n+1})' = \pi^{n+1}$, except for

1. $(\pi^{n+1})'(wa_3, w''a_4) = \pi^{n+1}(wa_3, w''a_4) - \varepsilon'$,
2. $(\pi^{n+1})'(w'a_1, wa_2) = \pi^{n+1}(w'a_1, wa_2) - \varepsilon'$,
3. $(\pi^{n+1})'(w'a_1, w''a_4) = \pi^{n+1}(w'a_1, w''a_4) + \varepsilon'$, and
4. $(\pi^{n+1})'(wa_3, wa_2) = \pi^{n+1}(wa_3, wa_2) + \varepsilon'$.

The joint distribution $(\pi^{n+1})'$ is feasible since it still satisfies the constraints (6). Note that, since the Baire's distance satisfies the strong triangular inequality (see Remark 2),

$$\begin{aligned} d_B(w'a_1, w''a_4) &\leq \max\{d_B(w'a_1, wa_2), d_B(wa_2, w''a_4)\} \\ &= \max\{d_B(w'a_1, wa_2), d_B(wa_3, w''a_4)\}. \end{aligned} \tag{15}$$

Moreover, $d_B(wa_3, wa_2) = 2^{-(n+1)}$. Now let $K'(p_1^{n+1}, p_2^{n+1})$ denote the solution corresponding to such $(\pi^{n+1})'$, we have that $K(p_1^{n+1}, p_2^{n+1}) - K'(p_1^{n+1}, p_2^{n+1})$ is

$$\begin{aligned} & -\varepsilon' \begin{bmatrix} + & d_B(w'a_1, wa_2) \\ + & d_B(wa_3, w''a_4) \\ - & d_B(w'a_1, w''a_4) \\ - & 2^{-(n+1)} \end{bmatrix} \\ & \leq -\varepsilon' \begin{bmatrix} + & d_B(w'a_1, wa_2) + d_B(wa_3, w''a_4) \\ - & \max\{d_B(w'a_1, wa_2), d_B(wa_3, w''a_4)\} \\ - & 2^{-(n+1)} \end{bmatrix} \\ & \leq -\varepsilon'[2^{-n} - 2^{-(n+1)}] \\ & \leq 0, \end{aligned} \tag{16}$$

which contradicts the fact that π^{n+1} is optimal. ■

We are now able to present the result that will allow us to write a dynamic programming algorithm to compute the Kantorovic distance.

Theorem 1. *For any $n \geq 1$, let π^n be the solution of (5). Then the following holds:*

$$K(p_1^{n+1}, p_2^{n+1}) = K(p_1^n, p_2^n) + 2^{-(n+1)} \sum_{w \in \mathcal{A}^n} \left[\pi^n(w, w) - \sum_{a \in \mathcal{A}} \pi^{n+1}(wa, wa) \right]. \tag{17}$$

Proof. For the sake of clarity, we note $K^n = K(p_1^n, p_2^n)$ and $r(w) = \min\{p_1^n(w), p_2^n(w)\}$. By Lemma 1, it is equivalent to prove that

$$K^{n+1} = K^n + 2^{-(n+1)} \sum_{w \in \mathcal{A}^n} \left[r(w) - \sum_{a \in \mathcal{A}} r(wa) \right]. \tag{18}$$

We first prove that the right hand side of (18) is a lower bound for K^{n+1}. First we note that K^{n+1} is equal to

$$\begin{aligned} &\sum_{w_1, w_2 \in \mathcal{A}^n} \sum_{a_1, a_2 \in \mathcal{A}} d_B(w_1 a_1, w_2 a_2) \pi^{n+1}(w_1 a_1, w_2 a_2) \\ &= \sum_{\substack{w_1, w_2 \in \mathcal{A}^n \\ w_1 \neq w_2}} d_B(w_1, w_2) \sum_{a_1, a_2 \in \mathcal{A}} \pi^{n+1}(w_1 a_1, w_2 a_2) + 2^{-(n+1)} \sum_{w \in \mathcal{A}^n} \sum_{\substack{a_1, a_2 \in \mathcal{A} \\ a_1 \neq a_2}} \pi^{n+1}(w a_1, w a_2) \\ &:= C_1 + C_2. \end{aligned} \tag{19}$$

We first prove that $C_1 \geq K^n$. To do this, let $\mu^n : \mathcal{A}^n \times \mathcal{A}^n \to [0, 1]$ be defined as

$$\mu^n(w_1, w_2) = \sum_{a_1, a_2 \in \mathcal{A}} \pi^{n+1}(w_1 a_1, w_2 a_2). \tag{20}$$

We show that μ^n satisfies the constraints (6). Indeed $\mu^n(w_1, w_2) \geq 0$,

$$\begin{aligned} \sum_{w_2 \in \mathcal{A}^n} \mu(w_1, w_2) &= \sum_{w_2 \in \mathcal{A}^n} \sum_{a_1, a_2 \in \mathcal{A}} \pi^{n+1}(w_1 a_1, w_2 a_2) \\ &= \sum_{a_1} p_1^{n+1}(w_1 a_1) \\ &= p_1^n(w_1), \end{aligned} \tag{21}$$

and similarly for the third condition in (6). This implies that μ^n is a coupling, thereby a feasible solution of (5). This yields

$$K^n \leq \sum_{a_1, a_2 \in \mathcal{A}} d_B(w_1, w_2) \mu^n(w_1, w_2) = C_1. \tag{22}$$

Now we show that, for all $w \in \mathcal{A}^n$,

$$\sum_{\substack{a_1, a_2 \in \mathcal{A} \\ a_1 \neq a_2}} \pi^{n+1}(w a_1, w a_2) = r(w) - \sum_{a \in \mathcal{A}} r(wa), \tag{23}$$

which implies that

$$C_2 = 2^{-(n+1)} \sum_{w \in \mathcal{A}^n} \left[\sum_{w \in \mathcal{A}^n} r(w) - \sum_{a \in \mathcal{A}} r(wa) \right]. \tag{24}$$

We prove the claim. Assume w.l.o.g. that w is such that $p_1^n(w) > p_2^n(w)$, then

$$\begin{aligned} &\sum_{a_1, a_2 \in \mathcal{A}} \pi^{n+1}(w a_1, w a_2) \\ &= \sum_{a_1 \in \mathcal{A}} \sum_{w' \in \mathcal{A}^n} \sum_{a_2 \in \mathcal{A}} \pi^{n+1}(w a_1, w' a_2) - \sum_{\substack{w' \in \mathcal{A}^n \\ w' \neq w}} \sum_{a_2 \in \mathcal{A}} \pi^{n+1}(w a_1, w' a_2) \\ &= \sum_{a_1 \in \mathcal{A}} p_1^n(w a_1) - \sum_{\substack{w' \in \mathcal{A}^n \\ w' \neq w}} \sum_{a_1, a_2 \in \mathcal{A}} \pi^{n+1}(w a_1, w' a_2) \end{aligned} \tag{25}$$

Following Lemma 2, this is equal to

$$p_1^n(w) - (p_1^n(w) - p_2^n(w)) = r(w). \tag{26}$$

And the following holds:

$$\begin{aligned} &\sum_{\substack{a_1,a_2\in\mathcal{A}\\ a_1\neq a_2}} \pi^{n+1}(wa_1, wa_2) \\ =&\sum_{\substack{a_1\neq a_2\in\mathcal{A}\\ a_1\neq a_2}} \pi^{n+1}(wa_1, wa_2) - \sum_{a\in\mathcal{A}} \pi^{n+1}(wa_1, wa_2) \\ =&\, r(w) - \sum_{a\in\mathcal{A}} r(wa) \end{aligned} \tag{27}$$

by Lemma 1. This concludes that the right hand side of (18) is a lower bound for K^{n+1}.

Now, to provide an upper bound, we will show that we can construct a feasible $n+1$ solution feasible μ^{n+1} such that

$$\sum_{w_1,w_2\in\mathcal{A}^{n+1}} d_B(w_1,w_2)\mu^{n+1}(w_1,w_2) = K^n + \sum_{w\in\mathcal{A}^n}\left[r(w) - \sum_{a\in\mathcal{A}} r(wa)\right]. \tag{28}$$

Consider π^n, an optimal solution at step n. We will construct μ^{n+1} in the following greedy way. Initialize μ^{n+1} with only zero elements, and for all $w\in\mathcal{A}^n$, $a\in\mathcal{A}$, we initialize $\delta(wa)=0$. We start by updating the blocks $\mu^{n+1}(w_1a_1, w_2a_2)$ where $w_1\neq w_2$. For all w such that $p_1^n(w) > p_2^n(w)$, for all $a\in\mathcal{A}$ such that $p_1^{n+1}(wa) > p_2^{n+1}(wa)$, do the following.

1. Let $\tilde{\delta}(wa) = p_1^{n+1}(wa) - p_2^{n+1}(wa)$.
   If $\sum_{a'\neq a}\delta(wa') + \tilde{\delta}(wa) > p_1^n(w) - p_2^n(w)$, let $\delta(wa) = (p_1^n(w) - p_2^n(w)) - \sum_{a'\neq a}\delta(wa')$.
   Else let $\delta(wa) = \tilde{\delta}(wa)$.

2. Find a $w'\neq w$ such that
$$\pi^n(w,w') > \sum_{a_1,a_2\in\mathcal{A}} \mu^{n+1}(wa_1, w'a_2). \tag{29}$$

   Now, for any $a'\in\mathcal{A}$, let
$$\psi(a') = p_2^{n+1}(w'a') - p_1^{n+1}(w'a') - \sum_{\substack{w''\in\mathcal{A}^n\\ w''\neq w'}}\sum_{a_1\in\mathcal{A}} \mu^{n+1}(w''a_1, w'a') \tag{30}$$

   Then, find $a'\in\mathcal{A}$ such that $\psi(a')>0$.
   Now, if $\delta(wa) > \psi(a')$, then:

   - Update $\mu(wa, w'a') \leftarrow \psi(a')$
   - Update $\delta(wa) \leftarrow \delta(wa) - \psi(a')$
   - Return to 2.

   Else, update $\mu(wa, w'a') \leftarrow \delta(wa)$.

We claim that, in the procedure above, there always exists such a w' for a given wa. Otherwise,

$$\sum_{w'\neq w} \pi^n(w,w') < p_1^n(w) - p_2^n(w), \tag{31}$$

which is impossible by Lemma 1. Also, we claim that there also always exists such a' for a given wa and w'. Otherwise, for all $a' \in \mathcal{A}$,

$$\sum_{a' \in \mathcal{A}} \sum_{\substack{w'' \in \mathcal{A}^n \\ w'' \neq w'}} \sum_{a_1 \in \mathcal{A}} \mu^{n+1}(w''a_1, w'a') = \sum_{a' \in \mathcal{A}} p_2^{n+1}(w'a') - p_1^{n+1}(w'a'), \tag{32}$$

which means by construction that

$$\sum_{a' \in \mathcal{A}} \sum_{\substack{w'' \in \mathcal{A}^n \\ w'' \neq w'}} \sum_{a_1 \in \mathcal{A}} \pi^{n+1}(w''a_1, w'a') > \sum_{a' \in \mathcal{A}} p_2^{n+1}(w'a') - p_1^{n+1}(w'a'), \tag{33}$$

which is $p_2^n(w) - p_1^n(w) > p_2^n(w) - p_1^n(w)$ by Lemma 2. Moreover, by construction we have that, for all $w \neq w'$,

$$\pi^n(w, w') = \sum_{a_1, a_2 \in \mathcal{A}^n} \mu^{n+1}(wa_1, w'a_2). \tag{34}$$

Now, we construct the diagonal blocks $\mu^{n+1}(wa_1, wa_2)$. For each $w \in \mathcal{A}^n$ and $a \in \mathcal{A}$, let

$$\begin{aligned} \tilde{p}_1^{n+1}(wa) &= p_1^{n+1}(wa) - \sum_{w' \neq w} \sum_{a \in \mathcal{A}} \mu^{n+1}(wa, w'a'), \\ \tilde{p}_2^{n+1}(wa) &= p_2^{n+1}(wa) - \sum_{w' \neq w} \sum_{a \in \mathcal{A}} \mu^{n+1}(w'a', wa). \end{aligned} \tag{35}$$

Now, for a given w, let us solve the following balanced optimal transport problem:

$$\begin{aligned} &\inf_{\mu^{n+1}} 2^{-(n+1)} \sum_{\substack{a_1, a_2 \in \mathcal{A} \\ a_1 \neq a_2}} \mu^{n+1}(wa_1, wa_2) \\ &\text{s.t. } \forall a_1 \in \mathcal{A} : \sum_{a_2} \mu^{n+1}(wa_1, wa_2) = \tilde{p}_1^{n+1}(wa_1), \\ &\qquad \forall a_2 \in \mathcal{A} : \sum_{a_1} \mu^{n+1}(wa_1, wa_2) = \tilde{p}_2^{n+1}(wa_2). \end{aligned} \tag{36}$$

Following the definition of $\tilde{p}$ and $\tilde{q}$, this is a balanced optimal transport whose trivial solution is given by

$$2^{-(n+1)} \left(r(w) - \sum_{a \in \mathcal{A}} r(wa) \right). \tag{37}$$

Now we conclude the proof. By (34) and (35), μ^{n+1} is a coupling of p_1^{n+1} and p_2^{n+1}. Indeed it is positive, and for any $w_1 \in \mathcal{A}^n$ and $a_1 \in \mathcal{A}$,

$$\begin{aligned} &\sum_{w_2 \in \mathcal{A}^n} \sum_{a_2 \in \mathcal{A}} \mu^{n+1}(w_1a_1, w_2a_2) \\ &= \sum_{a_2 \in \mathcal{A}} \mu^{n+1}(w_1a_1, w_1a_2) + \sum_{\substack{w_2 \in \mathcal{A}^n \\ w_2 \neq w_1}} \sum_{a_2 \in \mathcal{A}} \mu^{n+1}(w_1a_1, w_2a_2) \\ &= \tilde{p}_1^{n+1}(w_1a_1) + \left(p_1^{n+1}(w_1a_1) - \tilde{p}_1^{n+1}(w_1a_1) \right) \\ &= p_1^{n+1}(w_1a_1), \end{aligned} \tag{38}$$

and similarly for p_2^{n+1}. Finally,

$$\begin{aligned} &\sum_{w_1, w_2 \in \mathcal{A}^n} \sum_{a_1, a_2 \in \mathcal{A}} d_B(w_1a_1, w_2a_2) \mu^{n+1}(w_1a_1, w_2a_2) \\ &= \sum_{w_1 \neq w_2} d_B(w_1, w_2) \sum_{a_1, a_2} \mu^{n+1}(w_1a_1, w_2a_2) + 2^{-(n+1)} \sum_{w} \sum_{\substack{a_1, a_2 \\ a_1 \neq a_2}} \mu^{n+1}(wa_1, wa_2). \end{aligned} \tag{39}$$

By (34), the first term is K^n, and by (37), the second term is

$$\sum_{w\in\mathcal{A}^n}\left[r(w)-\sum_{a\in\mathcal{A}}r(wa)\right]. \tag{40}$$

This provides an upper bound on K^{n+1}, and the proof is completed. ■

Theorem 1 allows one to argue that Algorithm 1 computes efficiently the Kantorovich metric between p_1^n and p_2^n.

Algorithm 1 KANT(k,m,w,n)

for $i=1,\dots,|\mathcal{A}|$ **do**
  Compute $p_1^n(wa_i)$ and $p_2^n(wa_i)$ (see Remark 1)
  $r_i \leftarrow \min\{p_1^n(wa_i), p_2^n(wa_i)\}$
RES $= 2^{-(k+1)}(m-\sum_{i=1,\dots,|\mathcal{A}|} r_i)$
if $k+1=n$ **then**
  return RES
for $i=1,\dots,|\mathcal{A}|$ **do**
  if $r_i\neq 0$ **then**
    RES $\leftarrow$ RES + KANT$(k+1, r_i, wa_i, n)$
return RES

Corollary 1. *Let* KANT *be the algorithm described in Algorithm 1, then*

$$K(p_1^n,p_2^n)=\text{KANT}(0,1,\Lambda,n). \tag{41}$$

Moreover the latter terminates in less than $\mathcal{O}(|\mathcal{S}|^2|\mathcal{A}|^{n+1})$ *operations.*

Proof. We will prove that (41) holds by induction on the level of the execution tree of Algorithm 1. Let us prove that case $n=1$. The constraints (6) imply that

$$\sum_{a_1,a_2\in\mathcal{A}}\pi^1((a_1),(a_2))=1. \tag{42}$$

Therefore, by Lemma 1,

$$\begin{aligned} K(p_1^1,p_2^1) &= 2^{-1}\sum_{\substack{a_1,a_2\in\mathcal{A}\\ a_1\neq a_2}}\pi^1((a_1),(a_2)) \\ &= 2^{-1}\left[1-\sum_{a\in\mathcal{A}}\pi^1((a),(a))\right] \\ &= 2^{-1}\left[1-\sum_{a\in\mathcal{A}}\min\{p_1^1((a)),p_2^1((a))\}\right], \end{aligned} \tag{43}$$

which is the result of KANT$(0,1,\Lambda,1)$. Now, assume that (41) holds for n. By Theorem 1,

$$K(p_1^{n+1},p_2^{n+1})=\text{KANT}(0,1,\Lambda,n)+2^{-(n+1)}\sum_{w\in\mathcal{A}^n}\left[\pi^n(w,w)-\sum_{a\in\mathcal{A}}\pi^{n+1}(wa,wa)\right]. \tag{44}$$

Following the notations of Algorithm 1, let $m^w=\min\{p_1^n(w),p_2^n(w)\}$, and let $r_i^w=\min\{p_1^{n+1}(wa_i),p_2^{n+1}(wa_i)\}$. One can re-write (44) as

$$K(p_1^{n+1},p_2^{n+1})=\text{KANT}(0,1,\Lambda,n)+\sum_{w\in\mathcal{A}^n}2^{-(n+1)}\left[m^w-\sum_{i=1,\dots,|\mathcal{A}|}r_i^w\right]. \tag{45}$$

One can recognize $\textsc{Kant}(0, 1, \Lambda, n+1)$ in the right hand side of the equation above, and the proof of (41) is completed.

In terms of computational complexity, the bottleneck of Algorithm 1 is the computation of p_1^n and p_2^n at each node of the execution tree. Following Remark 1, this can be done in $\mathcal{O}(|\mathcal{S}|^2)$ operations. Since there are $\mathcal{O}(|\mathcal{A}|^{n+1}|)$ nodes in the execution tree, the total number of operations is $\mathcal{O}(|\mathcal{S}|^2|\mathcal{A}|^{n+1})$. ■

2.3 A metric between Markov chains

Let Σ_1 and Σ_2 be two Markov chains defined on the same alphabet $\mathcal{A}$. For a fixed value of n, they respectively generate p_1^n and p_2^n on $(\mathcal{A}^n, d_B)$ as described in (2). We are now able to define a notion of metric between Markov chains, with the presented Kantorovich metric, as follows:

$$d(\Sigma_1, \Sigma_2) = \lim_{n\to\infty} K(p_1^n, p_2^n). \tag{46}$$

Remark 3. The Baire's distance 2^{-l} can be interpreted as a discount factor. Therefore, the metric $d(\Sigma_1, \Sigma_2)$, if well-defined, can be interpreted as a discounted measure of the difference between the behaviours $\mathcal{B}(\Sigma_1)$ and $\mathcal{B}(\Sigma_2)$.

Theorem 2. *The metric $d(\Sigma_1, \Sigma_2)$ is well-defined. Moreover, for any $n \geq 1$,*

$$0 \leq d(\Sigma_1, \Sigma_2) - K(p_1^n, p_2^n) \leq 2^{-n}. \tag{47}$$

Proof. For the sake of clarity, let us note $K_n = K(p_1^n, p_2^n)$. First, we prove that, for $n \geq 1$,

$$0 \leq K_{n+1} - K_n \leq 2^{-n}. \tag{48}$$

Following Lemma 1 and Theorem 1, we have to prove that

$$0 \leq \sum_{w\in\mathcal{A}^n} \left[r(w) - \sum_{a\in\mathcal{A}} r(wa) \right] \leq 1, \tag{49}$$

where $r(w) = \min\{p_1^n(w), p_2^n(w)\}$, and $r(wa) = \min\{p_1^{n+1}(wa), p_2^{n+1}(wa)\}$. Since, by the law of total probability, the following holds

$$p_1^{n+1}(wa) = \sum_{a\in\mathcal{A}} p_1^n(w), \tag{50}$$

and similarly for $p_2^n(w)$, then

$$0 \leq r(w) - \sum_{a\in\mathcal{A}} r(wa) \leq r(w), \tag{51}$$

which implies (49). Now, (48) implies that the sequence $(K_n)_{n\geq 1}$ is monotone, and bounded since

$$\lim_{n\to\infty} K_n \leq \sum_{n\geq 1} (K_{n+1} - K_n) \leq \sum_{n\geq 1} 2^{-n} = 1. \tag{52}$$

Therefore, by the monotone convergence theorem, the limit exists and is

$$\lim_{n\to\infty} K_n = \sup_{n\geq 1} K_n. \tag{53}$$

Moreover, since K_n is a distance for every $n \geq 1$, and that the limit exists, then $\lim_{n\to\infty} K_n$ is also a distance. Finally, by (48),

$$\lim_{n\to\infty} K_n - K_p \leq \sum_{n\geq p} K_n = 2^{-p}, \tag{54}$$

for any $p \geq 1$, which concludes the proof. ■

Theorem 2 provides a guarantee on the approximation of $d(\Sigma_1, \Sigma_2)$ that we will be able to compute. Indeed, for any $\varepsilon > 0$,

$$0 \leq d(\Sigma_1, \Sigma_2) - K(p_1^n, p_2^n) \leq \varepsilon. \tag{55}$$

provided that $n \geq \lceil \log_2(\varepsilon^{-1}) \rceil$. Following Corollary 1, for a fixed number of labels and states, this implies that an ε-solution can be found in $\mathcal{O}(\varepsilon^{-1})$ computational complexity.

3 Application: data-driven model abstractions

We will now present a method using the metric $d(\Sigma_1, \Sigma_2)$ as a tool to construct abstractions based on adaptive refinements of the state-space.

3.1 Abstractions with adaptive refinement

In this section, we introduce a new abstraction based on adaptive refinements. We introduce a dynamical system, noting that, in the context of this work, we restrict ourselves to deterministic models.

Definition 4 (Dynamical system)**.** A dynamical system is the 4-tuple $S = (X, \mathcal{A}, F, H)$ that defines the relation

$$\begin{cases} x_{k+1} = F(x_k) \\ y_k = H(x_k), \end{cases} \tag{56}$$

where $X \subseteq \mathbb{R}^d$ is the state space, $\mathcal{A}$ is a finite alphabet called the output space, $F : X \to X$ is a transition function, and $H : X \to \mathcal{A}$ is the output function. The variables x_k and y_k are called the state and the output at time k.

Also, in parallel to the definition of behaviour of a Markov chain, we define the behaviour of a dynamical system $\mathcal{B}(S) \subseteq \mathcal{A}^*$ as follows. A sequence $w^* = (a_1, a_2, \dots) \in \mathcal{B}(S)$ if there exists $x_1, x_2, \dots \in X$ such that $x_{i+1} = F(x_i)$ and $H(x_i) = a_i$. Also, in parallel to equivalent classes (1) on Markov chains, we define equivalent classes of sequence of labels on the continuous state space X. A subset of states is an equivalent class if it satisfies the recursive relation

$$\begin{aligned} &[wa]_S = \{x \in [w]_S \,|\, H^n(x) = a\}, \\ &[\Lambda]_S = X, \end{aligned} \tag{57}$$

for any $w \in \mathcal{A}^n$ and $a \in \mathcal{A}$. In other words, for a given sequence $w = (a_1, \dots, a_n)$, a state $x \in [w]_S$ if $H(x) = a_1$, $H(F(x)) = a_2$, ..., and $H(F^{n-1}(x)) = a_n$. In this work, we only consider dynamical systems satisfying the following assumption.

Assumption 1. The dynamical system S is such that, for any $w \in \mathcal{A}^n$ and $a \in \mathcal{A}$, the following two conditions hold:

- If $\lambda_X([w]_S) = 0$, $[w]_S = \emptyset$.
- If $\lambda_X([wa]_S) = \lambda_X([w]_S)$, then $[w]_S = [wa]_S$.

We claim that Assumption 1 is not restrictive in a data-driven context: indeed, most sample-based method are not suitable for systems that do not satisfy it. Let $[wa]_S \subset [w]_S$ be such that $\lambda_X([wa]_S) = \lambda_X([w]_S)$, then one will never sample the points in $[w]_S \setminus [wa]_S$, which has zero measure, and will therefore never be able to capture such pathological behaviour.

Definition 5 (Adaptive partitioning)**.** Let $w_1 \in \mathcal{A}^{n_1}$, $w_2 \in \mathcal{A}^{n_2}$, ..., $w_k \in \mathcal{A}^{n_k}$ be k sequences of labels of different lengths. The set of sequences $\mathcal{W} = \{w_i\}_{i=1,\dots,k}$ is an adaptive paritioning for S if

$$\bigcup_{w \in \mathcal{W}} [w]_S = X, \tag{58}$$

$$\forall i \neq j,\ [w_i]_S \cap [w_j]_S = \emptyset. \tag{59}$$

We now introduce an abstraction procedure based on an adaptive partitioning refinements.

Definition 6 (Abstraction based on adaptive refinements)**.** Let $S = (X, \mathcal{A}, F, H)$ be a dynamical system, and let $\mathcal{W}$ be an adaptive partitioning for S. Then the corresponding abstraction based on adaptive refinements is the Markov chain $\Sigma_{\mathcal{W}} = (\mathcal{S}, \mathcal{A}, P, \mu, L)$ defined as follows:

- The states are the partitions, that is $\mathcal{S} = \mathcal{W}$.
- μ_w is the Lebesgue measure of the partition $[w]_S$ on X, that is

$$\mu_w = \lambda_X([w]_S). \tag{60}$$

- For $w_1 = (a_1, \ldots, a_{n_1})$, and $w_2 = (b_1, \ldots, b_{n_2})$, let $k = \min\{n_1 - 1, n_2\}$, $w_1' = (a_2, \ldots, a_{k+1})$, and $w_2' = (b_1, \ldots, b_k)$. If $w_1' \neq w_2'$ or $\lambda_X([w_1]_S) = 0$, then $P_{w_1,w_2} = 0$. Else

$$P_{w_1,w_2} = \frac{\lambda_X([a_1 w_2]_S)}{\lambda_X([w_1]_S)}. \tag{61}$$

- For $w = (a_1, \ldots, a_n)$, $L(w) = a_1$.

For a given adaptive partitioning $\mathcal{W}$, the abstraction $\Sigma_{\mathcal{W}}$ can be interpreted as follows. The initial probability to be in w is the proportion of $[w]_S$ in X, and the probability to jump from w_1 to w_2 is the proportion of $[w_1]_S$ that goes into $[w_2]_S$ given the dynamics.

Proposition 1. *Given a dynamical system S satisfying Assumption 1, consider abstraction $\Sigma_{\mathcal{W}}$. If for all $w_1, w_2 \in \mathcal{W}$, $P_{w_1,w_2} \in \{0, 1\}$, then $\mathcal{B}(\Sigma_{\mathcal{W}}) = \mathcal{B}(S)$.*

Proof. We first prove that, if there are $w_1, w_2 \in \mathcal{W}$ such that $P_{w_1,w_2} = 1$, then

$$F([w_1]_S) \subseteq [w_2]_S. \tag{62}$$

Let w_1, w_2, k, w_1' and w_2' be as in Definition 6. Let us note that

$$\begin{aligned} F([w_1]_S) &= \left\{ F(x) \in X \;\middle|\; \begin{array}{l} H(x) = a_1, \\ H(F(x)) = a_2 \\ \ldots \\ H(F^{n-1}(x)) = a_{n_1} \end{array} \right\} \\ &= F([(a_1)]_S) \cap [(a_2, \ldots, a_{n_1})]_S \end{aligned} \tag{63}$$

Now, since $P_{w_1,w_2} > 0$, then $w_1' = w_2'$. There are two cases, either $w_1' = (a_2, \ldots, a_{k+1})$ and $w_2' = w_2$, or $w_1' = w_1$ and $w_2' = (b_1, \ldots, b_k)$. Let us investigate these separately. In the first case, $(a_2, \ldots, a_{k+1}) = w_2$. By definition,

$$[(a_2, \ldots, a_{n_1})]_S \subseteq [(a_2, \ldots, a_{k+1})]_S = [w_2]_S. \tag{64}$$

Therefore (63) implies

$$F([w_1]_S) \subseteq F([(a_1)]_S) \cap [w_2]_S \subseteq [w_2]_S, \tag{65}$$

which is (62). In the second case, assume that

$$[w_1]_S = [a_1 w_2]_S, \tag{66}$$

then

$$\begin{aligned} F([w_1]_S) &= F([a_1 w_2]_S) \\ &= F([(a_1)]_S) \cap [w_2]_S \\ &\subseteq [w_2]_S, \end{aligned} \tag{67}$$

where a very similar as in (63) was used. It remains to show that (66) holds. Since we are in the second case cited above, then

$$a_1 w_2 = w_1(b_{k+1}, \dots, b_{n_2}), \tag{68}$$

which implies that $[a_1 w_2]_S \subseteq [w_1]_S$. Moreover, $P_{w_1,w_2} = 1$, that is

$$\lambda_X([a_1 w_2]_S) = \lambda_X([w_1]_S). \tag{69}$$

Following Assumption 1, it means that $[a_1 w_2]_S = [w_1]_S$, which proves (62).

Now we prove that (62) implies $\mathcal{B}(\Sigma_{\mathcal{W}}) = \mathcal{B}(S)$. We first prove that $\mathcal{B}(S) \subseteq \mathcal{B}(\Sigma_{\mathcal{W}})$. Let $w^* = (a_1, a_2, \dots) \in \mathcal{A}^*$ such that $w^* \notin \mathcal{B}(\Sigma_{\mathcal{W}})$, then there are a_i, a_{i+1} such that, for all $w_1, w_2 \in \mathcal{W}$ for which $L(w_1) = a_i$ and $L(w_2) = a_{i+1}$, $P_{w_1,w_2} = 0$. This could mean three things.

1. For all such w_1, $\lambda_X([(a_i)]_S) = 0$. Following Assumption 1, it means that $[(a_i)]_S = \emptyset$.
2. Let $w_2 = (a_{i+1}, b_2, \dots, b_{n_2})$. For all such w_2,

$$\lambda_X([(a_i, a_{i+1}, b_2, \dots, b_{n_2})]_S) = 0, \tag{70}$$

   which means by Assumption 1 that

$$[(a_i, a_{i+1}, b_2, \dots, b_{n_2})]_S = \emptyset. \tag{71}$$

   By Definition 5, it implies that $[a_i a_{i+1}]_S = \emptyset$.

In any case, it means that $w^* \neq \mathcal{B}(S)$. Now we prove $\mathcal{B}(\Sigma_W) \subseteq \mathcal{B}(S)$. Let $w^* = (a_1, a_2, \dots) \in \mathcal{B}(\Sigma_{\mathcal{W}})$. It means that there exists $w_1, w_2, \dots \in \mathcal{W}$ such that $L(w_i) = a_i$ and $P_{w_i,w_{i+1}} = 1$. Following (62), this implies that $F([w_i]_S) \subseteq [w_{i+1}]_S$. This implies that $[a_i a_{i+1}]_S \neq \emptyset$, which means $w^* \in \mathcal{B}(S)$. ■

3.2 A data-driven abstraction

In this section, we investigate a method to construct an abstraction based on adaptive refinements, from a data set comprising outputs sampled from the dynamical model S. Given an adaptive partitioning $\mathcal{W}$, we propose to construct $\Sigma_{\mathcal{W}}$ using empirical probabilities (see [14] for more details). In the context of this work, we consider that the number of samples is large enough to assume the following.

Assumption 2. For any abstraction $\Sigma_{\mathcal{W}} = (\mathcal{W}, \mathcal{A}, P, \mu, L)$, the transition probabilities P and the initial distribution μ are known exactly.

Now we are able to use the tool investigated in Section 2 to find a smart adaptive partitioning. Indeed, one can construct two abstractions $\Sigma_{\mathcal{W}_1}$ and $\Sigma_{\mathcal{W}_2}$ corresponding to two different partitioning, and efficiently compute the Kantorovich metric $d(\Sigma_{\mathcal{W}_1}, \Sigma_{\mathcal{W}_2})$ up to some accuracy ε following Corollary 1. This gives a discounted measure of the difference between $\mathcal{B}(\Sigma_{\mathcal{W}_1})$ and $\mathcal{B}(\Sigma_{\mathcal{W}_2})$ (see Remark 3). This reasoning leads to the greedy procedure Refine(S, N, ε) described in Algorithm 2.

An interpretation of Algorithm 2 goes as follows. Let $\mathcal{W}$ be a coarse partitioning, and $\mathcal{W}_1'$ and $\mathcal{W}_2'$ be two more refined partitionings. If $d(\Sigma_{\mathcal{W}}, \Sigma_{\mathcal{W}_1'}) > d(\Sigma_{\mathcal{W}}, \Sigma_{\mathcal{W}_2'})$, then one could argue that it is more interesting to choose $\mathcal{W}_1'$ over $\mathcal{W}_2'$, since the discounted measure between the behaviours corresponding to the coarse partitioning and the refined partitioning is larger. Moreover, if at some point $\Sigma_{\mathcal{W}}$ is such that $P_{w,w'} \in \{0, 1\}$, then one has a sufficient condition to stop the algorithm following Proposition 1, otherwise the algorithm stops after N iterations. An execution step of the algorithm can be found in Figure 2.

Corollary 2. *The algorithm* Refine(S, N, ε) *terminates in* $\mathcal{O}(|\mathcal{A}|^{n+4} N^4)$ *operations, with* $n = \lceil \log_2(\varepsilon^{-1}) \rceil$. *Moreover, for* S *satisfying Assumption 1, if* $\Sigma_{\mathcal{W}} =$ Refine(S, ∞, ε) *terminates, then* $\mathcal{B}(\Sigma_{\mathcal{W}}) = \mathcal{B}(S)$.

Algorithm 2 REFINE(S, N, ε)

$\mathcal{W} \leftarrow \{(a)\}_{a\in\mathcal{A}}$
Construct $\Sigma_{\mathcal{W}}$ from samples of S
while $\exists w_1, w_2 \in \mathcal{W} : P_{w_1,w_2} \in (0,1)$ **do**
  if $N = 0$ **then**
    return $\Sigma_{\mathcal{W}}$
  for $i = 1, \ldots, |\mathcal{W}|$ **do**
    $\mathcal{W}'_i \leftarrow \mathcal{W} \setminus \{w_i\}$
    $\mathcal{W}'_i \leftarrow \mathcal{W}'_i \cup \{w_i a\}_{a\in\mathcal{A}}$
    Construct $\Sigma_{\mathcal{W}'_i}$ from samples of S
    $d_i \leftarrow d(\Sigma_{\mathcal{W}}, \Sigma_{\mathcal{W}'_i})$ with precision ε
  $j = \arg\max_{i=1,\ldots,|\mathcal{W}|} d_i$
  $\mathcal{W} \leftarrow \mathcal{W}'_j$
  $\Sigma_{\mathcal{W}} \leftarrow \Sigma_{\mathcal{W}'_j}$
  $N \leftarrow N - 1$
return $\Sigma_{\mathcal{W}}$

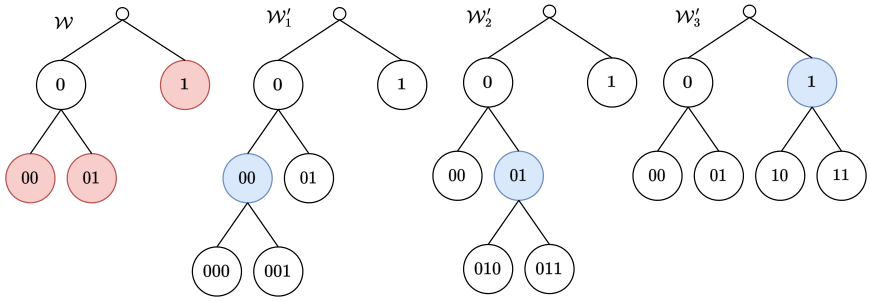

Figure 2: Illustration of the execution of Algorithm 2. Suppose that at some point $\mathcal{W} = \{00, 01, 1\}$, with the corresponding abstraction $\Sigma_{\mathcal{W}}$. Then the algorithm will explore the partitionings $\mathcal{W}'_1 = \{000, 001, 01, 1\}$, $\mathcal{W}'_2 = \{00, 010, 011, 1\}$ and $\mathcal{W}'_2 = \{00, 01, 10, 11\}$. For each one, it will compute $\Sigma_{\mathcal{W}'_i}$, and $d(\Sigma_{\mathcal{W}}, \Sigma_{\mathcal{W}'_i})$, and choose the one for which the distance is the largest.

Proof. Proposition 1 gives a sufficient condition to stop the algorithm, hence the second part of the claim. It remains to prove that the computational complexity is the claimed one. Let $\mathcal{W}^{(k)}$ and $\mathcal{W}_i^{'(k)}$ be the abstractions $\mathcal{W}$ and $\mathcal{W}'_i$ at iteration k in Algorithm 2. First we note that

$$\begin{aligned} |\mathcal{W}^{(k)}| &= k|\mathcal{A}| - (k-1) \\ |\mathcal{W}_i^{'(k)}| &= (k+1)|\mathcal{A}| - k \end{aligned} \tag{72}$$

At each iteration k, one has to compute $|\mathcal{W}^{(k)}|$ times the ε-accurate Kantorovich distance between two models of sizes given by (72). By Corollary 1, such computational complexity is

$$\begin{aligned} &\mathcal{O}\left(|\mathcal{W}^{(k)}|\left(|\mathcal{A}|^{n+1}\left(|\mathcal{W}^{(k)}|^2 + |\mathcal{W}_i^{'(k)}|^2\right)\right)\right) \\ =&\mathcal{O}\left(|\mathcal{A}|^{n+1}|\mathcal{W}_i^{'(k)}|^3\right) \\ =&\mathcal{O}\left(|\mathcal{A}|^{n+4}(k+1)^3\right). \end{aligned} \tag{73}$$

Now, the worst-case is when the algorithm does not converge to an abstraction where $P_{w,w'} \in \{0, 1\}$.

Therefore, the total computational complexity is

$$\begin{aligned}&\sum_{k=1}^{N} \mathcal{O}\left(|\mathcal{A}|^{n+4}(k+1)^3\right)\\ =&\,\mathcal{O}\left(|\mathcal{A}|^{n+4}\sum_{k=1}^{N}(k+1)^3\right)\\ =&\,\mathcal{O}\left(|\mathcal{A}|^{n+4}N^4\right),\end{aligned} \tag{74}$$

which is the claim. ■

3.3 Numerical examples

In this section, we demonstrate on an example that our greedy algorithm converges to a smart partitioning. We then show that one can use this method for controller design, among other applications.

Example 1. Consider $S = (X, \mathcal{A}, F, H)$ with $X = [0,2] \times [0,1]$, $\mathcal{A} = \{0,1\}$. Let F be defined as

$$F(x) = \begin{cases} x & \text{if } x \in P_1 \cup P_5, \\ (x_1/2 + 1/2, x_2 + 1/2) & \text{if } x \in P_2, \\ (x_1 - 1/2, x_2) & \text{if } x \in P_3, \\ (2x_1 + 1, 4x_2 - 3/4) & \text{else,} \end{cases} \tag{75}$$

where P_i are depicted in Figure 3, and

$$H(x) = \begin{cases} 0 & \text{if } x \in P_1, \\ 1 & \text{else.} \end{cases} \tag{76}$$

An illustration and interpretation of S is given in Figure 3.

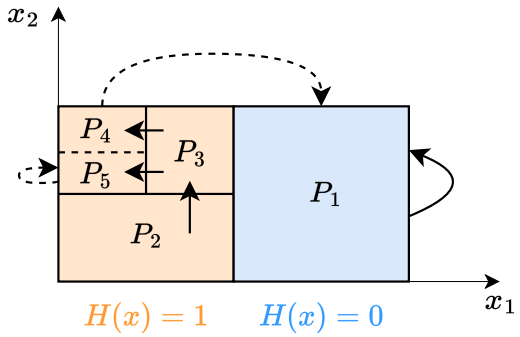

Figure 3: Illustration and description of the transition function F of Example 1. F has to be understood in the following way: P_1 is mapped to itself, P_2 is mapped to P_3, P_3 is mapped to $P_4 \cup P_5$, P_4 is mapped to P_1, and P_5 to itself.

We now show that our data-driven method refines the partitioning of Example 1 in a smart way, that is by choosing to refine the partition $[1]_S$ over $[0]_S$. Indeed refining $[0]_S$ in $[01]_S$ and $[00]_S$ would not bring more information, as $[01]_S = \emptyset$ and $[00]_S = [0]_S$. The result of the algorithm at all iterations k is depicted in Table 1.

Table 1: Results of Algorithm 2 for Example 1.

	$\mathcal{W}$	$d\left(\mathcal{W}, \mathcal{W}'_j\right)$	$P_{w,w'} \in \{0,1\}$?
$k=0$	$\{0, \mathbf{1}\}$	0.0015	No
$k=1$	$\{0, 10, \mathbf{11}\}$	0.0059	No
$k=2$	$\{0, 10, 110, \mathbf{111}\}$	0.0039	No
$k=3$	$\{0, 10, 110, 1110, 1111\}$	-	**Yes**

One can see that the expected behaviour is indeed observed: the algorithm refines the partitions leading to the output whose refinement is necessary to increase the accuracy of the abstraction compared to the true dynamical system. The algorithm stops at the third iteration since the obtained data-driven abstraction is such that $P_{w,w'} \in \{0,1\}$, which is a stopping criterion following Proposition 1. The corresponding partitioning is illustrated in Figure 4.

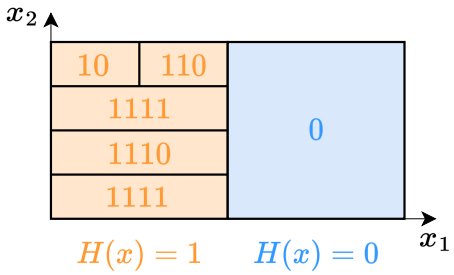

Figure 4: Illustration of the last partitioning $\mathcal{W}$ given by Algorithm 2 for Example 1.

We further demonstrate the quality of the obtained abstractions by designing a controller for a similar dynamical system.

Example 2. Consider the dynamical system S as described in Example 1, except that the dynamics is controlled as follows:

$$\tilde{x}_k = x_k + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u_k, \quad x_{k+1} = F(\tilde{x}_k) \tag{77}$$

where $u_k = K(x_k) \in \{0, 1/4, 1/2\}$ is an input to the system. Consider the reward

$$r(x) = \begin{cases} 1 & \text{if } H(x) = 0, \\ 0 & \text{else,} \end{cases} \tag{78}$$

and a discounted reward maximization objective, that is

$$\max_K \sum_k \gamma^k r(x_k), \tag{79}$$

where $\gamma = 0.95$ is a discount factor.

To solve this optimal control problem, we will use the abstractions constructed by Algorithm 2. For each partitioning $\mathcal{W}$ in Table 1, we will construct the data-driven abstraction $\Sigma^u_{\mathcal{W}}$ corresponding to the actions given in Example 2, that is $u = 0$, $u = 1/4$ and $u = 1/2$. We will then solve a Markov Decision Process (or MDP for short, see [28] for an introduction), and maximize the expected reward corresponding to these abstractions. For this, we used the implementation of the value iteration algorithm implemented in the `POMDPs.jl` Julia package [29]. The corresponding expected rewards are given in Table 2. One can see that the expected rewards increases as Algorithm 2 refines the state-space.

Table 2: MDPs expected rewards for Example 2.

$\mathcal{W}$	Expected reward
$\{0, \mathbf{1}\}$	17.0188
$\{0, 10, \mathbf{11}\}$	18.3736
$\{0, 10, 110, \mathbf{111}\}$	18.8233
$\{0, 10, 110, 1110, 1111\}$	19.0724

4 Conclusion and further research

We now summarize the contributions of this work. First, we proposed a Kantorovich metric between two Markov chains, defined on an underlying space defined by the Baire's distance. We showed that one can arbitrarily approximate this metric with an efficient algorithm, and therefore efficiently measure a discounted measure between the behaviours of two Markov chains. We then applied this metric to the construction of data-driven abstractions based on adaptive refinement, thereby reducing their size. More precisely, we propose a greedy procedure using the Kantorovich metric to assess the difference between two abstractions. We showed that, in some cases, the obtained abstraction has exactly the same behaviour as the original dynamical system. We also demonstrated that the quality of our procedure by designing a controller for the original system, from the obtained finite model.

As further research, we would like to characterize the underlying distances between sequences for which the Kantorovich metric can be efficiently computed, and interpret them. We would also like to investigate even more efficient algorithms to compute the metric, or link it with existing metrics in the litterature. Finally, we want to design a smart stopping criterion for our refinement procedure. For example, we could quantify a difference between the behaviour of the abstraction at any iteration, and the behaviour of the original system.

Acknowledgment

We thank Prof. Franck van Breugel and Prof. Prakash Panangaden for their insightful comments and the interesting conversations we had with them about this work.

References

[1] C. D. Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality, and robustness," *IEEE Transactions on Automatic Control*, vol. 65, pp. 909–924, 3 2020.

[2] Z. Wang and R. M. Jungers, "A data-driven method for computing polyhedral invariant sets of black-box switched linear systems," *IEEE Control Systems Letters*, vol. 5, pp. 1843–1848, 11 2021.

[3] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, 6 2021.

[4] A. Banse, Z. Wang, and R. M. Jungers, "Learning stability guarantees for data-driven constrained switching linear systems," 5 2022. [Online]. Available: http://arxiv.org/abs/2205.00696

[5] D. Lind and B. Marcus, *An Introduction to Symbolic Dynamics and Coding.* Cambridge University Press, 2003.

[6] A. van der Schaft, "Equivalence of dynamical systems by bisimulation," *IEEE Transactions on Automatic Control*, vol. 49, pp. 2160–2172, 2004.

[7] C. Baier and J. P. Katoen, *Principles of Model Checking.* MIT Press Books, 2008.

[8] P. Tabuada, *Verification and Control of Hybrid Systems*. Springer, 2009.

[9] M. Kwiatkowska, D. Parker, and G. Norman, "Prism 4.0: Verification of probabilistic real-time systems," 2011, pp. 1–6.

[10] C. Dehnert, S. Junges, J.-P. Katoen, and M. Volk, "A storm is coming: A modern probabilistic model checker," R. Majumdar and V. Kunčak, Eds., vol. 10427. Springer International Publishing, 2017.

[11] R. Majumdar, N. Ozay, and A. K. Schmuck, "On abstraction-based controller design with output feedback." Association for Computing Machinery, Inc, 4 2020.

[12] T. Badings, L. Romao, A. Abate, D. Parker, H. Poonawala, M. Stoelinga, and N. Jensen, "Robust control for dynamical systems with non-gaussian via formal abstractions," *Journal of Artificial Intelligence Research*, vol. 76, pp. 341–391, 2023.

[13] A. K. Schmuck and J. Raisch, "Asynchronous l-complete approximations," *Systems and Control Letters*, vol. 73, pp. 67–75, 2014.

[14] A. Banse, L. Romao, A. Abate, and R. M. Jungers, "Data-driven memory-dependent abstractions of dynamical systems," 2022. [Online]. Available: https://arxiv.org/abs/2212.01926

[15] A. Devonport, A. Saoud, and M. Arcak, "Symbolic abstractions from data: A pac learning approach," 4 2021. [Online]. Available: http://arxiv.org/abs/2104.13901

[16] R. Coppola, A. Peruffo, and M. M. Jr., "Data-driven abstraction for verification of deterministic systems (arxiv)," 2022.

[17] A. Lavaei, S. Soudjani, E. Frazzoli, and M. Zamani, "Constructing mdp abstractions using data with formal guarantees," 6 2022. [Online]. Available: http://arxiv.org/abs/2206.14402

[18] R. Baire, "Sur la représentation des fonctions discontinues: Première partie," *Acta Mathematica*, vol. 30, no. none, pp. 1–48, Jan. 1906, publisher: Institut Mittag-Leffler.

[19] Y. Deng and W. Du, "The Kantorovich Metric in Computer Science: A Brief Survey," *Electronic Notes in Theoretical Computer Science*, vol. 253, no. 3, pp. 73–82, Nov. 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1571066109004265

[20] Z. Rached, F. Alajaji, and L. Campbell, "The Kullback–Leibler Divergence Rate Between Markov Sources," *Information Theory, IEEE Transactions on*, vol. 50, pp. 917–921, Jun. 2004.

[21] S. Kiefer, "On Computing the Total Variation Distance of Hidden Markov Models," Apr. 2018, arXiv:1804.06170 [cs]. [Online]. Available: http://arxiv.org/abs/1804.06170

[22] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, "Metrics for labelled Markov processes," *Theoretical Computer Science*, vol. 318, no. 3, pp. 323–354, Jun. 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304397503006042

[23] F. van Breugel, B. Sharma, and J. Worrell, "Approximating a Behavioural Pseudometric Without Discount for Probabilistic Systems," in *Foundations of Software Science and Computational Structures*, ser. Lecture Notes in Computer Science, H. Seidl, Ed. Berlin, Heidelberg: Springer, 2007, pp. 123–137.

[24] N. Madras and D. Sezer, "Quantitative bounds for Markov chain convergence: Wasserstein and total variation distances," *Bernoulli*, vol. 16, no. 3, pp. 882–908, Aug. 2010, publisher: Bernoulli Society for Mathematical Statistics and Probability. [Online]. Available: https://projecteuclid.org/journals/bernoulli/volume-16/issue-3/Quantitative-bounds-for-Markov-chain-convergence--Wasserstein-and-total/10.3150/09-BEJ238.full

[25] D. Rudolf and N. Schweizer, "Perturbation theory for Markov chains via Wasserstein distance," *Bernoulli*, vol. 24, no. 4A, pp. 2610–2639, Nov. 2018, publisher: Bernoulli Society for Mathematical Statistics and Probability. [Online]. Available: https://projecteuclid.org/journals/bernoulli/volume-24/issue-4A/Perturbation-theory-for-Markov-chains-via-Wasserstein-distance/10.3150/17-BEJ938.full

[26] P. Bikash, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, Jan. 1989. [Online]. Available: https://www.academia.edu/1806671/A_tutorial_on_hidden_Markov_models_and_selected_applications_in_speech_recognition

[27] P. E. Bradley and A. C. Braun, "Finding the Asymptotically Optimal Baire Distance for Multi-Channel Data," *Applied Mathematics*, vol. 6, no. 3, pp. 484–495, Mar. 2015, number: 3 Publisher: Scientific Research Publishing. [Online]. Available: http://www.scirp.org/Journal/Paperabs.aspx?paperid=54511

[28] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[29] M. Egorov, Z. N. Sunberg, E. Balaban, T. A. Wheeler, J. K. Gupta, and M. J. Kochenderfer, "POMDPs.jl: A Framework for Sequential Decision Making under Uncertainty," *Journal of Machine Learning Research*, vol. 18, no. 26, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-300.html