The Complexity of Sequential Prediction in Dynamical Systems

Vinod Raman* VKRAMAN@UMICH.EDU
University of Michigan, Ann Arbor, 48104

Unique Subedi* SUBEDI@UMICH.EDU
University of Michigan, Ann Arbor, 48104

Ambuj Tewari TEWARIA@UMICH.EDU
University of Michigan, Ann Arbor, 48104

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

We study the problem of learning to predict the next state of a dynamical system when the underlying evolution function is unknown. Unlike previous work, we place no parametric assumptions on the dynamical system, and study the problem from a learning theory perspective. We define new combinatorial measures and dimensions and show that they quantify the optimal mistake and regret bounds in the realizable and agnostic settings respectively. By doing so, we find that in the realizable setting, the total number of mistakes can grow according to *any* increasing function of the time horizon T. In contrast, we show that in the agnostic setting under the commonly studied notion of Markovian regret, the only possible rates are $\Theta(T)$ and $\tilde{\Theta}(\sqrt{T})$.[1]

Keywords: Discrete-time Dynamical Systems, Online Learnability

1. Introduction

A discrete-time dynamical system is a mathematical model that describes the evolution of a system over discrete time steps. Formally, a discrete-time dynamical system is a tuple $(\mathbb{N}, \mathcal{X}, f)$, where $\mathbb{N}$ is the set of natural numbers that denote the timesteps, $\mathcal{X}$ is a non-empty set called the state space, and $f : \mathcal{X} \to \mathcal{X}$ is a deterministic map that describes the evolution of the state. Dynamical systems have been widely used in practice due to their ability to accurately model natural phenomena. For instance, boolean networks are an important class of discrete-time, discrete-space dynamical systems with widespread applicability to genetic modeling (Kauffman, 1969; Shmulevich et al., 2002). In a boolean network, the state space is $\mathcal{X} = \{0,1\}^n$ with $|\mathcal{X}|^{|\mathcal{X}|} = (2^n)^{2^n}$ possible evolution functions. For genetic modeling, n is taken to be the number of genes and $x \in \mathcal{X}$ indicates the expression of all n genes under consideration. As an example, "1" could represent the gene with a high concentration of a certain protein, and "0" could represent the gene with a low concentration. With such formulation, one can study how the state of these genes evolves over time under certain medical interventions. Beyond genetics, dynamical systems have been used in control (Li et al., 2019), computer vision (Doretto et al., 2003), and natural language processing (Sutskever et al., 2014; Belanger and Kakade, 2015).

In this work, we consider the problem of predicting the next state of a dynamical system when the underlying evolution function is unknown (Ghai et al., 2020). To capture the sequential nature of dynamical systems, we consider the model where the learner plays a sequential game with nature

* Equal contribution
1. The full paper is available at: https://arxiv.org/abs/2402.06614.

over T rounds. At the beginning of the game, nature reveals the initial state $x_0 \in \mathcal{X}$. In each round $t \in [T]$, the learner makes its prediction of the next state $\hat{x}_t \in \mathcal{X}$, nature reveals the true next state $x_t \in \mathcal{X}$, and the learner suffers loss $\mathbb{1}\{\hat{x}_t \neq x_t\}$. Given an evolution class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$, the goal of the learner is to output predictions of the next state such that its regret, the difference between its cumulative mistakes and the cumulative mistakes of the best-fixed evolution in hindsight (defined formally in Section 2.2), is small. The class $\mathcal{F}$ is said to be learnable if there exists a learning algorithm whose regret is a sublinear function of the time horizon T.

Although we allow the state space $\mathcal{X}$ to be arbitrary, we only consider the 0-1 loss, which may be more appropriate for *discrete-space* dynamical systems. However, even discrete-space dynamical systems can be very expressive, capturing complex processes like cellular automata (Hoekstra et al., 2010; Wolfram, 1986) and language modeling (Elman, 1995). For example, let $\mathcal{V}$ be a countable token space and $\mathcal{X} = \mathcal{V}^\star$ to be the state space containing all finite sequences of elements from $\mathcal{V}$. Consider a function class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ with the following property: for every $f \in \mathcal{F}$ and any input $x \in \mathcal{X}$, $f(x) = x \circ v$, where we use $\circ$ to represent the concatenation operator. Then, $\mathcal{F}$ is a set of auto-regressive models which given a sequence of tokens $x \in \mathcal{X}$, predicts the next token $v \in \mathcal{V}$ in the sequence. In particular, $\mathcal{F}$ could be a class of language models which use deterministic decoding strategies (e.g. greedy decoding) to output the next token (Chorowski and Jaitly, 2016). For such a function class $\mathcal{F}$, one might be interested in understanding the optimal number of mistakes for next-token prediction when the true sequence of tokens is generated by some unknown $f \in \mathcal{F}$.

Given any learning problem $(\mathcal{X}, \mathcal{F})$, we aim to find necessary and sufficient conditions for the learnability of $\mathcal{F}$ while also quantifying the minimax rates under two notions of expected regret: Markovian and Flow regret (Equations 1 and 2 respectively). To that end, our main contributions are summarized below.

(i) We provide a quantitative characterization of learnability in the realizable setting, when there exists an evolution function in the class $\mathcal{F}$ that generates the sequence of states. Our characterization is in terms of a new combinatorial complexity measure we call the Evolution complexity. Using this characterization, we show that all rates are possible for the minimax expected mistakes when learning dynamical systems in the realizable setting. This is in contrast to online multiclass classification where only two rates are possible. Finally, we compare realizable learnability of dynamical systems to realizable learnability in PAC and online classification.

(ii) In the agnostic setting, we lower and upper bound the minimax Markovian regret in terms of the Littlestone dimension. This result shows that the finiteness of the Littlestone dimension characterizes learnability under Markovian regret. As a corollary, we establish a separation between realizable and agnostic learnability under Markovian regret. We show this separation between realizable and agnostic learnability continues to hold when considering Flow regret. However, if the evolution class has uniformly bounded projections, we show that realizable and agnostic learnability under Flow regret are equivalent.

Our characterization of realizable learnability in terms of the Evolution complexity follows from standard techniques in online classification. However, our results showing all possible rates requires a careful construction of a family of classes which have not been studied in learning theory. Likewise, our comparisons of realizable learnability require a careful construction of non-trivial classes, and computing their combinatorial dimensions.

In the agnostic setting, our upper bound on the minimax Markovian regret in terms of the Littlestone dimension results from reducing learning dynamical systems to online multiclass classification. However, both of the lower bounds $\Omega(\sqrt{T})$ and $\Omega(\mathrm{L}(\mathcal{H}))$ in Theorem 15 are not standard. The lower bound of $\Omega(\mathrm{L}(\mathcal{H}))$ requires constructing a hard stream by traversing down a Littlestone tree *skipping* certain levels. The lower bound of $\sqrt{T}$ requires constructing a hard stream using two different evolution functions f_1, f_2 by: (a) setting x_0 to be a state they differ on (b) generating their trajectories starting from x_0 and finally (c) picking states from each trajectory in an alternating fashion. These arguments are different from the typical lower bound construction for online classification. In addition, the construction in Theorem 18 showing the separation between realizable and agnostic learnability under Flow regret is non-trivial. It involves constructing an $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ so that on a large subset of states in $\mathcal{X}$, every function $f \in \mathcal{F}$ effectively reveals itself.

1.1. Related Works

There has been a long line of work studying prediction and regret minimization when learning unknown dynamical systems (Hazan et al., 2017, 2018; Ghai et al., 2020; Lee, 2022; Rashidinejad et al., 2020; Kozdoba et al., 2019; Tsiamis and Pappas, 2022; Lale et al., 2020). However, these works focus on prediction for fully/partially-observed linear dynamical systems under various data-generating processes. Moreover, there is also a line of work on regret minimization for linear dynamical systems for online control problems (Abbasi-Yadkori and Szepesvári, 2011; Cohen et al., 2018; Agarwal et al., 2019). For non-linear dynamical systems, there has been some applied work studying data-driven approaches to prediction (Wang et al., 2016; Korda and Mezić, 2018; Ghadami and Epureanu, 2022). Regret minimization for non-linear dynamical systems has mainly been studied in the context of control (Kakade et al., 2020; Muthirayan and Khargonekar, 2021). Another related line of work is online learning for time series forecasting, where autoregressive models are used to predict the next state (Anava et al., 2013, 2015; Liu et al., 2016; Yang et al., 2018).

Another important line of work is that of system identification and parameter estimation (Åström and Eykhoff, 1971; Ljung, 1999). Here, the goal is to recover and estimate the parameters of the underlying evolution function from the observed sequence of states. There is a long history of work studying system identification in both the batch (Campi and Weyer, 2002; Vidyasagar and Karandikar, 2006; Foster et al., 2020; Sattar and Oymak, 2022; Bahmani and Romberg, 2020) and streaming settings (Kowshik et al., 2021a,b; Giannakis et al., 2023; Jain et al., 2021). Several works have also considered the problem of learning the unknown evolution rule of dynamical systems defined over discrete state spaces. For example, Wulff and Hertz (1992) trains a neural network on a sequence of observed states to approximate the unknown evolution rule of a cellular automaton. Grattarola et al. (2021) extends this work to learning the unknown evolution rule of a graph cellular automaton, a generalization of a regular cellular automaton, using graph neural networks. (Qiu et al.; Rosenkrantz et al., 2022) also consider the problem of PAC learning a discrete-time, finite-space dynamical system defined over a (un)directed graph. We also note the related work of Berry and Das (2023, 2025), who study the problem of learning dynamics observed through a continuous embedding and derive learning guarantees based on various structural properties of the underlying system. Finally, our work builds on a rich tradition in learning theory that characterizes learnability through complexity measures and combinatorial dimensions (Vapnik and Chervonenkis, 1971; Littlestone, 1987; Bartlett and Mendelson, 2002; Daniely et al., 2011).

2. Preliminaries

2.1. Discrete-time Dynamical Systems

A discrete-time dynamical system is a tuple $(\mathbb{N}, \mathcal{X}, f)$, where $\mathbb{N}$ is the set of natural numbers denoting the time steps, $\mathcal{X}$ is a non-empty set called the *state space*. In this work, we make no assumption on the cardinality of $\mathcal{X}$, so it can be unbounded and perhaps even uncountable. The function $f : \mathcal{X} \to \mathcal{X}$ is a deterministic map that defines the evolution of the dynamical system. That is, the $(t+1)$-th iterate of the dynamics can be expressed in terms of t-th iterate using the relation $x_{t+1} = f(x_t)$. Define f^t to be the t-fold composition of f. That is, $f^2 = f \circ f$, $f^3 = f \circ f \circ f$, and so forth. Given an initial state $x_0 \in \mathcal{X}$, the sequence $\{f^t(x_0)\}_{t\in\mathbb{N}}$ is called the *flow* of the dynamical system through x_0. Finally, let $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ denote a class of evolution functions on the state space $\mathcal{X}$ and $\mathcal{F}(x) = \{f(x) \mid f \in \mathcal{F}\} \subseteq \mathcal{X}$ to be the projection of $\mathcal{F}$ onto $x \in \mathcal{X}$.

2.2. Learning-to-Predict in Dynamical Systems

When learning-to-predict in dynamical systems, nature plays a sequential game with the learner over T rounds. At the beginning of the game, nature reveals the initial state $x_0 \in \mathcal{X}$. In each round $t \in [T]$, the learner $\mathcal{A}$ uses the observed sequence of states $x_{<t} := (x_0, ..., x_{t-1})$ to predict the next state $\mathcal{A}(x_{<t}) \in \mathcal{X}$. Nature then reveals the true state $x_t \in \mathcal{X}$, and the learner suffers the loss $\mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\}$. Given a class of evolution functions $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$, the goal of the learner is to make predictions such that its *regret*, defined as a difference between cumulative loss of the learner and the best possible cumulative loss over evolution functions in $\mathcal{F}$, is small.

Formally, given $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$, the expected Markovian regret of an algorithm $\mathcal{A}$ is defined as

$$\mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) := \sup_{(x_0, x_1, ..., x_T)} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\}\right] - \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \mathbb{1}\{f(x_{t-1}) \neq x_t\}, \tag{1}$$

where the expectation is taken with respect to the randomness of the learner $\mathcal{A}$. Given this definition of regret, we define agnostic learnability of an evolution class.

Definition 1 (Agnostic Learnability under Markovian Regret) *An evolution class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ is learnable in the agnostic setting if and only if* $\inf_{\mathcal{A}} \mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) = o(T)$[2].

Perhaps a more natural definition of expected regret in the agnostic setting is to compare the prediction of the learner to the prediction of the best-fixed *trajectory* generated by functions in our evolution class. To that end, define

$$\mathrm{FR}_{\mathcal{A}}(T, \mathcal{F}) := \sup_{(x_0, x_1, ..., x_T)} \Big(\mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq x_t\}\right] - \inf_{f\in\mathcal{F}} \sum_{t=1}^{T} \mathbb{1}\{f^t(x_0) \neq x_t\}\Big). \tag{2}$$

as the expected *Flow regret*. An analogous definition of agnostic learnability follows.

Definition 2 (Agnostic Learnability under Flow Regret) *An evolution class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ is learnable in the agnostic setting under Flow regret if and only if* $\inf_{\mathcal{A}} \mathrm{FR}_{\mathcal{A}}(T, \mathcal{F}) = o(T)$.

2. $o(T)$ refers to any sublinear function of T.

A sequence of states $x_0, x_1, \ldots, x_T$ is said to be *realizable* by $\mathcal{F}$ if there exists an evolution function $f \in \mathcal{F}$ such that $f(x_{t-1}) = x_t$ for all $t \in [T]$. In the realizable setting, the cumulative loss of the best-fixed function is 0, and the goal of the learner is to minimize its expected cumulative mistakes

$$\mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) := \sup_{x_0} \sup_{f \in \mathcal{F}} \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{1}\{\mathcal{A}(x_{<t}) \neq f(x_{t-1})\}\right].$$

Analogously, we define the realizable learnability of $\mathcal{F}$.

Definition 3 (Realizable Learnability) *An evolution class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ is learnable in the realizable setting if and only if* $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) = o(T)$.

2.3. Complexity Measures

In sequential learning tasks, complexity measures are often defined in terms of *trees*, a basic unit that captures temporal dependence. In this paper, we use complete binary trees to define a new combinatorial object called a trajectory tree. In the remainder of this section and Section 2.4, we use trajectory trees to define complexity measures and combinatorial dimensions for evolution classes.

Definition 4 (Trajectory tree) *A trajectory tree of depth d is a complete binary tree of depth d where internal nodes are labeled by states in $\mathcal{X}$.*

Given a trajectory tree $\mathcal{T}$ of depth d, a root-to-leaf path down $\mathcal{T}$ is defined by a string $\sigma \in \{-1, 1\}^d$ indicating whether to go left ($\sigma_t = -1$) or to go right ($\sigma_i = +1$) at each depth $t \in [d]$. A path $\sigma \in \{-1, 1\}^d$ down $\mathcal{T}$ gives a trajectory $\{x_t\}_{t=0}^d$, where x_0 denotes the instance labeling the root node and x_t is the instance labeling the edge following the prefix $(\sigma_1, ..., \sigma_t)$ down the tree. A path $\sigma \in \{-1, 1\}^d$ down $\mathcal{T}$ is *shattered* by $\mathcal{F}$ if there exists a $f \in \mathcal{F}$ such that $f(x_{t-1}) = x_t$ for all $t \in [d]$, where $\{x_t\}_{t=0}^d$ is the corresponding trajectory obtained by traversing $\mathcal{T}$ according to σ. If every path down $\mathcal{T}$ is shattered by $\mathcal{F}$, we say that $\mathcal{T}$ is shattered by $\mathcal{F}$.

To make this more rigorous, we define a trajectory tree $\mathcal{T}$ of depth d as a sequence $(\mathcal{T}_0, \mathcal{T}_1, ..., \mathcal{T}_d)$ of node-labeling functions $\mathcal{T}_t : \{-1, 1\}^t \to \mathcal{X}$, which provide the labels for each internal node. Then, $\mathcal{T}_t(\sigma_1, ..., \sigma_t)$ gives the label of the node by following the prefix $(\sigma_1, ..., \sigma_t)$ and $\mathcal{T}_0$ denotes the instance labeling the root node. For brevity, we define $\sigma_{\leq t} = (\sigma_1, ..., \sigma_t)$ and write $\mathcal{T}_t(\sigma_1, ..., \sigma_t) = \mathcal{T}_t(\sigma_{\leq t})$. Analogously, we let $\sigma_{<t} = (\sigma_1, ..., \sigma_{t-1})$. Using this notation, a trajectory tree $\mathcal{T}$ of depth d is shattered by the evolution function class $\mathcal{F}$ if $\forall \sigma \in \{-1, 1\}^d$, there exists a $f_\sigma \in \mathcal{F}$ such that $f_\sigma(\mathcal{T}_{t-1}(\sigma_{<t})) = \mathcal{T}_t(\sigma_{\leq t})$ for all $t \in [d]$. Moreover, we use this notation to define the *Branching factor* of a trajectory tree.

Definition 5 (Branching factor) *The* Branching factor *of a trajectory tree $\mathcal{T}$ of depth d is*

$$\mathrm{B}(\mathcal{T}) := \min_{\sigma \in \{-1,1\}^d} \sum_{t=1}^{d} \mathbb{1}\{\mathcal{T}_t((\sigma_{<t}, -1)) \neq \mathcal{T}_t\big((\sigma_{<t}, +1)\big)\}.$$

The branching factor of a path $\sigma \in \{-1, 1\}^d$ captures the distinctness of states labeling the two children of internal nodes in this path. In particular, it counts the number of nodes in the path whose two children are labeled by distinct states. The branching factor of a trajectory tree is just the smallest branching factor across all paths. Using the notion of shattering and Definition 5, we define a new complexity measure, termed the Evolution complexity, of a function class $\mathcal{F}$.

Definition 6 (Evolution complexity) *Let $\mathcal{S}(\mathcal{F}, d)$ be the set of all trajectory trees of depth $d \in \mathbb{N}$ shattered by $\mathcal{F}$. Then, the* Evolution complexity *of $\mathcal{F}$ at depth d is defined as* $\mathrm{C}_d(\mathcal{F}) := \sup_{\mathcal{T} \in \mathcal{S}(\mathcal{F},d)} \mathrm{B}(\mathcal{T})$.

In Section 3, we show that the Evolution complexity exactly (up to a factor of 2) captures the minimax expected mistakes in the realizable setting. We provide some examples of classes $\mathcal{F}$ and their evolution complexities in Theorem 14. We note that there is an existing notion of complexity for dynamical systems, termed topological entropy, that quantifies the complexity of a particular evolution function $f \in \mathcal{F}$ (Adler et al., 1965). However, topological entropy does not characterize learnability as $\mathcal{F} = \{f\}$ is trivially learnable when f has infinite topological entropy.

2.4. Combinatorial dimensions

In addition to complexity measures, combinatorial dimensions play an important role in providing crisp quantitative characterizations of learnability. For example, the Daniely Shalev-Shwartz dimension (DSdim), originally proposed by Daniely and Shalev-Shwartz (2014) and formally defined below, was recently shown by Brukhim et al. (2022) to provide a tight quantitative characterization of multiclass PAC learnability. In Section 3.2, we use the DSdim to relate the realizable learnability of dynamical systems to multiclass PAC learnability of $\mathcal{F}$.

Definition 7 (DS dimension (Daniely and Shalev-Shwartz, 2014)) *We say that $A \subseteq \mathcal{X}$ is DS-shattered by $\mathcal{F}$ if there exists a finite $\mathcal{H} \subset \mathcal{F}$ such that for every $x \in A$ and $h \in \mathcal{H}$, there exists a $g \in \mathcal{H}$ such that $g(x) \neq h(x)$ and $g(z) = h(z)$ for all $z \in A \setminus \{x\}$. The DS dimension of $\mathcal{F}$, denoted $\mathrm{DS}(\mathcal{F})$, is the largest $d \in \mathbb{N}$ such that there exists a shattered set $A \subset \mathcal{X}$ with cardinality d. If there are arbitrarily large sets $A \subseteq \mathcal{X}$ that are shattered by $\mathcal{F}$, then we say that $\mathrm{DS}(\mathcal{F}) = \infty$.*

Analogously, for online multiclass classification, the Littlestone dimension (Ldim), originally proposed by Littlestone (1987) for binary classification and later extended to multiclass classification by Daniely et al. (2011), provides a tight quantitative characterization of learnability (Hanneke et al., 2023).

Definition 8 (Littlestone dimension (Littlestone, 1987; Daniely et al., 2011)) *Let $\mathcal{T}$ be a complete binary tree of depth d whose internal nodes are labeled by a sequence $(\mathcal{T}_0, \ldots, \mathcal{T}_{d-1})$ of node-labeling functions $\mathcal{T}_{t-1} : \{-1, 1\}^{t-1} \to \mathcal{X}$. The tree $\mathcal{T}$ is shattered by $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ if there exists a sequence $(Y_1, ..., Y_d)$ of edge-labeling functions $Y_t : \{-1, 1\}^t \to \mathcal{X}$ such that for every path $\sigma = (\sigma_1, ..., \sigma_d) \in \{-1, 1\}^d$, there exists a function $f_\sigma \in \mathcal{F}$ such that for all $t \in [d]$, $f_\sigma(\mathcal{T}_{t-1}(\sigma_{<t})) = Y_t(\sigma_{\leq t})$ and $Y_t((\sigma_{<t}, -1)) \neq Y_t((\sigma_{<t}, +1))$. The Littlestone dimension of $\mathcal{F}$, denoted $\mathrm{L}(\mathcal{F})$, is the maximal depth of a tree $\mathcal{T}$ that is shattered by $\mathcal{F}$. If there exists shattered trees of arbitrarily large depth, we say $\mathrm{L}(\mathcal{F}) = \infty$.*

3. Warmup: Realizable Learnability

In this section, we provide qualitative and quantitative characterizations of realizable learnability in terms of the Evolution complexity. Our main result in this section is Theorem 9, which provides bounds on the minimax expected number of mistakes.

Theorem 9 (Minimax Expected Mistakes) *For any $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$, we have $\frac{1}{2}\,\mathrm{C}_T(\mathcal{F}) \leq \inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \leq \mathrm{C}_T(\mathcal{F})$. Moreover, the upper bound is achieved constructively by a deterministic learner.*

The factor of $\frac{1}{2}$ in the lower bound is due to randomized learners, and the lower bound of $\mathrm{C}_T(\mathcal{F})$ can be obtained if the learner is restricted only to deterministic learning rules.

We now describe our high-level proof strategy and defer the full proof of Theorem 9 to Appendix A. Our lower bound involves picking the worst-case shattered tree with the largest branching factor and traversing down this tree uniformly at random to generate the sequence of states. For such a sequence, the learner can do no better than random guessing at nodes where the branching occurred, yielding the lower bound $\mathrm{C}_T(\mathcal{F})/2$. Next, for our upper bound, we first define a localized complexity measure $\mathrm{C}_T(\mathcal{F}, x_0)$, where we only consider shattered trees rooted at the revealed initial state x_0. Our minimax learner is then a version space algorithm that predicts the state that will result in the largest reduction in the complexity measure if a mistake occurs. This learner is a generalization of the celebrated Standard Optimal Algorithm due to Littlestone (1987).

3.1. Minimax Rates in the Realizable Setting

While Theorem 9 provides a quantitative and qualitative characterization of realizable learnability, it does not shed light on how $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(\mathcal{F}, T)$ may depend on the time horizon T. In online classification with the 0-1 loss, the seminal work by Littlestone (1987) and Daniely et al. (2011) show that only two rates are possible: $\Theta(T)$ and $\Theta(1)$. That is, if a hypothesis class is online learnable in the realizable setting, then it is learnable with a constant mistake bound (i.e. the Littlestone dimension). Perhaps surprisingly, this is not the case for learning dynamical systems in a strong sense: *every* rate is possible.

Theorem 10 *For every $S \subset \mathbb{N} \cup \{0\}$, there exists $\mathcal{F}_S \subseteq \mathbb{Z}^{\mathbb{Z}}$ such that $\mathrm{C}_T(\mathcal{F}_S) = \sup_{n \in \mathbb{N} \cup \{0\}} |S \cap \{n, n+1, ..., n+T-1\}|$.*

Theorem 10, proved in Appendix B, along with Theorem 9 implies that any minimax rate in the realizable setting is possible. As an example, suppose we would like to achieve the rate $\Theta(T^\alpha)$ for some $\alpha < 1$. Then, picking $S = \{\lfloor t^{\frac{1}{\alpha}} \rfloor : t \in \mathbb{N} \cup \{0\}\}$ suffices since $\sup_{x \in \mathbb{N} \cup \{0\}} |S \cap \{n, n+1, ..., n+T-1\}| = |\{\lfloor t^{\frac{1}{\alpha}} \rfloor : t \in \mathbb{N} \cup \{0\}\} \cap \{0, 1, ..., T-1\}| = \Theta(T^\alpha)$. Likewise, one can get logarithmic rates by picking $S = \{2^t : t \in \mathbb{N} \cup \{0\}\}$ and constant rates by picking $S \subset \mathbb{N} \cup \{0\}$ such that $|S| < \infty$. In light of Theorem 10 and the fact that only constant mistake bounds are possible for online multiclass classification, it is natural to ask *when* one can achieve constant mistake bounds for learning dynamical systems. To answer this question, we introduce a new combinatorial dimension termed the Branching dimension.

Definition 11 (Branching dimension) *The* Branching dimension*, denoted $\mathrm{Bd}(\mathcal{F})$, is the smallest natural number $d \in \mathbb{N}$ such that for every shattered trajectory tree $\mathcal{T}$, we have $\mathrm{B}(\mathcal{T}) \leq d$. If for every $d \in \mathbb{N}$, there exists a shattered trajectory tree $\mathcal{T}$ with $\mathrm{B}(\mathcal{T}) > d$, we say $\mathrm{Bd}(\mathcal{F}) = \infty$.*

Theorem 12, proved via non-constructive arguments in Appendix C, shows that $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) = \Theta(1)$ if and and only if $\mathrm{Bd}(\mathcal{F}) < \infty$.

Theorem 12 (Constant Minimax Expected Mistakes) *For any $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$, we have (i) $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \leq \mathrm{Bd}(\mathcal{F})$ and (ii) if $\mathrm{Bd}(\mathcal{F}) = \infty$, then $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) = \omega(1)$*[3].

3. Recall that $f(n) = \omega(1)$ if for all $k > 0$, there exists n_k, such that for all $n > n_k$ we have $f(n) > k$.

3.2. Relations to PAC and Online Multiclass Classification

By studying abstract state spaces $\mathcal{X}$ and evolutions function classes $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$, we can also compare realizable learnability of dynamical systems to existing notions of realizable learnability in the well-known PAC and online classification settings. Our main result relates the evolution complexity to the DS and Littlestone dimensions, which characterize PAC and online classification respecitively.

Theorem 13 (Relations to the DS and Littlestone dimension) *The following statements are true.*

(i) There exists $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ *such that* $\mathrm{DS}(\mathcal{F}) = \infty$ *but* $\mathrm{C}_T(\mathcal{F}) = \Theta(\log(T))$.

(ii) There exists $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ *such that* $\mathrm{DS}(\mathcal{F}) = 1$ *but* $\mathrm{C}_T(\mathcal{F}) = T$.

(iii) For any $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$*, we have that* $\mathrm{C}_T(\mathcal{F}) \leq \mathrm{L}(\mathcal{F})$.

(iv) There exists $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ *such that* $\mathrm{L}(\mathcal{F}) = \infty$ *but* $\mathrm{C}_T(\mathcal{F}) = 1$.

The proof of Theorem 13 is in Appendix D. Parts (i) and (ii) show that the finiteness of the DSdim is neither necessary nor sufficient for learning dynamical systems in the realizable setting. On the other hand, parts (iii) and (iv) show that finite Ldim is sufficient but not necessary for learning dynamical systems in the realizable setting. Overall, learning dynamical systems is always easier than online multiclass classification, but can be both easier and harder than multiclass PAC classification. The proof of (i), (ii), and (iv) are combinatorial in nature, while the proof of (iii) involves reducing learning-to-predict in dynamical systems to online multiclass classification.

3.3. Examples

In this section, we establish the minimax rates for discrete linear systems and linear Boolean networks. The proof of Theorem 14 is in Appendix E.

Theorem 14 (Linear Systems)

(i) Let $\mathcal{X} = \mathbb{Z}^n$ *and* $r < n$*. For* $\mathcal{F} = \{x \mapsto Wx : W \in \mathbb{Z}^{n\times n}, \operatorname{rank}(W) \leq r\}$ *and* $T > r$*, we have* $\mathrm{C}_T(\mathcal{F}) = r + 1$.

(ii) Let $\mathcal{X} = \{0,1\}^n$ *and* $T \geq n$*. For* $\mathcal{F} = \{x \mapsto Wx \,(\mathrm{mod}\, 2) : W \in \mathbb{Z}^{n\times n}\}$*, we have* $\mathrm{C}_T(\mathcal{F}) = n$.

(iii) Let $\mathcal{X} = \{0,1\}^n$ *and* $T \geq n$*. For* $\mathcal{F} = \{x \mapsto \mathbb{1}\{Wx > 0\} : W \in \{0,1\}^{n\times n}\}$*, we have* $n \leq \mathrm{C}_T(\mathcal{F}) \leq n^2$.

Thresholded Boolean networks have been used to model genetic regulatory dynamics (Mendoza and Alvarez-Buylla, 1998) and social networks (Kempe et al., 2003). Modulo Boolean networks have been studied by Chandrasekhar et al. (2023) in the context of stability.

4. Agnostic Learnability

4.1. Markovian Regret

In this section, we go beyond the realizable setting and consider the case where nature may reveal a trajectory that is not consistent with any evolution function in the class. Our main result in this section establishes bounds on the minimax expected Markovian regret.

Theorem 15 (Minimax Expected Markovian Regret) *For any $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$,*

$$\max\left\{\frac{\mathrm{L}(\mathcal{F})}{18}, \frac{\sqrt{T}}{16\sqrt{3}}\right\} \leq \inf_{\mathcal{A}} \mathrm{MR}_{\mathcal{A}}(T, \mathcal{F}) \leq \mathrm{L}(\mathcal{F}) + \sqrt{T\,\mathrm{L}(\mathcal{F})\log T}.$$

Theorem 15 shows that the finiteness of the Littlestone dimension of $\mathcal{F}$ is both necessary and sufficient for agnostic learnability. This is in contrast to Theorem 13, which shows that the finiteness of the Littlestone dimension of $\mathcal{F}$ is sufficient but *not necessary* for realizable learnability. Thus, Theorem 15 and 13 imply that realizable and agnostic learnability are not equivalent.

Corollary 16 (Realizable Learnability $\not\equiv$ Agnostic Learnability under Markovian Regret) *There exists a class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ such that $\mathcal{F}$ is learnable in the realizable setting but not in the agnostic setting under Markovian regret.*

One such class exhibiting the separation is the thresholds $\mathcal{F} = \{x \mapsto \mathbb{1}\{x \geq a\} : a \in (0,1)\}$ used in the proof of part (iv) in Theorem 13. Beyond the qualitative separation of realizable and agnostic learnability under Markovian regret, we also observe a quantitative separation in terms of possible minimax rates. Recall that Theorem 10 shows that every rate is possible in the realizable setting. However, Theorem 15 shows that only two types of rates are possible in the agnostic setting: $\Theta(T)$ whenever $\mathrm{L}(\mathcal{F}) = \infty$ and $\tilde{\Theta}(\sqrt{T})$ whenever $\mathrm{L}(\mathcal{F}) < \infty$. More precisely, when $\sqrt{T} \geq \mathrm{L}(\mathcal{F})$, we have a lower bound of $\Omega(\sqrt{T})$ and an upper bound of $O(\sqrt{T\,\mathrm{L}(\mathcal{F})\log T})$. This raises the natural question of what the right minimax rate is. In Appendix G, we provide an evolution class and establish a lower bound of $\Omega(\sqrt{T\,\mathrm{L}(\mathcal{F})})$, showing that the upper bound is tight up to $\sqrt{\log T}$.

Our proof of the upper bound in Theorem 15 reduces learning dynamical systems to online multiclass classification and uses a result due to Hanneke et al. (2023). To prove the lower bound $\frac{\mathrm{L}(\mathcal{F})}{18}$, we construct a hard stream by carefully sampling a random path down a Littlestone tree of depth $\mathrm{L}(\mathcal{F})$. To prove the lower bound of $\frac{\sqrt{T}}{16\sqrt{3}}$, we construct a hard randomized stream using just two different evolution functions in $\mathcal{F}$. The full proof is in Appendix F.

4.2. Flow Regret

The necessity of the Littlestone dimension for agnostic learnability under Markovian regret is quite restrictive. For example, a simple (but unnatural) class like one-dimensional thresholds $\mathcal{F} = \{x \mapsto \mathbb{1}\{x \leq a\} : a \in (0,1)\}$ has $\mathrm{L}(\mathcal{F}) = \infty$ but $\mathrm{C}_T(\mathcal{F}) = 1$. The key idea in the lower bound of Theorem 17 is that the adversary can simulate the online multiclass classification game, where finiteness of the Littlestone dimension is necessary, by "giving up" every other round. This is possible because of the definition of Markovian regret. In particular, the evaluation of the "best-fixed evolution function in hindsight" under Markovian regret is only penalized on one-step prediction error but not on long-term consistency of the generated dynamics starting from the initial state x_0.

This motivates the following natural question. Which evolution classes $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ are agnostic learnable under Flow regret? Is the finiteness of Ldim still necessary? Theorem 17, whose proof can be found in Appendix H, provides partial answers to these question by bounding the minimax expected Flow regret for classes where $\sup_{x\in\mathcal{X}} |\mathcal{F}(x)|$ is uniformly bounded.

Theorem 17 (Minimax Expected Flow Regret) *For any ordered set $\mathcal{X}$ and $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$,*

$$\frac{\mathrm{C}_T(\mathcal{F})}{2} \leq \inf_{\mathcal{A}} \mathrm{FR}_{\mathcal{A}}(T, \mathcal{F}) \leq \mathrm{C}_T(\mathcal{F}) + \sqrt{\mathrm{C}_T(\mathcal{F})\, T \ln\Big(\frac{T\, K_{\mathcal{F}}}{\mathrm{C}_T(\mathcal{F})}\Big)}.$$

where $K_{\mathcal{F}} = \sup_{x\in\mathcal{X}} |\mathcal{F}(x)|$. Moreover, both the lower- and upper bound can be tight.

Note that the upper bound becomes vacuous when $\sup_{x\in\mathcal{X}} |\mathcal{F}(x)| = \infty$, and finding a general characterization of Flow regret learnability remains an open question. Unlike agnostic learnability under Markovian regret, the finiteness of the Ldim is not necessary for agnostic learnability under Flow regret. Indeed, while the class of one-dimensional thresholds is not agnostic learnable under Markovian regret, it is agnostic learnable under Flow regret. Although Markovian learnability is not necessary for Flow regret learnability, when $K_{\mathcal{F}} < \infty$, learnability under Markovian regret is sufficient learnability under flow regret. To see this, recall that Theorem 15 states that learnability under Markovian regret implies $\mathrm{L}(\mathcal{F}) < \infty$. Then, since part (iii) of Theorem 13 states $\mathrm{C}_T(\mathcal{F}) \leq \mathrm{L}(\mathcal{F})$, we can use Theorem 17 to infer that $\mathcal{F}$ is also learnable under flow regret with regret $\leq \mathrm{L}(\mathcal{F}) + \sqrt{\mathrm{L}(\mathcal{F}) T \ln(T K_{\mathcal{F}})}$.

Theorem 17 shows that realizable and agnostic learnability under Flow regret are equivalent as long as $\sup_{x\in\mathcal{X}} |\mathcal{F}(x)| < \infty$. But this equivalence breaks down when the projection sizes are unbounded, as shown by Theorem 18.

Theorem 18 (Realizable learnability $\not\equiv$ Agnostic Learnability under Flow Regret) *There exists an ordered set $\mathcal{X}$ and $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ such that (i) $\inf_{\mathcal{A}} \mathrm{M}_{\mathcal{A}}(T, \mathcal{F}) \leq 3$ but (ii) $\inf_{\mathcal{A}} \mathrm{FR}_{\mathcal{A}}(T, \mathcal{F}) \geq \frac{T}{6}$.*

To prove Theorem 18 (see Appendix I), we construct a class $\mathcal{F} \subseteq \mathcal{X}^{\mathcal{X}}$ such that on large subset of states in $\mathcal{X}' \subset \mathcal{X}$, every function $f \in \mathcal{F}$ effectively reveals its identity.

5. Discussion and Future directions

In this work, we studied the problem of learning-to-predict in discrete-time dynamical systems under the 0-1 loss. A natural extension is to consider continuous state spaces with real-valued losses. For example, one can take $\mathcal{X}$ to be a bounded subset of a Hilbert space and consider the squared norm as the loss function. Another natural extension is to consider learnability under partial observability, where the learner only observes some transformation $\phi(x_t)$ instead of the true state x_t. Such feedback model is standard in prediction for linear dynamical systems (Hazan et al., 2018). It is also natural to study the learnability of function classes where the output of the evolution rules $f : \mathcal{X}^p \to \mathcal{X}$, depend on the previous $p > 1$ states (e.g. the p-th order VAR model). Lastly, the learning algorithms in this work are *improper*: they use evolution functions that may not lie in $\mathcal{F}$ to make predictions. This might be undesirable as improper learning algorithms may be incompatible with downstream system identification and control tasks. To this end, characterizing *proper* learnability of dynamical systems is an important future direction.

Acknowledgments

AT and US acknowledge the support of NSF via grant DMS-2413089. US also acknowledges the support of Rackham International Student Fellowship. VR acknowledges the support of the NSF GRFP. We thank Chinmaya Kaushik for pointing out a technical result, which helped us to prove part (i) Theorem 14.

References

Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 1–26. JMLR Workshop and Conference Proceedings, 2011.

Roy L Adler, Alan G Konheim, and M Harry McAndrew. Topological entropy. *Transactions of the American Mathematical Society*, 114(2):309–319, 1965.

Naman Agarwal, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh. Online control with adversarial disturbances. In *International Conference on Machine Learning*, pages 111–119. PMLR, 2019.

Oren Anava, Elad Hazan, Shie Mannor, and Ohad Shamir. Online learning for time series prediction. In *Conference on learning theory*, pages 172–184. PMLR, 2013.

Oren Anava, Elad Hazan, and Assaf Zeevi. Online time series prediction with missing data. In *International conference on machine learning*, pages 2191–2199. PMLR, 2015.

Karl Johan Åström and Peter Eykhoff. System identification—a survey. *Automatica*, 7(2):123–162, 1971.

Sohail Bahmani and Justin Romberg. Convex programming for estimation in nonlinear recurrent models. *The Journal of Machine Learning Research*, 21(1):9563–9582, 2020.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

David Belanger and Sham Kakade. A linear dynamical system model for text. In *International Conference on Machine Learning*, pages 833–842. PMLR, 2015.

Tyrus Berry and Suddhasattwa Das. Learning theory for dynamical systems. *SIAM Journal on Applied Dynamical Systems*, 22(3):2082–2122, 2023.

Tyrus Berry and Suddhasattwa Das. Limits of learning dynamical systems. *SIAM Review*, 67(1): 107–137, 2025.

Nataly Brukhim, Daniel Carmon, Irit Dinur, Shay Moran, and Amir Yehudayoff. A characterization of multiclass learnability, 2022. URL https://arxiv.org/abs/2203.01550.

Marco C Campi and Erik Weyer. Finite sample properties of system identification methods. *IEEE Transactions on Automatic Control*, 47(8):1329–1334, 2002.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Karthik Chandrasekhar, Claus Kadelka, Reinhard Laubenbacher, and David Murrugarra. Stability of linear boolean networks. *Physica D: Nonlinear Phenomena*, 451:133775, 2023.

Jan Chorowski and Navdeep Jaitly. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695*, 2016.

Alon Cohen, Avinatan Hasidim, Tomer Koren, Nevena Lazic, Yishay Mansour, and Kunal Talwar. Online linear quadratic control. In *International Conference on Machine Learning*, pages 1029–1038. PMLR, 2018.

Amit Daniely and Shai Shalev-Shwartz. Optimal learners for multiclass problems. In *Conference on Learning Theory*, pages 287–316. PMLR, 2014.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. In Sham M. Kakade and Ulrike von Luxburg, editors, *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pages 207–232, Budapest, Hungary, 09–11 Jun 2011. PMLR.

Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic textures. *International journal of computer vision*, 51:91–109, 2003.

Jeffrey L Elman. Language as a dynamical system. *Mind as motion: Explorations in the dynamics of cognition*, pages 195–223, 1995.

Dylan Foster, Tuhin Sarkar, and Alexander Rakhlin. Learning nonlinear dynamical systems from a single trajectory. In *Learning for Dynamics and Control*, pages 851–861. PMLR, 2020.

Amin Ghadami and Bogdan I Epureanu. Data-driven prediction in dynamical systems: recent developments. *Philosophical Transactions of the Royal Society A*, 380(2229):20210213, 2022.

Udaya Ghai, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. No-regret prediction in marginally stable systems. In *Conference on Learning Theory*, pages 1714–1757. PMLR, 2020.

Dimitrios Giannakis, Amelia Henriksen, Joel A Tropp, and Rachel Ward. Learning to forecast dynamical systems from streaming data. *SIAM Journal on Applied Dynamical Systems*, 22(2): 527–558, 2023.

Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Learning graph cellular automata. *Advances in Neural Information Processing Systems*, 34:20983–20994, 2021.

Steve Hanneke, Shay Moran, Vinod Raman, Unique Subedi, and Ambuj Tewari. Multiclass online learning and uniform convergence. *Proceedings of the 36th Annual Conference on Learning Theory (COLT)*, 2023.

Elad Hazan, Karan Singh, and Cyril Zhang. Learning linear dynamical systems via spectral filtering. *Advances in Neural Information Processing Systems*, 30, 2017.

Elad Hazan, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. Spectral filtering for general linear dynamical systems. *Advances in Neural Information Processing Systems*, 31, 2018.

Alfons G Hoekstra, Jiri Kroc, and Peter MA Sloot. *Simulating complex systems by cellular automata*. Springer, 2010.

Prateek Jain, Suhas S Kowshik, Dheeraj Nagaraj, and Praneeth Netrapalli. Streaming linear system identification with reverse experience replay. *arXiv preprint arXiv:2103.05896*, 2021.

Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. *Advances in Neural Information Processing Systems*, 33:15312–15325, 2020.

Stuart Kauffman. Homeostasis and differentiation in random genetic control networks. *Nature*, 224 (5215):177–178, 1969.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

Suhas Kowshik, Dheeraj Nagaraj, Prateek Jain, and Praneeth Netrapalli. Near-optimal offline and streaming algorithms for learning non-linear dynamical systems. *Advances in Neural Information Processing Systems*, 34:8518–8531, 2021a.

Suhas Kowshik, Dheeraj Nagaraj, Prateek Jain, and Praneeth Netrapalli. Streaming linear system identification with reverse experience replay. *Advances in Neural Information Processing Systems*, 34:30140–30152, 2021b.

Mark Kozdoba, Jakub Marecek, Tigran Tchrakian, and Shie Mannor. On-line learning of linear dynamical systems: Exponential forgetting in kalman filters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4098–4105, 2019.

Sahin Lale, Kamyar Azizzadenesheli, Babak Hassibi, and Anima Anandkumar. Logarithmic regret bound in partially observable linear dynamical systems. *Advances in Neural Information Processing Systems*, 33:20876–20888, 2020.

Holden Lee. Improved rates for prediction and identification of partially observed linear dynamical systems. In *International Conference on Algorithmic Learning Theory*, pages 668–698. PMLR, 2022.

Yingying Li, Xin Chen, and Na Li. Online optimal control with linear dynamics and predictions: Algorithms and regret analysis. *Advances in Neural Information Processing Systems*, 32, 2019.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1987.

Chenghao Liu, Steven CH Hoi, Peilin Zhao, and Jianling Sun. Online arima algorithms for time series prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Lennart Ljung. System identification: theory for the user. *PTR Prentice Hall, Upper Saddle River, NJ*, 28:540, 1999.

Luis Mendoza and Elena R Alvarez-Buylla. Dynamics of the genetic regulatory network forarabidopsis thalianaflower morphogenesis. *Journal of theoretical biology*, 193(2):307–319, 1998.

Deepan Muthirayan and Pramod P Khargonekar. Online learning robust control of nonlinear dynamical systems. *arXiv preprint arXiv:2106.04092*, 2021.

Zirou Qiu, Abhijin Adiga, Madhav V Marathe, SS Ravi, Daniel J Rosenkrantz, Richard E Stearns, and Anil Vullikanti. Learning the topology and behavior of discrete dynamical systems.

Paria Rashidinejad, Jiantao Jiao, and Stuart Russell. Slip: Learning to predict in unknown dynamical systems with long-term memory. *Advances in Neural Information Processing Systems*, 33:5716–5728, 2020.

Daniel J Rosenkrantz, Abhijin Adiga, Madhav Marathe, Zirou Qiu, SS Ravi, Richard Stearns, and Anil Vullikanti. Efficiently learning the topology and behavior of a networked dynamical system via active queries. In *International Conference on Machine Learning*, pages 18796–18808. PMLR, 2022.

Yahya Sattar and Samet Oymak. Non-asymptotic and accurate learning of nonlinear dynamical systems. *The Journal of Machine Learning Research*, 23(1):6248–6296, 2022.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014.

Ilya Shmulevich, Edward R Dougherty, and Wei Zhang. From boolean to probabilistic boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, 90(11):1778–1792, 2002.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

Anastasios Tsiamis and George J Pappas. Online learning of the kalman filter with logarithmic regret. *IEEE Transactions on Automatic Control*, 68(5):2774–2789, 2022.

Vladimir Naumovich Vapnik and Aleksei Yakovlevich Chervonenkis. On uniform convergence of the frequencies of events to their probabilities. *Teoriya Veroyatnostei i ee Primeneniya*, 16(2): 264–279, 1971.

Mathukumalli Vidyasagar and Rajeeva L Karandikar. A learning theory approach to system identification and stochastic adaptive control. *Probabilistic and randomized methods for design under uncertainty*, pages 265–302, 2006.

Wen-Xu Wang, Ying-Cheng Lai, and Celso Grebogi. Data-based identification and prediction of nonlinear and complex dynamical systems. *Physics Reports*, 644:1–76, 2016.

Stephen Wolfram. Theory and applications of cellular automata. *World Scientific*, 1986.

N Wulff and J A Hertz. Learning cellular automaton dynamics with neural networks. *Advances in Neural Information Processing Systems*, 5, 1992.

Haimin Yang, Zhisong Pan, Qing Tao, and Junyang Qiu. Online learning for vector autoregressive moving-average time series prediction. *Neurocomputing*, 315:9–17, 2018.