Best of many worlds: Robust model selection for online supervised learning

Vidya Muthukumar[*] **Mitas Ray**[*,†] **Anant Sahai**[*] **Peter L Bartlett**[*,+]
[*]UC Berkeley EECS [+]UC Berkeley Statistics [†]University of Washington EE

Abstract

We introduce algorithms for online, full-information prediction that are computationally efficient and competitive with contextual tree experts of unknown complexity, in both probabilistic and adversarial settings. We incorporate a novel probabilistic framework of structural risk minimization into existing adaptive algorithms and show that we can robustly learn not only the presence of stochastic structure when it exists, but also the correct model order. When the stochastic data is actually realized from a predictor in the model class considered, we obtain regret bounds that are competitive with the regret of an optimal algorithm that possesses strong side information about both the true model order and whether the process generating the data is stochastic or adversarial. In cases where the data does not arise from any of the models, our algorithm selects models of higher order as we play more rounds. We display empirically improved *overall prediction error* over other adversarially robust approaches.

1 Introduction

In full-information online learning, there are no generative assumptions on the data. We consider *online supervised learning* where we observe pairs of covariates and responses, and need to minimize regret with respect to the best function in hindsight from a *fixed model class*. In the case where covariates and responses are discrete, we can consider the $0-1$ loss function, and characterize the performance of *tree experts* (also called *contextual experts*) that map a covariate to an appropriate response. A natural goal is to minimize *minimax cumulative regret* as a function of the number of rounds T. This is well known to scale [CBFH$^+$97] as $\mathcal{O}(\sqrt{T}\cdot(\text{max. model complexity}))$. Once this is guaranteed, we are especially interested in adaptive algorithms that preserve this guarantee and also adapt to "easier" stochastic structure. Again, it is well known that we can get much faster $\mathcal{O}((\text{max. model complexity}))$ rates in this case; essentially, constant regret. Recent work [CBMS07, EKRG11, DRVEGK14, LS15, KVE15, KGvE16] constructs algorithms that adapt to these faster rates while preserving the minimax rate; thus obtaining the *best of both worlds*.

A more classical goal of adaptivity is *adapting to the complexity of the true model class*. *Offline model selection* has a rich history [Boz87, Vap99, Mas07] - typically a structured hierarchy of models is studied, and the right model for the problem can be chosen in a data-adaptive fashion when the data is independent and identically distributed. It is clear that model adaptivity is a natural goal in online learning – after all, while low regret is important, so is the right choice of benchmark with respect to which to minimize regret. And the importance of model selection is reflected very naturally in regret: either our data is not well-expressed by the used model class, leading us to question what a good regret rate really means, or our data is actually realized from a simple model and we spend more time than needed looking for the right predictor, building up unnecessary regret. Even if the data is actually generated by a very complex model, it may be well-approximated by simpler models – in which case we might still prefer predicting according to the simpler, easier-to-learn model. In these very general settings, while we often frame the objective as regret minimization, it is important not to forget that our actual goal is minimizing overall expected loss/prediction error. And so the choice of benchmark is as important as guaranteeing achievement of it.

In this context, we have a natural goal. Starting with

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

absolutely no assumptions, we still wish to protect ourselves from adversaries with the minimax regret rates (up to constants). However, we also want to adapt simultaneously to the existence *and* statistical complexity of stochastic structure, and perform almost as well as an algorithm with oracle knowledge of that structure would.

Typically, we use adaptive entropy regularization with a changing learning rate to interpolate between the stochastic and adversarial regimes. Structural risk minimization has been considered in purely stochastic, or purely adversarial environments, and uses a very different kind of model complexity regularization. In this paper, we develop a probabilistic notion of structural risk minimization and adaptively recover the stochastic model selection framework in a contextual experts setting under mild generative assumptions. When the generative model is actually within the model class, we obtain near-optimal, theoretical guarantees on regret in expectation and with high probability. Under *model-misspecified settings* that are nevertheless approximable by finite-order models, we empirically show that our algorithm adapts to higher-order models after playing more rounds – and demonstrate the added advantage of achieving this kind of two-fold adaptivity.

Our contributions We show that an adaptive variant of the computationally efficient *tree expert forecaster* adapts not only to stochastic structure but also the *order* of that stochastic structure that best describes the mapping between covariates and responses.

In the case when the sequence is actually generated from one of the tree expert models, we obtain a near-optimal regret guarantee, stated below. (For a formal statement of the theorem, see Theorem 1.)

Theorem 1 (informal): *Let D be the maximum model order of tree experts. The regret of our algorithm with respect to the best d^{th}-order tree expert is $O(\sqrt{T}2^d)$ in an adversarial setting and $O(d \cdot D \ln D \cdot 2^{2d})$* ***with high probability*** *when the data is actually generated by a d^{th}-order tree expert, for any $d \in \{0, \ldots, D\}$.*

Thus, we can recover stochastic online model selection in an adversarial framework – our regret rate for d^{th}-order processes is achieved without knowing the value of d in advance, or even that the process is stochastic. In stochastic environments, this rate is competitive with the optimal regret rate (which is $\mathcal{O}(2^d)$) that would be achieved by a greedy algorithm possessing side information about both the existence of stochastic structure and the true model order. In the adversarial regime, the rate is optimal in terms of its dependence on the time horizon T. To the best of our knowledge, this is the first provable guarantee on simultaneous adaptivity by an *efficient algorithm*[1].

Interestingly, we are able to obtain these guarantees for an algorithm that is a natural adaptation of the standard exponential weights framework, and our results have an intuitive interpretation. We combine the adaptivity to stochasticity of an existing "best-of-both-worlds" algorithm (called AdaHedge [EKRG11, DRVEGK14]) with the prior weighting on tree experts that is used in tree forecasters [HS97][2]. As is intuitive, the prior is inversely proportional to the complexity of the tree expert.

Our analysis recovers the stochastic structural risk minimization framework in a probabilistic sense. There are two penalties involved: the complexity of the model selected (to achieve model selection) as well as determinism (to ensure protection against adversaries). Remarkably, our algorithm uses a common time-varying, data-dependent learning rate, defined in the elegant AdaHedge style, to *learn* the correct proportion with which to apply both regularizers.

We also empirically consider more challenging stochastic settings in which the data does not, in fact, arise from any of the models considered. The quantity of regret does not necessarily make sense here, as there is no natural benchmark that contains a generative model for the sequence. We still do care about overall prediction error – ideally, our approach should select higher-order models as we play more rounds as well as form better estimates under these models. We display the empirical benefit of our model-order-selecting framework in simulations in Section 5 on predicting sequences realized by hidden Markov models, which have been shown to be approximable by finite-memory models [SKLV18].

Related work The framework for *offline* structural risk minimization in purely stochastic environments was laid out in seminal work (for a review, see [Mas07]). Generalization bounds are used to characterize model order complexity, and empirical process theory is used to show that data-adaptive model selection can be performed with high probability. Online bandit approaches for stochastic model selection have also been considered more recently [ADBL11].

[1] A clever application of the Bernstein condition on the recently proposed Squint [KGvE16] shows simultaneous adaptivity in the realizable case, but Squint as directly applied to the tree expert problem is computationally prohibitive.

[2] Most interestingly, this prior distribution was designed for the original *tree expert forecaster* [HS97], but this algorithm could not effectively utilize the prior because of the fixed learning rate.

On the other side, the paradigm for *adversarial* regret minimization was laid out in the discrete "experts" setting in seminal work (for a review, see [CBFH+97]), and subsequently lifted up to the more general *online convex optimization* framework (for a review, see [SS+12]). The next natural goal was adaptivity to several types of "easier" instances while preserving the worst-case guarantees. Most pertinent to our work are the easier *stochastic losses* [DRVEGK14], under which the greedy Follow-the-Leader algorithm achieves regret $\mathcal{O}(1)$. In the experts setting, multiple algorithms have been proposed [CBMS07, EKRG11, DRVEGK14, LS15, KVE15, KGvE16] that adaptively achieve $\mathcal{O}(1)$ regret. Some of these guarantees have been extended to online optimization [vEK16]. As we will see, naively extending these analyses to the tree expert forecaster problem gives a pessimistic $\mathcal{O}(2^D)$ regret bound. In our work, we show that we can get the best of *many worlds* and greatly improve the exponent to $\widetilde{\mathcal{O}}(2^{2d})$, reducing the dependence on the maximum model complexity D from exponential to linear.

Recent guarantees on adapting to a simpler model class, *but not to stochasticity*, have also been developed [RS13, Ora14, LS15, KVE15, OP16, FKMS17]. Many of these approaches [RS13, Ora14, OP16, FKMS17] do not improve the $\mathcal{O}(\sqrt{T})$ rate for stochastic data. We address in particular two recent algorithms, AdaNormalHedge [LS15] and Squint [KVE15], both of which obtain second-order quantile regret bounds in terms of a "variance" term and the correct model complexity *in the worst case*. AdaNormalHedge can be implemented two ways: inefficiently by considering all tree experts in each round, and efficiently using a sleeping experts reduction [**?**]. For both implementations, proving the *stochastic model selection guarantee* is non-trivial and does not follow from existing analysis[3]. Squint cleverly applies the Bernstein condition [KGvE16] and obtains the optimal stochastic rate of $\mathcal{O}(2^d)$ for the special "realizability" case. However, the computational complexity of Squint necessarily scales linearly with the number of experts, which in the case of the tree expert problem is a prohibitive *double-exponential-in-D* complexity. It is not obvious how to reduce Squint's computational complexity, as the algorithm uses a black-box framework on prediction using expert advice and results in a more complex update. We consider the broad exponential weights framework that is well-known to be efficiently implementable for the tree expert problem [HS97], and our analysis intuitively tackles the model selection problem directly.

The most complex *model-misspecified* example we consider is that of hidden Markov models, of which parameter estimation is an active area of research. Perhaps surprisingly, finite-memory models are reasonable approximators of a HMM [SKLV18], and can be reliably estimated under a mixing condition on the hidden Markov chain. Our empirical results support this recent theory.

2 Problem statement

We consider a contextual prediction setting over $T > 0$ rounds, in which we receive context-output pairs $(X_t, Y_t)_{t=1}^T$. We consider $X_t \in \mathcal{X}^D, Y_t \in \mathcal{X}$, where $\mathcal{X} = \{0, 1\}$ is the binary alphabet[4]. It will also be natural to consider the truncated version of X_t that only represents the last d coordinates – we denote this by $X_t(d)$, with the convention that $X_t := X_t(D)$. Note that this includes the *universal sequence prediction* paradigm in which the context $X_t(d) = \{Y_s\}_{s=t-d}^{t-1}$ comprises of previous observation values itself.

We follow the online supervised learning paradigm: before round t, we are given access to X_t, but not Y_t. Let $\mathcal{F}_D$ denote the set of all tree experts, expressed as Boolean functions from $\mathcal{X}^D$ to $\mathcal{X}$. We will also be considering tree experts that map from the subcontexts $\{X_t(h)\}$ to outputs Y_t, denoted by $\mathbf{f}_h \in \mathcal{F}_h$ for all values of h in $\{0, 1, \ldots, D\}$. (In universal prediction, these can be thought of as *finite-memory predictors*.) We use the shorthand notation $\mathbf{f} := \mathbf{f}_D \in \mathcal{F}_D$. We define the *order* of a tree expert, denoted by $\text{order}(\mathbf{f}_h)$, as the minimum value of $d \leq h$ for which its functionality can be expressed equivalently in terms of a function from $\mathcal{X}^d$ to $\mathcal{X}$. That is,

$$\text{order}(\mathbf{f}_h) := \min\{d \leq h : \text{ there exists } \mathbf{f}'_d \in \mathcal{F}_d \text{ s.t. } \mathbf{f}_h(x(h)) = \mathbf{f}'_d(x(d)) \text{ for all } x(h) \in \mathcal{X}^h\}.$$

We define our randomized online algorithm for *prediction using tree experts* in terms of a sequence of probability distributions $\{\mathbf{w}_t^{(\text{tree})}\}_{t=1}^T$ over the set $\mathcal{F}_D$ of all tree experts. Note that $\mathbf{w}_t^{(\text{tree})}$ cannot depend on $\{(X_s, Y_s)\}_{s \geq t+1}$ or Y_t. We denote the realization of the prediction at time t by $\widehat{Y}_t \in \mathcal{X}$, and the distribution on $\widehat{Y}_t$ by $\mathbf{w}_t$ (clearly induced by $\mathbf{w}_t^{(\text{tree})}$). After

[3] In more detail: Theorem 2 part 2 of the paper, which gives regret bounds in the stochastic regime, requires a single best expert in every round and a positive gap between that expert and all others. Noting that the expected gap for the tree expert problem is $1/2^D$, applying this directly to the inefficient version gives a pessimistic $\mathcal{O}(2^{D+d^*})$ regret bound. For the sleeping expert version, different experts are awake on every round and the conditions of the theorem do not directly apply. Analyzing this would likely require reasoning directly about model order selection guarantees as we have done.

[4] As a general note, all our analysis can easily be extended to the m-ary case. We present the binary case for simplicity.

prediction, the actual value Y_t is revealed, and the expected loss is modeled as $0-1$ loss depending on whether we get the prediction right. Formally, we have $\mathbf{l}_t = \begin{bmatrix} I[Y_t \neq 0] & I[Y_t \neq 1] \end{bmatrix}$, and the expected loss of the algorithm in round t is given by $\langle \mathbf{w}_t, \mathbf{l}_t \rangle = w_{t,1-Y_t}$. We also call this the expected *prediction error* of the algorithm. We denote as shorthand (for all $h \leq D$)

$$L_{t,\mathbf{f}} := \sum_{s=1}^{t} I[Y_s \neq \mathbf{f}(X_s(h))] \text{ for all } \mathbf{f} \in \mathcal{F}_h$$

$$L_{X,t,y} := \sum_{s=1}^{t} I[X_s = X; Y_s \neq y] \text{ for all } X \in \mathcal{X}^h, y \in \mathcal{X}$$

$$\mathbf{L}_{X,t} := \begin{bmatrix} L_{X,t,0} & L_{X,t,1} \end{bmatrix} \text{ for all } X \in \mathcal{X}^h.$$

2.1 Adaptive regret minimization and ContextTreeAdaHedge(D)

The traditional quantity of *regret* measures the loss of an algorithm with respect to the loss of the algorithm that possessed oracle knowledge of the best single "action" to take in hindsight, after seeing the entire sequence offline. In the context of online supervised learning, this "action" represents the best d^{th}-order Boolean function $\widehat{F}_d(T) \in \mathcal{F}_d$. The expected regret with respect to the best d^{th}-order tree expert is defined as $R_{T,d} := \sum_{t=1}^{T} \langle \mathbf{w}_t, \mathbf{l}_t \rangle - L_{T,\widehat{F}_d(T)}$.

Our algorithm is effectively an exponential-weights update on tree experts equipped with a *time-varying, data-dependent learning rate* and a suitable prior distribution on tree experts. We start by describing the structure of the prior distribution.

Definition 1. *For any non-negative-valued function $g : \{0, 1, \ldots, D\} \to \mathbb{R}_+ \cup \{0\}$, we define the prior distribution on all tree experts in $\mathcal{F}_D$, $\mathbf{w}_{1,\mathbf{f}}^{(\mathsf{tree})}(g) = \frac{\sum_{h=order(\mathbf{f})}^{D} g(h)}{Z(g)}$, where $Z(g)$ is the normalizing factor.*

We select a function $g(\cdot)$ and use the prior defined above to effectively downweight more complex experts. We will see that the choice of prior is crucial to recovering stochastic model selection.

A good *data-adaptive* choice of $\{\eta_t\}_{t\geq 1}$ has been an intriguing question of significant recent interest. The idea is that we want to learn the correct learning rate for the problem. We consider a particularly elegant choice based on the algorithm AdaHedge, that was defined for the simpler experts setting. We denote $\eta_{s_1}^{s_2} = \{\eta_s\}_{s=s_1}^{s=s_2}$ for shorthand.

Definition 2 ([DRVEGK14])**.** *The* AdaHedge *learning rate process $\{\eta_t\}_{t\geq 1}$ is described as*

$$\eta_t = \frac{\ln 2}{\Delta_{t-1}(\eta_1^{t-1})}, \tag{1}$$

where $\Delta_t(\eta_1^{t-1})$ is called the "cumulative mixability gap" at time t and is given by

$$\Delta_t(\eta_1^{t-1}) := \sum_{s=1}^{t} \delta_s(\eta_s) \text{ where} \tag{2}$$

$$\delta_s(\eta_s) := \langle \mathbf{w}_s(\eta_s), \mathbf{l}_s \rangle + \frac{1}{\eta_s} \ln \langle \mathbf{w}_s(\eta_s), e^{-\eta_s \mathbf{l}_s} \rangle. \tag{3}$$

We are now ready to describe our main algorithm.

Definition 3. *The algorithm* ContextTreeAdaHedge(D) *whose prior is derived from the function $g(\cdot)$ updates its probability distribution on tree experts as follows:*

$$w_{t,\mathbf{f}}^{(\mathsf{tree})}(\eta_t; g) = \frac{\left(\sum_{h=order(\mathbf{f})}^{D} g(h)\right) e^{-\eta_t L_{t,\mathbf{f}}}}{\sum_{\mathbf{f}' \in \mathcal{F}_D} \left(\sum_{h=order(\mathbf{f}')}^{D} g(h)\right) e^{-\eta_t L_{t,\mathbf{f}'}}}. \tag{4}$$

and learning rate update $\{\eta_t\}_{t\geq 1}$ made according to Equations (1) *and* (2)*.*

The algorithm ContextTreeAdaHedge(D) appears to have a prohibitive computational complexity of $\mathcal{O}(|\mathcal{F}_D|) = \mathcal{O}(2^{2^D})$. However, the distributive law enables a clever reduction in computational complexity to $\mathcal{O}(2^D)$. The main idea is that instead of keeping track of cumulative losses of all the 2^{2^D} functions in $\mathcal{F}_D$, represented by $\{L_{t,\mathbf{f}}\}_{\mathbf{f}\in\mathcal{F}_D}$, we only need to keep track of the cumulative losses of making certain predictions as a function of certain contexts, represented by $\{\{L_{x,t,y}\}_{y\in\mathcal{X}}\}_{x\in\mathcal{X}^D}$. This reduction was first considered for tree expert prediction in the worst-case [HS97], with a fixed learning rate $\eta > 0$, and can easily be extended to the broader class of exponential-weights updates. Proposition 2, which is stated and proved in Appendix C for completeness, shows that the update on probability distribution on *tree experts*, described in Equation (4) – can be equivalently written as a computationally faster update on probability distribution on *predictors*:

$$w_{t,y}(\eta_t; g) = \frac{\sum_{h=0}^{D} g'(h; \eta_t) e^{-\eta_t L_{X_t(h),t,y}}}{\sum_{h=0}^{D} g'(h; \eta_t) \left(\sum_{y\in\mathcal{X}} e^{-\eta_t L_{X_t(h),t,y}}\right)} \tag{5a}$$

$$\text{where } g'(h; \eta_t) = g(h) \prod_{x(h)\neq X_t(h)} \left(\sum_{y\in\mathcal{X}} e^{-\eta_t L_{x(h),t,y}}\right) \tag{5b}$$

The equivalence is in the sense that the expected loss incurred by updates (4) and (5a) is the same.

2.2 Potential generative assumptions on data

As we have mentioned informally, we would like to get greatly improved regret rates for data generated in a certain way (without apriori knowledge of such generation).

We work with the following general *stochastic, stationary, predictable condition* on our data.

Definition 4 (Stationary stochastic condition)**.** *We say that our data $(X_t, Y_t)_{t\geq 1}$ satisfies the stationary stochastic condition if the following hold:*

1. *The random vectors $\{(X_t, Y_t)\}_{t\geq 1}$ are identically distributed across $t \geq 1$ (not necessarily independent). We have $X_t \sim Q^*_D(\cdot)$, $Y_t|\{X_t(h), (X_{t-1}, Y_{t-1}), \ldots, (X_1, Y_1)\} \sim P^*(\cdot|X_t(h))$ for all $X_t(h) \in \mathcal{X}^h$ and $h \in \{0, 1, \ldots, D\}$.*

We denote the marginal distribution on $X_t(h)$ by $Q^*_h(\cdot)$. For this setting, it is natural to define the best "external predictor" for any $h \leq d$:

$$f^*(x(h)) :\in \arg\max_{y\in\mathcal{X}} P^*(y|x(h)) \text{ for all } x(h) \in \mathcal{X}^h, \tag{6}$$

Based on this, we also define the important notions of asymptotic *unpredictability* for all model orders $h \in \{0, 1, \ldots, D\}$. The definitions and notation are directly inspired by information-theoretic limits on sequence compression and prediction [FMG92].

Definition 5 ([FMG92])**.** *For data $(X_t, Y_t)_{t\geq 1}$ satisfying the stationary stochastic condition, we define its asymptotic unpredictability under the h^{th}-order predictive model by –*

$$\pi^*_h := \sum_{x(h)\in\mathcal{X}^h} Q^*_h(x(h)) \left[1 - \max_{y\in\mathcal{X}}\{P^*(y|x(h))\}\right] \tag{7}$$

In general, the sequence $\{\pi^*_h\}_{h=0}^D$ is decreasing in h. We now formally define a formally realizable sequence.

Definition 6 (Realizable sequence)**.** *We say that our data is realized from a d^{th}-order model if we have $Y_t|\{X_t, (X_s, Y_s)_{s=1}^{t-1}\} \sim P^*(\cdot|X_t(d))$ for all t and all $X_t \in \mathcal{X}^D$. This implies that $\pi^*_h = \pi^*_d < 1/2$ for all $h \geq d$.*

The *realizability condition* implies that Y_t is independent of all previous observations given $X_t(d)$. This includes simple environments like *contextual prediction* where pairs (X_t, Y_t) are drawn iid – but also *sequence prediction* under a d^{th}-memory Markov process. We assume that the best d^{th}-order predictor is unique[5], i.e.

$$P^*(f^*(x(d))|x(d)) > P^*(y|x(d)) \text{ for all } y \neq f^*(x(d)) \text{ and for all } x(d) \in \mathcal{X}^d.$$

and denote the parameter[6]

$$\beta(x(d)) = P^*(f^*(x(d))|x(d)) \tag{8}$$

$$\beta^* := \min_{x(d)\in\mathcal{X}^d} \beta(x(d)). \tag{9}$$

We also empirically consider *hidden Markov models* that are not actually realized from any finite-memory model (theoretically, they have infinite-memory dependence), but are approximable by them.

3 Main results

Different choices of the function $g(\cdot)$ used to describe the prior distribution on tree experts yield vastly different results. Consider the choice $g_{\mathsf{unif}}(h) := \mathbb{I}[h = D]$, which corresponds to the typical *prior-free* implementation of exponential weights (i.e Equation (4) with a uniform prior). With this choice, Proposition 1 in Appendix A.2.3 describes the "best-of-both-worlds" bound that we obtain: worst-case regret $\mathcal{O}(\sqrt{T}\cdot 2^D)$, and regret $\mathcal{O}(2^{2D})$ in the stochastic case. Note that the stochastic regret bound, while constant and thus independent of the horizon T, is highly suboptimal in its dependence on the maximum model order D. The bound does not improve for drastically simpler cases; for example, $Y_t \sim$ i.i.d and Y_t is independent of X_t.

We now consider the realizable case in which the data is actually coming from model order d. We study the algorithm ContextTreeAdaHedge(D) with the following choice of model-order-proportional prior function.

$$g_{\mathsf{prop}}(h) = 2^{-2^{h+1}} \tag{10}$$

Our first result shows that the algorithm with this choice of prior helps us effectively learn the model order while staying worst-case robust.

Theorem 1. *1. For any sequence $\{X_t, Y_t\}_{t=1}^T$, the algorithm* ContextTreeAdaHedge(D) *with prior defined according to function $g_{\mathsf{prop}}(\cdot)$ gives us regret rate*

$$R_{T,d} = \mathcal{O}\left(\sqrt{T}2^d\right) \tag{11}$$

[5]This is the *Tsybakov margin condition* [T+04] that is *required* for learnability. If, for e.g., $\pi^*_d = 1/2$, we would unavoidably suffer a $\sqrt{T}$ regret rate [CBL06, Chap. 3,].

[6]Note that the uniqueness of best-predictor assumption directly implies that $\beta^* > 1/2$, since we are working with a binary alphabet.

with respect to the best d^{th}-order tree expert in hindsight, and for every $d \in \{0, 1, \ldots, D\}$.

2. *Consider any $\delta \in (0, 1]$. Let the stationary stochastic sequence $(X_t, Y_t)_{t\geq 1}$ satisfy the d^{th}-order realizability condition with parameter β^*. Denote $\alpha_{d-1,d} := \frac{\pi^*_{d-1} - \pi^*_d}{2}$. Then,* ContextTreeAdaHedge(D) *with prior function $g_{\mathsf{prop}}(\cdot)$ incurs regret with probability greater than or equal to $(1-\delta)$:*

$$R_{T,d} = \mathcal{O}\Big(2^{2d}\Big(\frac{d^2}{\alpha^2_{d-1,d}} \ln\left(\frac{d}{\alpha^2_{d-1,d}\delta}\right) \tag{12}$$

$$+ \frac{D \cdot d}{(\alpha^*)^2} \ln\left(\frac{D}{\alpha^*\epsilon}\right)\Big)\Big) \tag{13}$$

where $\alpha^* = \min\{\alpha_{d-1,d}, 2\beta^* - 1)\}$.

The proof of Theorem 1 follows from a careful combination of adversarial-stochastic interpolation and structural risk minimization, and are deferred to the appendix. We provide an intuitive sketch of the proof in Section 4. The stochastic result in Theorem 1 hold for a broad class of stochastic sequences; three special cases of which we list below:

1. Independent and identically distributed data.
2. Universal sequence prediction ($X_t = Y_{t-D}^{t-1}$) with Y_t following a d-memory mixing Markov process.
3. Universal sequence prediction with Y_t generated by a mixing hidden Markov model[7].

Appendix B provides a detailed exposition of the application of our results to these types of processes.

Theorem 1 shows that the efficient algorithm ContextTreeAdaHedge(D) obtains comparable regret rates as would be achieved by an algorithm that had oracle knowledge about the presence of stochasticity *and* the model order. This is the strongest possible side information that an algorithm could conceivably possess keeping the online learning problem non-trivial.

In simulation, we also demonstrate the empirical advantage of algorithms that are able to adapt to model complexity. These are better in terms of regret and overall prediction error than other adversarially robust approaches that do not adapt the benchmark, and are interestingly comparable to purely greedy stochastic model selection approaches. The advantage of offline data-driven model selection is well established, and we see this advantage even more naturally while measuring regret, or even average prediction error, in online learning.

[7]Since the "best expert" is of model order D, we get a pessimistic regret bound in this case.

4 Proof sketch of Theorem 1

Initially, we mirror the established style of "best-of-both-worlds" results. The first step is always to prove a regret bound that is dependent on the data $\{(X_t, Y_t)\}_{t=1}^T$; in particular, a bound of the form $R_{T,d} = \mathcal{O}\left(\sqrt{V_T(\eta_1^T; g_{\mathsf{prop}}) \cdot 2^d}\right)$ where $V_T(\eta_1^T; g_{\mathsf{prop}})$ represents the cumulative variance of loss incurred by the algorithm. Curiously, we are easily able to get a bound (commonly called a second-order bound) that is adaptive to the model order using exponential weights with a prior[8]!

The *cumulative variance term* V_T is telling us something about how random the randomized updates in the algorithm are. In the worst case, $V_T \leq \frac{T}{4}$ and we automatically recover the adversarial result; but this term can be significantly smaller. It is easy to see that this randomness will greatly reduce when the losses are *stochastic* in the sense that one tree expert looks consistently better than the others. It will also reduce in the presence of a favorable prior $g_{\mathsf{prop}}(\cdot)$ if that best expert possesses simpler structure. However, existing analysis of efficient algorithms [CBMS07, EKRG11, DRVEGK14, LS15, KVE15] only exploits the former property, and not the latter – thus giving a pessimistic scaling of $\mathcal{O}(2^D)$ or $\mathcal{O}(2^{2D})$ for our problem.

Our main technical contribution is tackling the more difficult problem of finely controlling the cumulative variance of the algorithm under a favorable prior – showing that it in fact scales as the significantly smaller $\sqrt{V_T} = \mathcal{O}(2^d)$. We achieve this by making an explicit connection to *probabilistic model selection by complexity regularization*. To see this, consider Equation (4) written equivalently as the optimization problem in the Follow-the-Regularized Leader [SS+12] update:

$$\mathbf{w}_t^{(\mathsf{tree})} := \arg\min_{\mathbf{w}^{(\mathsf{tree})}} \Big[\langle \mathbf{w}^{(\mathsf{tree})}, \mathbf{L}_t^{(\mathsf{tree})}\rangle + \tag{14}$$

$$\frac{1}{\eta_t}\left(-\underbrace{H(\mathbf{w}^{(\mathsf{tree})})}_{\text{entropy}} + \underbrace{\langle \mathbf{w}^{(\mathsf{tree})}, \mathbf{C}^{(\mathsf{tree})}\rangle}_{\text{complexity}}\right)\Big], \tag{15}$$

where $C_{\mathbf{f}}^{(\mathsf{tree})} := 2^{\mathrm{order}(\mathbf{f})} \log 2$ and $H(\cdot)$ denotes the

[8]The careful reader will notice that there is nevertheless a suboptimality in the dependence on d as compared to the second-order bound obtained by algorithms like Squint [KVE15] and AdaNormalHedge [LS15].

entropy functional on a probability distribution over a discrete-valued random variable. Viewed this way, the algorithm ContextTreeAdaHedge(D) updates to minimize the cumulative loss *adaptively* regularized with entropy (to protect against a potential adversary) and model complexity (to adapt to simpler models faster).

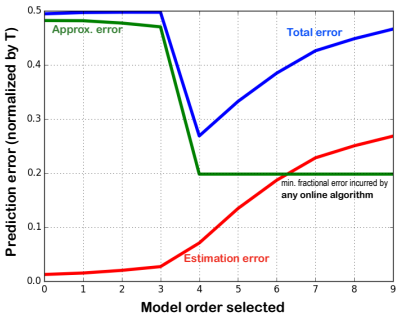


Figure 1. Illustration of the tradeoff between estimation error and approximation error for various choices of model order. The true model order is 4 and the plot made is of performance of uniform-prior ContextTreeAdaHedge(h) for different choices of h, measured at $T = 1500$.

Figure 1 illustrates the classical tradeoff in *stochastic* model selection in an example where the true model order is 4 – the estimation error increases with model order, and the approximation error decreases with model order, and plateaus out at the true model order 4 (note that this is the minimum average prediction error that any online learning algorithm should be expected to pay). Clearly, the true model order minimizes the appropriate combination of estimation error and approximation error.

Our algorithm additionally retains adversarial robustness and can be interpreted as *probabilistically* selecting a model in every round. Explicitly, it maintains a *meta-expert layer* where the meta-experts correspond to *algorithms* (AdaHedge(h) corresponding to every model order h)[9]. Effectively, we show a *probabilistic model selection guarantee*, i.e. we can pick the true model with high probability. We do this by ruling out lower and higher-order models alike. On one hand, the more (superfluously) complex a model is, the more it is going to *overfit*, contributing to unnecessary accumulated regret – however, the more its unfavorable prior drags it down to rule it out. On the other hand, the more (unnecessarily) simple a model is, the worse it is going to *approximate* – and since this approximation error is directly penalized in Equation (14), the less likely it is to be picked.

The reason the classical analysis of stochastic model selection [Mas07] does not directly apply here is in the requirement to adapt *multi-fold*, between adversity and stochasticity of varying model complexity. The primary technical difficulty is in characterizing the extent of adaptivity, encapsulated in the time-varying, *data-dependent* learning rate which is known to be notoriously difficult to track [DRVEGK14, KVE15, KGvE16]. It is perilous for the learning rate to remain too high (in which case the algorithm is effectively greedy, and overfits for too long), or sink too low (in which case we remain stuck selecting poorly fitting models). Remarkably, we are able to carefully sandwich the learning rate in high probability to ensure model selection, *in both cases* using the fundamental inverse relationship between the learning rate and regret that is used to *learn the learning rate* in adaptive algorithms. This clever relationship has been exploited to achieve stochastic-adversarial adaptivity; here, we show that its power is significantly higher, in being able to additionally adapt to model complexity[10]. Once the (high-probability) model selection guarantee is obtained, analysis proceeds with slight generalization of the AdaHedge analysis [EKRG11] to the tree experts setting.

5 Simulations

We now provide a brief empirical illustration of the power of two-fold adaptivity to stochasticity *and model complexity* with ContextTreeAdaHedge(D) equipped with the prior function $g_{\mathsf{prop}}(\cdot)$. We showcase two examples of stochastic sequences – one that is actually generated from a 3-memory Markov process with parameters:

$$Y_t \sim \mathrm{Ber}\left(0.6 \cdot (Y_{t-3} \oplus Y_{t-2} \oplus Y_{t-1}) + 0.2\right)$$

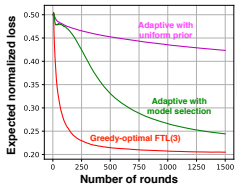
and one that is generated from a hidden Markov model with parameters:

$$\text{Hidden state evolution } W_{t+1} \sim \mathrm{Ber}\left(|W_t - 0.9|\right)$$
$$Y_t|W_t = 0 \sim \mathrm{Ber}(0.2)\ ,\ Y_t|W_t = 1 \sim \mathrm{Ber}(0.9).$$

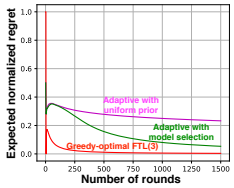
Figure 2 compares our model-adaptive algorithm, ContextTreeAdaHedge(D) (with prior function $g_{\mathsf{prop}}(\cdot)$ and $D = 8$), with robust and greedy algorithms, for the above examples of stochastic sequences. Under the sequence generated by a 3-memory Markov process, we compare our

[9]The master algorithm framework was also considered explicitly for similar problems in the contextual bandit regime [ALNS17], where the primary difficulty is paucity of representative samples for each algorithm.

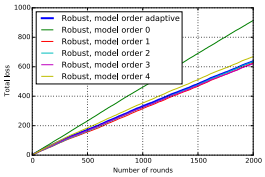
[10]In fact, the same conceptual idea underlies the approaches to *learn the learning rate*, prevalent in Squint, MetaGrad and AdaNormalHedge.



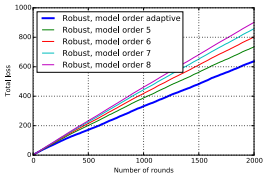
(a) Total loss as a function of T under 3-memory Markov process.



(b) $R_{T,3}$ as a function of T under 3-memory Markov process.



(c) Total loss as compared to robust lower-order model selection under HMM.



(d) Total loss as compared to robust higher-order model selection under HMM.

Figure 2. Comparison of model-adaptive ContextTreeAdaHedge(D) with robust and greedy algorithms.

algorithm to the optimal online algorithm with *oracle knowledge of this structure* (the greedy Follow-the-Context-Leader(3)); and uniform-prior ContextTreeAdaHedge(D), which adapts to stochasticity but not model order. We consider the expected normalized regret $\frac{R_{T,3}}{T}$ and expected normalized *cumulative loss* of the algorithms. We observe that model adaptivity makes a significant difference to regret and overall loss: ContextTreeAdaHedge(D) equipped with uniform prior does not adapt to model order, and pays for it with loss (regret) accumulated due to overfitting. Further, our main adaptive algorithm, which is effectively *learning the presence of stochasticity and the right model order* is remarkably competitive with the optimal Follow-the-Leader algorithm, which possesses oracle knowledge of both.

We also include the HMM example to display that our algorithm appears to be model-adaptive even in stochastic environments that are not exactly realized from a finite-memory model. In these cases, we care about the overall loss incurred by an algorithm, as there is no natural benchmark for regret minimization. We consider a HMM with quickly transitioning hidden states, for which the 1-memory model appears to be the best fit in hindsight[11]. Our adaptive algorithm, without knowledge of the parameters of the HMM, tracks such a model in terms of overall loss. We observe that lower-order models (such as model order 0) underfit and higher-order models overfit for this simple example. We should expect ContextTreeAdaHedge(D) to adapt to higher-order models as more rounds are played in a way to minimize the estimation-approximation tradeoff; it would be interesting to show this theoretically in future work.

[11] An additional example of a HMM with slowly transitioning hidden states is also included in Appendix E; there, the best model order fit is slightly higher.

6 Discussion

Summarization of contributions We study the problem of binary contextual prediction (easily generalizable to m-ary contextual prediction) with $0-1$ loss. We design an algorithm that incorporates recent advances in adaptivity with contextual pre-weighting, and show that we can simultaneously adapt to the model order complexity *and* the existence of stochasticity. By adaptively recovering the stochastic structural risk minimization framework, we are able to select the right d^{th}-order model for the stochastic process, and obtain regret rates that are competitive with those of the optimal greedy algorithm which knows not only the presence of stochastic structure, but the exact value of d. Our analysis is interpretable and directly analyzes the probability with which we select a particular model order.

Future directions Many future directions arise from this work. First, we acknowledge that the regret rate we obtain is not exactly optimal, particularly in terms of the multiplicative factor of d in the exponent. This suboptimality appears to arise from an overly conservative regularization of model order complexity (because the same regularization parameter is also used for entropy); it would be interesting to design an algorithm with two different adaptive regularization parameters. Second, we are hopeful that the concept of probabilistic structural risk minimization introduced here can be leveraged to recover the full stochastic model selection framework in which higher-order models are provably selected as more rounds are played. We thus hope that it can be generalized in multiple ways. For example, we desire high-probability bounds on overall prediction error in the setting where the sequence is stochastic and not realized by a finite-order model, but still approximable by one. Finally, it would be interesting to extend the positive results obtained here to oracle-efficient online supervised classification and/or regression, as has been noted by others [FKMS17], as well as limited information feedback.

References

[ADBL11] Alekh Agarwal, John C Duchi, Peter L Bartlett, and Clement Levrard. Oracle inequalities for computationally budgeted model selection. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 69–86, 2011.

[ALNS17] Alekh Agarwal, Haipeng Luo, Behnam Neyshabur, and Robert E Schapire. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, pages 12–38, 2017.

[Boz87] Hamparsum Bozdogan. Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, 52(3):345–370, 1987.

[CBFH+97] Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997.

[CBL06] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[CBMS07] Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3):321–352, 2007.

[DRVEGK14] Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15(1):1281–1316, 2014.

[EKRG11] Tim V Erven, Wouter M Koolen, Steven D Rooij, and Peter Grünwald. Adaptive hedge. In *Advances in Neural Information Processing Systems*, pages 1656–1664, 2011.

[FKMS17] Dylan J Foster, Satyen Kale, Mehryar Mohri, and Karthik Sridharan. Parameter-free online learning via model selection. In *Advances in Neural Information Processing Systems*, pages 6022–6032, 2017.

[FMG92] Meir Feder, Neri Merhav, and Michael Gutman. Universal prediction of individual sequences. *IEEE transactions on Information Theory*, 38(4):1258–1270, 1992.

[HS97] David P Helmbold and Robert E Schapire. Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, 27(1):51–68, 1997.

[KGvE16] Wouter M Koolen, Peter Grünwald, and Tim van Erven. Combining adversarial guarantees and stochastic fast rates in online learning. In *Advances in Neural Information Processing Systems*, pages 4457–4465, 2016.

[KVE15] Wouter M Koolen and Tim Van Erven. Second-order quantile methods for experts and combinatorial games. In *Conference on Learning Theory*, pages 1155–1175, 2015.

[LS15] Haipeng Luo and Robert E Schapire. Achieving all with no parameters: Adanormalhedge. In *Conference on Learning Theory*, pages 1286–1304, 2015.

[Mas07] Pascal Massart. *Concentration inequalities and model selection*, volume 6. Springer, 2007.

[OP16] Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. In *Advances in Neural Information Processing Systems*, pages 577–585, 2016.

[Ora14] Francesco Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems*, pages 1116–1124, 2014.

[RS13] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. 2013.

[SKLV18] Vatsal Sharan, Sham Kakade, Percy Liang, and Gregory Valiant. Prediction with a short memory. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1074–1087. ACM, 2018.

[SS+12] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[T+04] Alexander B Tsybakov et al. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

[Vap99] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[vEK16] Tim van Erven and Wouter M Koolen. Metagrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems*, pages 3666–3674, 2016.