

Prospective Learning in Retrospect

Yuxin Bai^{1,*} [0009-0007-0689-7487], Cecelia Shuai^{1,*} [0009-0003-9882-7566], Ashwin De Silva^{1,*} [0000-0002-6406-7090], Siyu Yu¹ [0009-0001-1459-2567], Pratik Chaudhari² [0000-0003-4590-1956], and Joshua T. Vogelstein¹ [0000-0003-2487-6237]

¹ Johns Hopkins University

² University of Pennsylvania

{ybai31, xshuai3, ldesilv2, jovo}@jhu.edu, pratikac@seas.upenn.edu

Abstract. In most real-world applications of artificial intelligence, the distributions of the data and the goals of the learners tend to change over time. The Probably Approximately Correct (PAC) learning framework, which underpins most machine learning algorithms, fails to account for dynamic data distributions and evolving objectives, often resulting in suboptimal performance. Prospective learning is a recently introduced mathematical framework that overcomes some of these limitations. We build on this framework to present preliminary results that improve the algorithm and numerical results, and extend prospective learning to sequential decision-making scenarios, specifically foraging. Code is available at: <https://github.com/neurodata/prolearn2>.

Keywords: Distribution Shifts · Out-of-Distribution Generalization · Learning Theory · Sequential Decision-Making

1 Introduction

Learning involves updating decision rules or policies, based on past experiences, to improve future performance. The Probably Approximately Correct (PAC) learning [1, 2] framework has led to the development of learning algorithms that provably minimize the risk (expected loss) over unseen future samples during inference. When proving such guarantees, PAC learning assumes that data is independently and identically (*iid*) distributed according to a fixed distribution at training and inference time.

While this assumption has been useful, it is rarely held true in practice. In fact, the future is more likely to be different from the past as distributions of data and goals of the learner may change over time. Therefore, the true hypothesis can be time-variant, and classical PAC learning does not address this situation. Although sub-disciplines such as transfer learning [3], continual/lifelong learning [4–7], online learning [8], meta-learning [9, 10], sequential decision-making [11, 12], forecasting [13], reinforcement learning [14–17], out-of-distribution generalization [18] have introduced attractive solutions that retrospectively adapt to distributions that change over time, they often fail to anticipate and generalize to future even when data evolves in simple but predictable ways as shown in [19, 20].

* Equal contribution

Prospective learning [19, 20] is a recently developed mathematical framework to bridge this gap. Instead of data arising from a fixed distribution, prospective learning assumes that data is drawn from a stochastic process, that the loss considers the future, and that the optimal hypothesis may change over time. A prospective learner uses samples received up to some time $t \in \mathbb{N}$ to output an infinite sequence of predictors, which it uses for making predictions on data at all future times $t' > t$. An exhaustive comparison between prospective learning and related sub-fields of machine learning literature is provided in De Silva et al. [19].

We build on the prospective learning framework and introduce several preliminary results. The rest of the paper is organized as follows; Section 2 provides a concise summary of the prospective learning framework, Section 3 presents several empirical observations on deep learning-based prospective learners, Section 4 introduces a decision-tree based prospective learner and demonstrates its performance, and finally, Section 5 introduces prospective foraging, showing that the prospective learning framework extends beyond supervised learning and performs competitively with reinforcement learning in preliminary results.

2 Preliminaries

Let input and output be denoted by $x_t \in \mathcal{X}$ and $y_t \in \mathcal{Y}$, respectively. Let $z_t = (x_t, y_t)$. We will find it useful to denote the observed data, $z_{\leq t} = \{z_1, \dots, z_t\}$, and the unobserved data, $z_{>t}$. In contrast to PAC learning, t is not just a dummy variable, but rather, indexes time. We therefore define the data triple (x_t, y_t, t) , and augment the input space to include the time of the input, $\mathcal{X} \leftarrow \mathcal{X} \times \mathcal{T}$. Consider an infinite sequence of hypotheses $h \equiv (h_1, \dots, h_t, h_{t+1}, \dots)$ where $h_t : \mathcal{X} \rightarrow \mathcal{Y}$. The hypothesis class \mathcal{H} is the space that contains such sequences. With a slight abuse of notation, we will refer to a sequence of hypotheses h as a *hypothesis*, where each element of this sequence $h_t : \mathcal{X} \mapsto \mathcal{Y}$.³

Loss The instantaneous loss, $\ell(h(x), y)$, is a map from $\mathcal{Y} \times \mathcal{Y}$ to \mathbb{R} . In all real-world problems we care about the integrated future loss. Let $w(i)$ be a non-increasing non-negative weighting function that sums to one, that is $\sum_i w(i) = 1$ and $0 \leq w(i) \leq 1 \forall i$. We thus define prospective loss as

$$\bar{\ell}(h, z_{>t}) = \sum_{s>t} w(s-t) \ell(h(x_s), y_s) \quad (1)$$

which is the weighted cumulative loss over all the future data.⁴

Taking the expectation of Eq. (1) over the future conditioned on the observed data $z_{\leq t}$, we arrive at the prospective risk at time t ,

$$R_t(h) = \mathbb{E} [\bar{\ell}_t(h, Z) | z_{\leq t}] = \int \bar{\ell}_t(h, Z) d\mathbb{P}_{Z|z_{\leq t}}. \quad (2)$$

³ One could also think of prospective learning as using a single time-varying hypothesis $h : \mathbb{N} \times \mathcal{X} \mapsto \mathcal{Y}$, i.e., the hypothesis takes both time and the datum as input to make a prediction.

⁴ If we let $w(i)$ be a constant function of i , we recover the classical loss in PAC learning.

We consider a family of stochastic processes \mathcal{Z} to be strongly prospective-learnable if there exists a learner that likely returns an approximately optimal hypothesis (with a risk close to the Bayes risk R_t^*) after observing enough past data $z_{\leq t}$ from any process $Z \in \mathcal{Z}$. Theorem 1 from De Silva et al. [19] guarantees that under certain general assumptions, a slightly modified Empirical Risk Minimizer (ERM) a learner returning the hypothesis

$$\hat{h} = \arg \min_{h \in \mathcal{H}_t} \sum_{t' > 0}^t \bar{\ell}(h, z_{>t'}) = \arg \min_{h \in \mathcal{H}_t} \sum_{t' > 0}^t \sum_{s > t'} w(s - t') \ell(h_{t'}(x_s), y_s). \quad (3)$$

is a strong prospective learner for a finite family of stochastic processes under certain assumptions. We refer to this learner as Prospective ERM.

To implement prospective ERM in practice, one may modify a predictor (e.g. a multi-layer perceptron) to take $(\varphi(s), x_s)$ as the input and train it to predict the label y_s , where $\varphi(s)$ is a suitable embedding on the time s . We refer to this predictor as *Prospective-MLP*. Inspired by the positional encoding of the Transformer [21], De Silva et al. [19] have used the time-embedding,

$$\varphi_f(t) = (\sin(\omega_1 t), \dots, \sin(\omega_{d/2} t), \cos(\omega_1 t), \dots, \cos(\omega_{d/2} t)), \quad (4)$$

where $\omega_i = \pi/i$ for $i = 1, \dots, d/2$. It is shown that the prospective-MLP converges to the Bayes risk on certain stochastic processes over synthetic and image data.

Prospective Learning Scenarios

Depending on the nature of the stochastic process, one can consider 4 scenarios of prospective learning; (1) Data is independent and identically distributed, (2) Data is independent but not identically distributed, (3) Data is neither independent nor identically distributed (e.g. Markov Processes), and (4) Future depends on the current prediction (Stochastic Decision Processes). Scenario 1 puts us back in the PAC learning setting. Scenario 4, on the other hand, is arguably a special case of Scenario 3 and has implications for reinforcement learning and control. As we will introduce in Section 5, prospective foraging is closely related to this scenario.

Example Stochastic Processes

Here we describe three stochastic processes we will consider in the experiments outlined in Sections 3 and 4.

First, consider two binary classification distributions (“tasks”) (see Fig. 1 (Left)). The inputs for both tasks are drawn from a uniform distribution on the set $[-2, -1] \cup [1, 2]$. Ground-truth labels correspond to the sign of the input for Task 1, and the negative of the sign of the input for Task 2. The process switches

⁵ This is a slight abuse of notation, because previously ℓ mapped from a hypothesis and data corpus, but now we are saying that it maps from a hypothesis and a collection of random variables. We have used the shorthand $\mathbb{E}[Y | x]$ for $\mathbb{E}[Y | X = x]$. Note that $\mathbb{E}[Z | z_{\leq t}]$ is equivalent to $\mathbb{E}[Z_{>t} | z_{\leq t}]$ because the past z ’s are given.

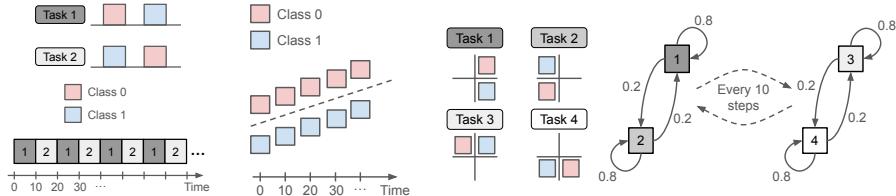


Fig. 1: Pictorial depictions of 3 types of stochastic processes considered in our experiments. (**Left**) Periodic Process, (**Middle**) Linear process, and (**Right**) Hierarchical hidden Markov Process. The periodic and linear processes belong to scenario 2 whereas the hierarchical hidden Markov process is an instance of scenario 3.

the two tasks every 10 time steps, resembling a reversal learning problem. We refer to this as the “periodic” process.

The second is a stochastic process whose marginal distribution at time t is defined as follows: The input x_t is drawn from a uniform distribution over $[\epsilon t + 10, \epsilon t + 11] \cup [\epsilon t - 10, \epsilon t - 11]$ and its label y_t is 0 if $x_t > t$ and 1 otherwise. In other words, it is a process where the task is hiking up a slope with a small gradient ϵ (see Fig. 1 (Middle)). This process yields an infinite number of tasks, in contrast to the first one, where there are finite (two) tasks. Thus, we refer to it as the “infinite task process”.

The third and final process includes four tasks that are created using 2-dimensional inputs as shown in Fig. 1 (Right). After every 10 time-steps, a different Markov chain would govern the transitions among tasks (one Markov chain for tasks 1 and 2, and another for tasks 3 and 4 as illustrated in the figure). Therefore, the data distribution is effectively distributed according to the hierarchical hidden Markov model. We refer to it as the “dependent structured task process”.

Relationship between Continual and Prospective Learning

Although continual and prospective learning both involve learning over sequences of tasks, they differ fundamentally in their objectives and assumptions. As formalized in Section 2, the goal of a prospective learner is to perform well on future tasks. In contrast, the objective in continual learning, though typically less formally defined, is to maintain good performance on previously seen tasks and avoid catastrophic forgetting. Continual learning often assumes a task-aware setting, where the learner receives a batch of data per task along with the task identity. Prospective learning, by contrast, does not assume access to such task labels or boundaries. A notable exception is task-agnostic online continual learning [22], which operates under similar assumptions to prospective learning. However, De Silva et al. [19] shows that such methods still fail to improve upon chance-level prospective risk when learning the periodic process above. Furthermore, continual learning benchmarks typically involve sequences of unrelated tasks without any predictable structure, whereas prospective learning is meaningful when the tasks evolve over time in a predictable manner. To address this gap, we introduce simple yet representative benchmarks designed to assess the ability of learners to generalize to future tasks. For additional experiments and a deeper comparison of prospective learning with related paradigms, we refer the reader to De Silva et al. [19].

Training and Evaluating Learners

The next two sections present experiments including Prospective-MLPs and Prospective-Trees, and a time-agnostic Follow-the-Leader (FTL) baseline that minimizes empirical risk over all past data without incorporating time as an input. When training and evaluating these learners, we roughly follow the steps detailed in the Section 6 of De Silva et al. [19].

3 Several Empirical Observations on Prospective-MLPs

Prospective-MLP prevails under heterogeneous sampling.

The experiments in De Silva et al. [19] assume homogeneous past data where exactly one sample is received at each time step. This assumption overlooks more realistic scenarios where samples may be missing or multiple samples may be available per time step. To model the heterogeneity of sampling, we assume that the number of samples received from the process at each time step is distributed according to a Poisson distribution with $\lambda = 1$. We train Follow-the-Leader (FTL) and prospective-MLP on data collected this way and compare them against their counterparts trained on homogeneous data (see Fig. 2). It is evident that prospective-MLP manages to secure a good prospective risk regardless of how the data is sampled.

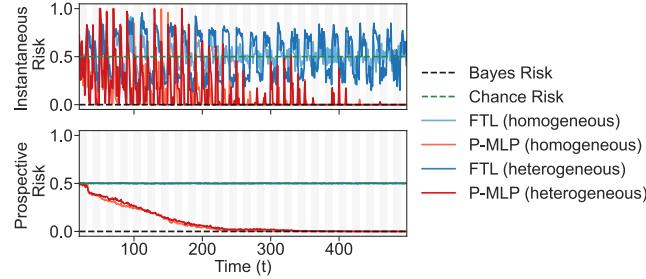


Fig. 2: Instantaneous (**top**) and prospective (**bottom**) risks of Follow-the-Leader (FTL, blue) and Prospective-MLP (P-MLP, red) trained on homogeneously (lighter shade) and heterogeneously (darker shade) sampled data from the periodic process. Homogeneous sampling is where you get exactly one sample each time step. In heterogeneous sampling, there can be missing samples and/or multiple samples available per time step.

Prospective-MLP prevails in infinite task scenarios.

Experiments in De Silva et al. [19] are mostly based on stochastic processes that include several tasks that periodically switch between each other. On such processes, it has been shown that the Prospective-MLPs equipped with the time-embedding defined by Eq. (4) are able to achieve a low prospective risk and generalize over the future. It is intuitive that a time-embedding comprised of Fourier basis functions is appropriate for a periodic process assuming that it contains the function with the true switching frequency.

However, aside from capturing periodic patterns, the utility of a Fourier-based time embedding is likely to be limited. To demonstrate this, we consider the linear process (see Section 2 and Fig. 1 (Middle)). There, we get a new task at each time and the task evolves according to a linear trend in time. The prospective learner must exploit this trend in order to generalize over the future.

To illustrate the effect the choice of time-embedding has on the learner, we train two Prospective-MLPs, one with the Fourier embedding from Eq. (4) and the other with a time-embedding based on monomial basis functions given by $\varphi_m(t) = (t, t^2, t^3, \dots, t^d)$. We repeat the same routine with the periodic process (see Section 2 and Fig. 1 (Left)) as well.

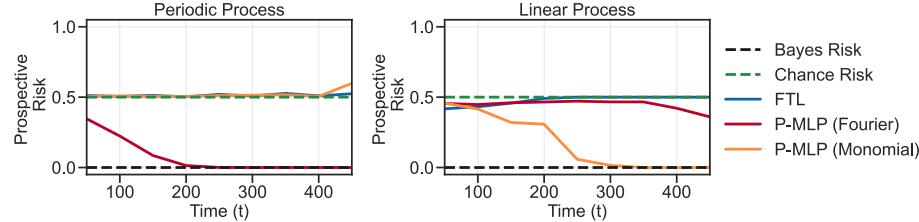


Fig. 3: Prospective risk of Follow-the-Leader (FTL), and Prospective-MLP with Fourier embeddings, and Prospective-MLP with monomial embeddings on periodic (Right top) and linear (Right bottom) processes. Prospective-MLP with Fourier embeddings performs best on the periodic process, whereas the variant with monomial embeddings achieves the best performance on the linear process.

As expected, the Prospective-MLP with the monomial embedding outperforms other learners trained on the samples from the linear process with an infinite number of tasks. However, it fails to perform well on the periodic process, where the Fourier embedding is more appropriate. The key takeaways from this experiment is that prospective learning can perform well even when there are an infinite number of tasks, but to do so, it must leverage an appropriate time-embedding for the underlying process.

Prospective-MLPs can be trained in a streaming or online manner.

So far, we have considered scenarios where Prospective-MLPs are trained in an offline or batched setting, using a fixed dataset of past samples drawn from the process. This requires the learner to have access to a memory where it may store the dataset used for training. When there are constraints on the memory allowed for the learner, batched-learning is no longer feasible. Here, we consider an extreme but realistic setting, where the learner will see the sample drawn from the process at time t only once. Therefore, the learner is expected to perform a parameter update after observing each new sample. In Fig. 4, we plot the risk of a Prospective-MLP that was trained in this manner over the data sampled from the periodic process. Notice that the batch-trained Prospective-MLP converges to the optimal risk within approximately 250 samples (see Fig. 2 (Bottom)), whereas its online-trained counterpart requires nearly 10 times as many samples to reach the same level of performance. This is expected as the model is exposed to the same training datum more than once during batched-training.

4 Prospective Forests

4.1 Motivation and background

Decision forests, including Random Forests (RF) and Gradient Boosted trees (GBTs) continue to empirically outperform deep learning methods on tabular and vector-valued data [23, 24] while offering superior interpretability. However, most

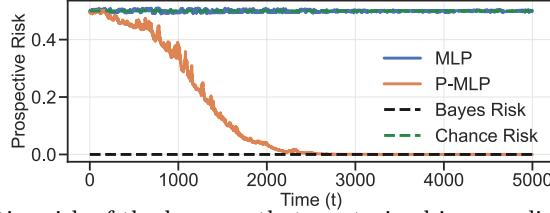


Fig. 4: Prospective risk of the learners that are trained in an online manner on data from the periodic process.

existing results are for problems under the assumptions of the PAC framework [25]. In addition to the strong theoretical guarantees, including universal consistency [25–27], decision forests can be efficiently trained in parallel or sequentially [28]. Motivated by these strengths, we extend decision forests to the prospective learning regime. Our preliminary results indicate that Prospective Decision Trees perform comparably to the deep learning-based Prospective-MLP.

Conventional CART (Classification and Regression Trees), as defined in Breiman et al. [28], is a greedy algorithm that builds a hierarchical structure through recursive binary splitting. We define a prospective variant of CART in the following.

Definition 1 (Prospective CART). Consider \mathcal{Z} to be a finite family of stochastic processes. Suppose there is an increasing sequence of hypothesis class $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots$ with each $\mathcal{H}_t \subseteq (\mathcal{Y}^{\mathcal{X}})^{\mathbb{N}}$. $\mathcal{H}_t^{\text{tree}}$ is a subset of \mathcal{H}_t and is the collection of all hypothesis $h \in \mathcal{H}_t^{\text{tree}}$ returned by decision trees regressor. We define Prospective CART as the learner minimizes the empirical risk over past data $z \leq t$, i.e.:

$$\hat{h} = \arg \min_{h \in \mathcal{H}_t^{\text{tree}}} \sum_{t' > 0}^t \sum_{s > t'} w(s - t') \ell(h_{t'}(x_s), y_s), \quad (5)$$

where $w(i)$ is non-increasing non-negative weighting function defined in Section 2

Naturally, random forest is a randomized ensemble of decision trees; analogously, aggregating prospective trees yields a prospective forest. However, another ensemble technique, GBTs often outperform RF [29, 30] in certain PAC settings. Therefore, we also introduce a prospective version of GBTs.

Definition 2 (Prospective Gradient Boosted Trees). Moreover, based on the definition of prospective forests, we defined prospective gradient boosting trees (GBTs) also as an ensemble of trees $h_t^B = \sum_{b=1}^B w_t^b h_t^b(x; \Theta_t^b)$. Unlike the prospective forests defined above, each tree grows independently. In prospective GBTs, the parameters Θ_t^b and the weights w_t^b are iteratively updated by minimizing the empirical risk. This iterative process ensures that each step improves the model by reducing the residual error over the past data $z \leq t$. i.e.

$$\hat{h} = \arg \min_{h^B \in \text{lin}(\mathcal{H}_t^{\text{tree}})} \sum_{t' > 0}^t \sum_{s > t'} w(s - t') \ell(h_{t'}(x_s), y_s), \quad \text{subject to } h^B = \sum_{b=1}^B w_t^b h_t^b(x; \Theta_t^b),$$

where $\text{lin}(\mathcal{H}_t^{\text{tree}})$ is the set of all linear combinations of functions in $\mathcal{H}_t^{\text{tree}}$.

4.2 Preliminary results

We consider data drawn from a periodic process and the Hierarchical Markov Process described in Section 2 and illustrated in Fig. 1. To each data point from these processes, we append two additional noise dimensions sampled from a standard normal distribution, ensuring the presence of both informative and noisy components.

As discussed in Section 2, we implement Prospective-GBTs by giving it $(x_s, \varphi(s))$ as input and training it to predict the label y_s , where we use the time-embedding defined in Eq. (4). In Fig. 5, we compare the prospective risks between several learners including Prospective-GBTs and Prospective-MLP.

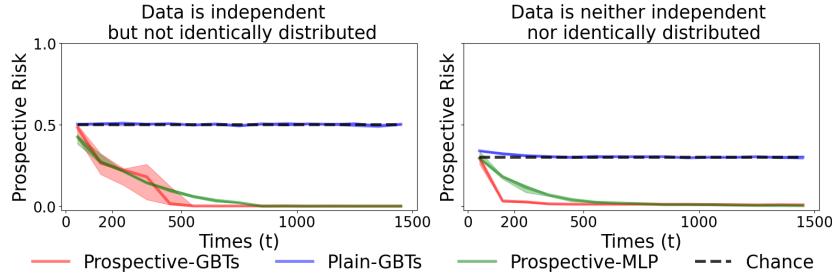


Fig. 5: Prospective risk of Prospective-GBTs (red), Prospective-MLP (green) and Time-agnostic Gradient Boosted trees (Plain-GBTs, blue) across two scenarios where, (1) data is independent but not identically distributed (**Left**), and (2) data is neither independent nor identically distributed (**Right**). In both cases, the risk of Prospective-GBTs and Prospective-MLP approach the Bayes risk, with Prospective-GBTs converging faster. In contrast, the time-agnostic GBTs do not converge consistently. For comparison, the chance prospective risk is 0.5 in the left panel and 0.3 in the right panel.

5 Prospective Foraging

5.1 Motivation and background

Foraging—searching for food, water, and mates—is vital for survival and reproduction, relying on predictions of environmental fluctuations and resource availability. However, standard machine-learning and reinforcement learning (RL) methods—whether minimizing past errors or requiring extensive trial-and-error—are ill-suited to the real-time, single-lifetime risks of foraging. To bridge this gap, we introduce the prospective learning framework in which agents project into possible future states under a one-life constraint. We implement it in a simplified OpenAI Gym foraging scenario [31], compare standard actor-critic RL agents [32, 33] to prospectively augmented versions. Finally, we show that prospective learning framework can be extended beyond the supervised learning problem and that it outperforms an actor critic RL algorithm in a foraging task.

At each discrete time step $t \in \mathbb{N}$ the agent observes $x_t = (s_t, a_{t-19:t}) \in \mathcal{X}$ —its current spatial location and the last 20 actions—and receives a scalar reward $y_t \in \mathcal{Y}$. The only data it may inspect is the trajectory $z_{\leq t} = \{(x_s, y_s)\}_{s=1}^t$. Standard on-policy control maximizes the generalized advantage estimator (GAE) \hat{A}_t^{GAE} . For notational uniformity we instead minimize the loss $\ell(t, \hat{y}_t, y_t) = -\hat{A}_t^{\text{GAE}}$. We define a prospective forager that minimizes the sum of weighted cumulative instantaneous losses on the observed trajectory.

Definition 3 (Prospective Forager). Consider \mathcal{Z} to be a finite family of stochastic processes. Let $\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots \subseteq (\mathcal{Y}^X)^\mathbb{N}$ be an expanding hypothesis class. We employ an actor-critic architecture: the critic evaluates any $h \in \mathcal{H}_t$, whereas the actor can select from $\mathcal{H}_t^{\text{actor}} \subseteq \mathcal{H}_t$. We define Prospective Forager as the learner that minimizes the empirical risk over past data $z_{\leq t}$, i.e:

$$\hat{h} = \arg \min_{h \in \mathcal{H}_t^{\text{actor}}} \max_{u_{it} \leq m \leq t} \frac{1}{m} \sum_{s=1}^m \frac{1}{m-s+1} \sum_{r=s}^m \ell(s, h_s(x_s), y_s) \quad (6)$$

Due to the double summation in Eq. (6), more recent events are weighted more heavily when minimizing the empirical risk.

5.2 Preliminary results

Experiments Setting

An agent forages along a 1×7 linear track that contains two reward patches, A and B, positioned three grid spaces apart (see Fig. 6 (Left)). Reward availability alternates between the two patches every 10 timesteps. Once the reward availability starts, the reward amount decays in an exponential fashion. A reward can be collected at time t only if the agent is at the patch that is available at that moment. The agent moves one grid per timestep, and traveling between A and B takes at least three timesteps. The goal of the agent is to maximize the total amount of reward in its single lifetime, which means that there is no reset in location or time within each run.

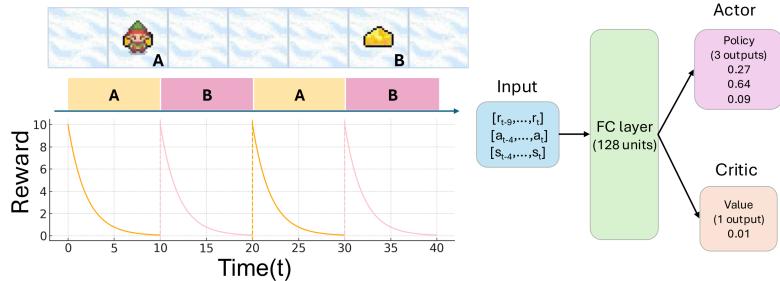


Fig. 6: (Left) Schematics of the foraging task. Agents forage in a 1×7 linear track with two reward patches (A and B), whose reward decays in an exponential fashion. Active patches with reward availability alternates every 10 timesteps. (Right) The actor-critic architecture used for retrospective and prospective agents.

Optimal Solution

Since the reward function is fixed, we can derive an optimal foraging strategy: the agent should leave the current patch (Patch A) before its reward is depleted and arrive at the next patch (Patch B) exactly when its reward peaks. Since no reward is available during travel, an optimal agent accepts zero immediate reward (during travel) over a low immediate reward (by staying in Patch A), in order to maximize future reward. Hence, an agent has to prospect into the future to learn the optimal solution.

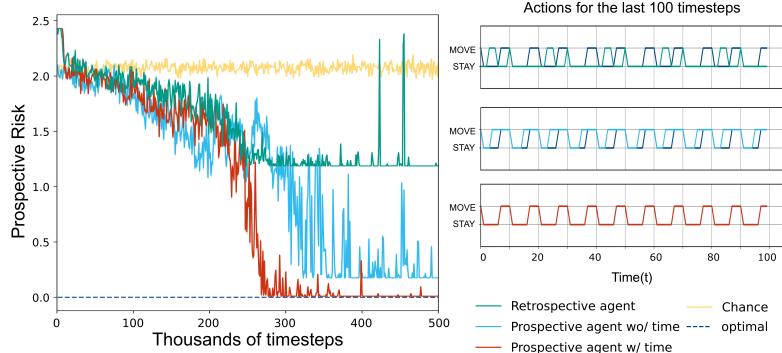


Fig. 7: (Left) Prospective risk of prospective agents (blue and red) and retrospective agents (green) in foraging task, compared to Bayes risk (dotted blue) and chance (yellow) performance. The prospective agent without time converges to a suboptimal risk closer to Bayes risk than retrospective agent, whereas prospective agent with time converges to Bayes risk. (Right) Agent actions plotted for the last 100 timesteps of training. Prospective agent with time embedding (red) shows the same action plan as the optimal agent, prospective agent without time embedding (blue) leaves the patch few steps earlier than optimal, and retrospective agent does not follow the action patterns of the optimal agent.

Preliminary Result

We first implemented a standard actor-critic RL agent in the prospective foraging environment. The model architecture is described in Fig. 6 (Right). We then implemented the prospective forager by integrating prospective risk minimization into the actor-critic agent. Finally, we added time to the prospective forager, using the same time embedding defined by Eq. (4). At each timestep, time embedding is concatenated to the input x_t in addition to agent state and action. Fig. 7 shows that the prospective agent significantly outperforms the standard actor-critic algorithm, and the inclusion of time further improves the performance to near optimal.

6 Conclusion

In this work, we presented preliminary results that extend the prospective learning framework. We began by revisiting the foundational concepts and then analyzed the behavior of deep learning-based prospective learners by experimenting with heterogeneous sampling, two different choices of time-embeddings, and online-training. Next, we introduced Prospective-Trees, a nonparametric alternative that offers competitive performance against Prospective-MLPs. Finally, we proposed prospective foraging, demonstrating the framework’s potential beyond supervised learning settings and highlighting its promise in sequential decision-making tasks. Collectively, these results motivate further mathematical and algorithmic exploration of prospective learning to improve learning in dynamic environments.

Bibliography

- [1] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991.
- [2] Leslie Valiant. *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. Basic Books, June 2013. ISBN 9780465032716.
- [3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.
- [4] Sebastian Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.
- [5] Joshua T Vogelstein, Jayanta Dey, Hayden S Helm, Will LeVine, Ronak D Mehta, Tyler M Tomita, Haoyin Xu, Ali Geisa, Qingyang Wang, Gido M van de Ven, et al. A simple lifelong learning approach. *arXiv preprint arXiv:2004.12908*, 2020.
- [6] Rahul Ramesh and Pratik Chaudhari. Model zoo: A growing" brain" that learns continually. *arXiv preprint arXiv:2106.03027*, 2021.
- [7] Antreas Antoniou, Massimiliano Patacchiola, Mateusz Ochal, and Amos Storkey. Defining benchmarks for continual few-shot learning. *arXiv preprint arXiv:2004.11967*, 2020.
- [8] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.
- [9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.
- [10] Andreas Maurer and Tommi Jaakkola. Algorithmic stability and meta-learning. *Journal of Machine Learning Research*, 6(6), 2005.
- [11] B K Ghosh and P K Sen. *Handbook of Sequential Analysis (Statistics: A Series of Textbooks and Monographs)*. CRC Press, 1 edition, April 1991. ISBN 9780824784089. URL <https://play.google.com/store/books/details?id=JPHWDNSGCyEC>.
- [12] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.
- [13] Fotios Petropoulos, Daniele Apiletti, Vassilios Assimakopoulos, Mohamed Zied Babai, Devon K Barrow, Souhaib Ben Taieb, Christoph Bergmeir, Ricardo J Bessa, Jakub Bijak, John E Boylan, et al. Forecasting: theory and practice. *International Journal of forecasting*, 38(3):705–871, 2022.
- [14] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- [15] Annie Chen, Archit Sharma, Sergey Levine, and Chelsea Finn. You only live once: Single-life reinforcement learning. *Advances in Neural Information Processing Systems*, 35:14784–14797, 2022.
- [16] Saurabh Kumar, Henrik Marklund, Ashish Rao, Yifan Zhu, Hong Jun Jeon, Yueyang Liu, and Benjamin Van Roy. Continual learning as computationally constrained reinforcement learning. *arXiv preprint arXiv:2307.04345*, 2023.

- [17] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.
- [18] Ashwin De Silva, Rahul Ramesh, Carey Priebe, Pratik Chaudhari, and Joshua T Vogelstein. The value of out-of-distribution data. In *International Conference on Machine Learning*, pages 7366–7389. PMLR, 2023.
- [19] Ashwin De Silva, Rahul Ramesh, Rubing Yang, Siyu Yu, Joshua T. Vogelstein, and Pratik Chaudhari. Prospective learning: Learning for a dynamic future. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 123055–123090. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/de85d3cff8512f72fd50b862979f1731-Paper-Conference.pdf.
- [20] Ashwin De Silva, Rahul Ramesh, Lyle Ungar, Marshall Hussain Shuler, Noah J Cowan, Michael Platt, Chen Li, Leyla Isik, Seung-Eon Roh, Adam Charles, et al. Prospective learning: Principled extrapolation to the future. In *Conference on Lifelong Learning Agents*, pages 347–357. PMLR, 2023.
- [21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [22] Chen Zeno, Itay Golan, Elad Hoffer, and Daniel Soudry. Task-agnostic continual learning using online variational bayes with fixed-point updates. *Neural Computation*, 33(11):3139–3177, 2021.
- [23] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15(90):3133–3181, 2014. ISSN 1533-7928, 1533-7928. <https://doi.org/10.5555/2627435.2697065>. URL <https://jmlr.org/papers/v15/delgado14a.html>.
- [24] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/pdf?id=Fp7_phQszn.
- [25] Gérard Biau, Luc Devroye, and Gábor Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(66):2015–2033, 2008. URL <http://jmlr.org/papers/v9/biau08a.html>.
- [26] Erwan Scornet, Gérard Biau, and Jean-Philippe Vert. Consistency of random forests. *The Annals of Statistics*, 43(4), August 2015. ISSN 0090-5364. <https://doi.org/10.1214/15-aos1321>. URL <http://dx.doi.org/10.1214/15-AOS1321>.
- [27] Jason M. Klusowski and Peter M. Tian. Large scale prediction with decision trees, 2023. URL <https://arxiv.org/abs/2104.13881>.
- [28] Leo Breiman, Jerome Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.
- [29] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
- [30] Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. The elements of statistical learning, 2009.

- [31] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016. URL <https://arxiv.org/abs/1606.01540>.
- [32] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.
- [33] Sergey Shuvaev, Sarah Starosta, Duda Kvitsiani, Adam Kepcs, and Alexei Koulakov. R-learning in actor-critic model offers a biologically relevant mechanism for sequential decision-making. *Advances in neural information processing systems*, 33:18872–18882, 2020.