ReVar: Strengthening Policy Evaluation via Reduced Variance Sampling

Subhojyoti Mukherjee¹  Josiah P. Hanna²  Robert Nowak¹

¹Department of Electrical and Computer Engineering, University of Wisconsin-Madison, USA
²Computer Sciences Department, University of Wisconsin-Madison, USA

Abstract

This paper studies the problem of data collection for policy evaluation in Markov decision processes (MDPs). In policy evaluation, we are given a target policy and asked to estimate the expected cumulative reward it will obtain in an environment formalized as an MDP. We develop theory for optimal data collection within the class of tree-structured MDPs by first deriving an oracle data collection strategy that uses knowledge of the variance of the reward distributions. We then introduce the Reduced Variance Sampling (ReVar) algorithm that approximates the oracle strategy when the reward variances are unknown a priori and bound its sub-optimality compared to the oracle strategy. Finally, we empirically validate that ReVar leads to policy evaluation with mean squared error comparable to the oracle strategy and significantly lower than simply running the target policy.

1 INTRODUCTION

In reinforcement learning (RL) applications, there is often a need for policy evaluation to determine (or estimate) the expected return (future cumulative reward) of a given policy. Policy evaluation is also required in other sequential decision-making settings outside of RL. For example, testing an autonomous vehicle stack or ad-serving system can be seen as policy evaluation applications. Accurate and data efficient policy evaluation is critical for safe and trust-worthy deployment of autonomous systems.

This paper studies data collection for low mean squared error (MSE) policy evaluation in sequential decision-making tasks formalized as Markov decision processes (MDPs). The objective of policy evaluation is to estimate the expected return that will be obtained by running a target policy which is a given probabilistic mapping from states to actions.

To evaluate the target policy, we require data from the environment in which it will be deployed. Collecting data requires running a (possibly non-stationary) behavior policy to generate state-action-reward trajectories. Our goal is to find a behavior policy that leads to a minimum MSE evaluation of the target policy.

The most natural choice is on-policy sampling in which we use the target policy as the behavior policy. However, we show that in some cases this choice is far from optimal (e.g., Figure 2 in our empirical analysis) as it fails to actively take actions from which the expected return is uncertain. Instead, an optimal behavior policy should take actions in any given state to reduce uncertainty in the current estimate of the expected return from that state.

Our paper makes the following main contributions. We first derive an optimal “oracle” behavior policy for finite tree-structured MDPs assuming oracle access to the MDP transition probabilities and variances of the reward distributions. Sampling trajectories according to the oracle behavior policy minimizes the MSE of the estimator of the target policy’s expected. As a special case (depth 1 tree MDPs), we recover the optimal behavior policy for multi-armed bandits [Carpentier et al., 2015].

We then introduce a practical algorithm, Reduced Variance Sampling (ReVar), that adaptively learns the optimal behavior policy by observing rewards and adjusting the policy to select actions that reduce the MSE of the estimator. The main idea of ReVar is to plug-in upper-confidence bounds on the reward distribution variances to approximate the oracle behavior policy. We define a notion of policy evaluation regret compared to the oracle behavior policy, and bound the regret of ReVar. The regret converges rapidly to 0 as the number of sampled episodes grows, theoretically guaranteeing that ReVar quickly matches the performance of the oracle policy. Finally, we implement ReVar and show it leads to low MSE policy evaluation in both a tree-structured and a general finite-horizon MDP. Taken together, our contributions provide a theoretical foundation towards optimal
data collection for policy evaluation in MDPs.

The remainder of the paper is organized as follows. In Section 3 we reformulate our problem in the bandit setting and discuss related bandit works. In Section 4 we extend the bandit formulation to the tree MDP. Finally, we introduce the more general Directed Acyclic Graph (DAG) MDP in Section 5 and discuss some limitations of our sampling behavior. We show numerical experiments in Section 6 and conclude in Section 7.

2 BACKGROUND

In this section, we introduce notation, define the policy evaluation problem, and discuss the prior literature.

2.1 NOTATION

A finite-horizon Markov Decision Process, M, is the tuple \((S, A, P, R, \gamma, d_0, L)\), where \(S\) is a finite set of states, \(A\) is a finite set of actions, \(P : S \times A \times S \rightarrow [0, 1]\) is a state transition function, \(R\) is the reward distribution (formalized below), \(\gamma \in [0, 1]\) is the discount factor, \(d_0\) is the starting state distribution, and \(L\) is the maximum episode length. A (stationary) policy, \(\pi : S \times A \rightarrow [0, 1]\), is a probability distribution over actions conditioned on a given state. We assume data can only be collected through episodic interaction: an agent begins in state \(S_0 \sim d_0\) and then at each step \(t\) takes an action \(A_t \sim \pi(\cdot|S_t)\) and proceeds to state \(S_{t+1} \sim P(\cdot|S_t, A_t)\). Interaction terminates in at most \(L\) steps. Each time the agent takes action \(a_t\) in state \(s_t\) it observes a reward \(R_t \sim R(s_t, a_t)\).

We assume \(R(s, a) = \mathcal{P}(\mu(s, a), \sigma^2(s, a))\), where \(\mathcal{P}\) denotes a parametric distribution with mean \(\mu(s, a)\) and variance \(\sigma^2(s, a)\). The entire interaction produces a trajectory \(H := \{(S_t, A_t, R_t)\}_{t=1}^{L}\). We assume \(d_0\) is known but \(P\) and the reward distributions are unknown. We define the value of a policy as: \(v(\pi) := \mathbb{E}_\pi[\sum_{t=1}^{L} \gamma^{t-1} R_t]\), where \(\mathbb{E}_\pi\) is the expectation w.r.t. trajectories sampled by following \(\pi\).

We will make use of the fact that the value of a policy can be written as: \(v(\pi) = \mathbb{E}[v^\pi(S_0)|S_0 \sim d_0]\), where:

\[
v^\pi_t(s) := \sum_a \pi(a|s) \mu(s, a) + \gamma \sum_{s'} P(s'|s, a) v^\pi_{t+1}(s')
\]

for \(t \leq L\) and \(v^\pi_T(s) = 0\) for \(t > L\).

2.2 POLICY EVALUATION

We now formally define our objective. We are given a target policy, \(\pi\), for which we want to estimate \(v(\pi)\). To estimate \(v(\pi)\) we will generate a set of \(K\) trajectories where each trajectory is generated by following some policy. Let \(H^k := \{s^k_t, a^k_t, R^k_t(s^k_t, a^k_t)\}_{t=1}^{L}\) be the trajectory collected in episode \(k\) and let \(b^k\) be the policy ran to produce \(H^k\). The entire set of collected data is given as \(D := \{H^k, b^k\}_{k=1}^{K}\).

Once \(D\) is collected, we estimate \(v(\pi)\) with a certainty-equivalence estimate [Sutton 1988]. Suppose \(D\) consists of \(n = KL\) state-action transitions. We define the random variable representing the estimated future reward from state \(s\) at time-step \(t\) as:

\[
Y_n(s, t) := \sum_a \pi(a|s) \hat{\mu}(s, a) + \gamma \sum_{s'} \hat{P}(s'|s, a) Y_n(s', t+1),
\]

where \(Y_n(s, t) := 0\) if \(t \geq L\), \(\hat{\mu}(s, a)\) is an estimate of \(\mu(s, a)\) and \(\hat{P}(s'|s, a)\) is an estimate of \(P(s'|s, a)\), both computed from \(D\). Finally, the estimate of \(v(\pi)\) is computed as \(Y_n := \sum_s d_0(s) Y_n(s, 0)\). In the policy evaluation literature, the certainty-equivalence estimator is also known as the direct method [Jiang and Li 2016] and, in tabular settings, can be shown to be equivalent to batch temporal-difference estimators [Sutton 1988, Pavse et al. 2020]. Thus, it is representative of two of policy evaluation estimators that often give strong empirical performance [Voloshin et al. 2019].

Our objective is to determine the sequence of behavior policies that minimize error in estimation of \(v(\pi)\). Formally, we seek to minimize mean squared error which is defined as: \(\mathbb{E}_D[\{Y_n - v(\pi)\}^2]\) where the expectation is over the collected data set \(D\).

2.3 RELATED WORK

Our paper builds upon work in the bandit literature for optimal data collection for estimating a weighted sum of the mean reward associated with each arm [Antos et al. 2008] study estimating the mean reward of each arm equally well and show that the optimal solution is to pull each arm proportional to the variance of its reward distribution. Since the variances are unknown a priori, they introduce an algorithm that pulls arms in proportion to the empirical variance of each reward distribution [Carpentier et al. 2015] extend this work by introducing a weighting on each arm that is equivalent to the target policy action probabilities in our work. They show that the optimal solution is then to pull each arm proportional to the product of the standard deviation of the reward distribution and the arm weighting. Instead of using the empirical standard deviations, they introduce an upper confidence bound on the standard deviation and use it to select actions. Our work is different from these earlier works in that we consider more general tree-structured MDPs of which bandits are a special case.

In RL and MDPs, exploration is widely studied with the objective of finding the optimal policy. Prior work attempts to balance exploration to reduce uncertainty with exploitation to converge to the optimal policy. Common approaches are based on reducing uncertainty [Osband et al. 2016].
O’Donoghue et al. [2018] or incentivizing visitation of novel states [Barto, 2013; Pathak et al., 2017; Burda et al., 2018]. These works differ from our work in that we focus on evaluating a fixed policy rather than finding the optimal policy. In our problem, the trade-off becomes balancing taking actions to reduce uncertainty with taking actions that the target policy is likely to take.

Our work is similar in spirit to work on adaptive importance sampling [Rubinstein and Kroese, 2013] which aims to lower the variance of Monte Carlo estimators by adapting the data collection distribution. Adaptive importance sampling was used by Hanna et al. [2017] to lower the variance of policy evaluation in MDPs. It has also been used to lower the variance of policy gradient RL algorithms [Bouchard et al., 2016; Ciosek and Whiteson, 2017]. AIS methods attempt to find a single optimal sampling distribution whereas our approach attempts to reduce uncertainty in the estimated mean rewards. In a similar spirit, Talebi and Maillard [2019] adapt the behavior policy to minimize error in estimating the transition model $P$.

3 OPTIMAL DATA COLLECTION IN MULTI-ARMED BANDITS

Before we address optimal data collection for policy evaluation in MDPs, we first revisit the problem in the bandit setting as addressed by earlier work [Carpentier et al., 2015]. The bandit setting provides intuition for how a good data collection strategy should select actions, though it falls short of an entire solution for MDPs.

Observe that the policy value in a bandit problem is defined as $v(\pi) := \sum_{a=1}^{A} \pi(a) \mu(a)$ where the bandit consists of a single state $s$ and $A$ actions indexed as $a = 1, 2, \ldots, A$. In this setting, the horizon $L = 1$ so we return to the same state after taking an action $a$ at time $t$. Hence, we drop the state $s$ from our standard notation.

Suppose we have a budget of $n$ samples to divide between the arms and let $T_n(a)$ be the number of samples allocated to actions $1, 2, \ldots, A$ at the end of $n$ rounds. We define the estimate:

$$Y_n := \sum_{a=1}^{A} \pi(a) \frac{T_n(a)}{T_n(a)} R_h(a) = \sum_{a=1}^{A} \pi(a) \bar{\mu}(a).$$  \hspace{1cm} (1)

where, $R_h(a)$ is the $h$th reward received after taking action $a$. Note that, once all actions where $\pi(a) > 0$ have been tried, $Y_n$ is an unbiased estimator of $v(\pi)$ since $\bar{\mu}(a)$ is an unbiased estimator of $\mu(a)$. Thus, reducing MSE requires allocating the $n$ samples to reduce variance. As shown by Carpentier et al. [2015], the minimal-variance allocation is given by pulling each arm with the proportion $b^*(a) \propto \pi(a) \sigma(a)$.

4 OPTIMAL DATA COLLECTION IN TREE MDPs

In this section, we derive the optimal action proportions for tree-structured MDPs assuming the variances of the reward distributions are known, introduce an algorithm that approximates the optimal allocation when the variances are unknown, and bound the finite-sample MSE of this algorithm. Tree MDPs are a straightforward extension of the multi-armed bandit model to capture the fact that the optimal allocation for each action in a given state must consider the future states that could arise from taking that action.

We first define a discrete tree MDP as follows:

**Definition 1. (Tree MDP)** An MDP is a discrete tree MDP
We first consider an oracle data collection strategy which
T ⊂ M (see Figure[1]) if the following holds:

(1) There are L levels indexed by ℓ where ℓ = 1, 2, . . . , L.
(2) Every state is represented as sℓi where ℓ is the level of the
state s indexed by i.
(3) The transition probabilities are such that one can only
transition from a state in level ℓ to one in level ℓ + 1 and each
non-initial state can only be reached through one state
and only one action in that state. Formally, ∀s′, P(s′|s, a) ̸= 0
for only one state-action pair s, a and if s′ is in level ℓ + 1
then s is in level ℓ. Finally, P(sL′|sL′i, a) = 0, ∀a.
(4) For simplicity, we assume that there is a single starting
state s1 (called the root). It is easy to extend our results to
multiple starting states with a starting state distribution, d0,
by assuming that there is only one action available in the
root that leads to each possible start state, s, with probability
d0(s). The leaf states are denoted as sℓ1.
(5) The interaction stops after L steps in state sℓ1 after taking
an action a and observing the reward Rℓ1(sℓ1, a).

Note that, because we assume a single initial state, s1, we
have that estimating v(π) is equivalent to estimating v(s1).
A similar Tree MDP model has been previously used in theo-
retical analysis by Jiang and Li [2016]; our model is slightly
more general as we consider per-step stochastic rewards whereas
Jiang and Li [2016] only consider deterministic rewards at the end of trajectories.

4.1 ORACLE DATA COLLECTION

We first consider an oracle data collection strategy which
knows the variance of all reward distributions and knows the
state transition probabilities. After observing n state-action-
reward tuples, the oracle computes the following estimate
of νn(s1) (or equivalently v(π)):

\[
\begin{align*}
Y_n(s_1) := \sum_{a=1}^{A} \pi(a|s_1) \left( \frac{1}{T_n(s_1)} \sum_{b=1}^{T_n(s_1)} R_b(s_1, a) \right) \\
+ \gamma \sum_{s_{j+1}^{(1)}} P(s_{j+1}^{(1)}|s_1, a) Y_n(s_{j+1}^{(1)}) \\
= \sum_{a=1}^{A} \pi(a|s_1) \left( \bar{R}(s_1, a) + \gamma \sum_{s_{j+1}^{(1)}} P(s_{j+1}^{(1)}|s_1, a) Y_n(s_{j+1}^{(1)}) \right)
\end{align*}
\]

(2)

where Tn(s, a) denotes the number of times that the oracle
took action a in state s. Note that in Section 2.2 we define
Yn(s, t) but now we use Yn(s) as timestep is implicit in
the layer of the tree. Also [2] differs from the estimator defined
in Section 2.2 as it uses the true transition probabilities, P,
instead of their empirical estimate, ̂P. The MSE of Yn is:

\[
\begin{align*}
\mathbb{E}_D[(Y_n(s_1) - v^n(s_1))^2] \\
= \text{Var}(Y_n(s_1)) + \text{bias}^2(Y_n(s_1)).
\end{align*}
\]

(3)

The bias of this estimator becomes zero once all (s, a)-pairs
with π(a|s) > 0 have been visited a single time, thus we
focus on reducing Var(Yn(s1)). Before defining the oracle
data collection strategy, we first state an assumption on D.

Assumption 1. The data D collected over n state-action-
reward samples has at least one observation of each state-
action pair, (s, a), for which π(a|s) > 0.

Assumption[1] ensures that Yn is an unbiased estimator of
v(π) so that reducing MSE is equivalent to reducing variance.
Before stating our main result, we provide intuition with a lemma that gives the optimal proportion for each
action in a 2-depth tree.

Lemma 1. Let T be a 2-depth stochastic tree MDP as
defined in Definition[7] (see Figure[7] in Appendix B). Let
Yn(s1) be the estimated return of the starting state s1 after
observing n state-action-reward samples. Note that v^n(s1)
is the expectation of Yn(s1) under Assumption[7]. Let D be
the observed data over n state-action-reward samples. Min-
imal MSE, \mathbb{E}_D[(Y_n(s_1) - v^n(s_1))^2], is obtained by taking
actions in each state in the following proportions:

\[
b^*(a|s_1) \propto \pi(a|s_1) \sigma^2(s_1, a)
\]

\[
b^*(a|s_1) \propto \pi^2(a|s_1) \left( \sigma^2(s_1, a) + \gamma^2 \sum_{s_2^1} P(s_2^1|s_1, a) B^2(s_2^1) \right),
\]

where, B(s2^1) = \sum_a \pi(a|s_1) \sigma^2(s_1, a).

Proof (Overview): We decompose the MSE into its vari-
ance and bias terms and show that Yn is unbiased under
Assumption[1]. Next note that the reward in the next state
is conditionally independent of the reward in the current
state given the current state and action. Hence we can write
the variance in terms of the variance of the estimate in the
initial state and the variance of the estimate in the final layer.
We then rewrite the total samples of a state-action pair i.e
Tn(s1, a) in terms of the proportion of the number of times
the action was sampled in the state i.e b(a|s1). To do so, we
take into account the tree structure to derive the expected
proportion of times that action a is taken in each state in
layer 2 as follows:

\[
b(a|s_2^1) = \frac{T_n(s_2^1)}{\sum_{a'} T_n(s_2^1, a')} \left( \frac{T_n(s_1)}{T_n(s_2^1, a) + \gamma^2 \sum_{s_2^1} P(s_2^1|s_1, a) T_n(s_2^1, a)/n} \right)
\]

where in (a) the action a is used to transition to state s2^1
from s1 and so \sum_a Tn(s2^1, a) = P(s2^1|s1, a)Tn(s1, a). We
next substitute the b(a|s1) for each state-action pair into
the variance expression and determine the \( b \) values that minimize the expression subject to \( \forall s, \sum_i b(a(s)) = 1 \) and \( \forall s, b(a(s)) > 0 \). The full proof is given in Appendix [B].

Note that the optimal proportion in the leaf states, \( b^*(a|s^2_1) \), is the same as in [Carpenter and Munos 2011] (see Proposition [1]) as terminal states can be treated as bandits in which actions do not affect subsequent states. The key difference in the root state, \( s^1_1 \), where the optimal action proportion, \( b^*(a|s^1_1) \) depends on the expected leaf state normalization factor \( B(s^2_1) \), where \( s^2_1 \) is a state sampled from \( P(\cdot|s^1_1, a) \). The normalization factor, \( B(s^2_1) \), captures the total contribution of state \( s^2_1 \) to the variance of \( Y_n \), and thus actions in the root state must be chosen to 1) reduce variance in the immediate reward estimate and to 2) get to states that contribute more to the variance of the estimate. We explore the implications of the oracle action proportions in Lemma [1] with the following two examples.

**Example 1. (Child Variance matters)** Consider a 2-depth, 2-action tree MDP \( T \) with deterministic \( P \), i.e., \( P(s^2_1|s^1_1, 2) = P(s^2_1|s^1_1, 1) = 1 = \gamma \) (see Figure [2] (Left) in Appendix [C]). Suppose the target policy is the uniform distribution \( \pi(s) = \frac{1}{2} \) in all states. The reward distribution variances are given by \( \sigma^2(s^1_1) = 400, \sigma^2(s^2_1, 1) = 400, \sigma^2(s^2_1, 2) = 400, \sigma^2(s^2_2, 1) = 4, \) and \( \sigma^2(s^2_2, 2) = 4 \). So the right sub-tree at \( s^1_1 \) has higher variance (larger \( B \)-value) than the left sub-tree. Following the sampling rule in Lemma [1], we can show that \( b^*(1|s^1_1) > b^*(2|s^1_1) \) (the full calculation is given in Appendix [C]). Hence the right sub-tree with higher variance will have a higher proportion of pulls which allows the oracle to get to the high variance state \( s^1_1 \). Observe that taking action 2 leads to state \( s^2_2 \) which contributes much less to the total variance. Thus, this example highlights the need to consider the variance of subsequent states.

**Example 2. (Transition Model matters)** Consider a 2-depth, 2-action tree MDP \( T \) in which we have \( P(s^2_2|s^1_1, 1) = p, P(s^2_1|s^1_1, 1) = 1 - p, P(s^2_2|s^1_1, 2) = p, \) and \( P(s^2_1|s^1_1, 2) = 1 - p \). This example is shown in Figure [2] (Right) in Appendix [C]. Following the result of Lemma [1] if \( p \gg (1 - p) \) it can be shown that the variances of the states \( s^1_1 \) and \( s^2_1 \) have greater importance in calculating the optimal sampling proportions of \( s^1_1 \). The calculation is shown in Appendix [D]. Thus, less likely future states have less importance for computing the optimal sampling proportion in a given state.

Having developed intuition for minimal-variance action selection in a 2-depth tree MDP, we now give our main result that extends Lemma [1] to an \( L \)-depth tree.

**Theorem 1.** Assume the underlying MDP is an \( L \)-depth tree MDP as defined in Definition [7]. Let the estimated return of the starting state \( s^1_1 \) after \( n \) state-action-reward samples be defined as \( Y_n(s^1_1) \). Note that the \( \nu^s(s^1_1) \) is the expectation of \( Y_n(s^1_1) \) under Assumption [7]. Let \( D \) be the observed data over \( n \) state-action-reward samples. To minimize MSE \( \mathbb{E}_D[|Y_n(s^1_1) - \mu(Y_n(s^1_1))|^2] \) the optimal sampling proportions for any arbitrary state is given by:

\[
b^*(a|s^1_1) \propto \pi^2(s^1_1) \left[ \sigma^2(s^1_1, a) + \gamma^2 \sum_{j = 1}^L P(s^j_1|s^1_1, a) B^2(s^j_1) \right],
\]

where, \( B(s^j_1) \) is the normalization factor defined as follows:

\[
B(s^j_1) = \sum_a \pi^2(s^j_1) \left( \sigma^2(s^j_1, a) + \gamma^2 \sum_{j = 1} P(s^j_{j+1}|s^j_1, a) B^2(s^j_{j+1}) \right)
\]

**Proof (Overview):** We prove Theorem [1] by induction. Lemma [1] proves the base case of estimating the sampling proportion for level \( L = 1 \) and \( L \). Then, for the induction step, we assume that all the sampling proportions from level \( L \) till some arbitrary level \( \ell + 1 \) can be subsequently built up using dynamic programming starting from level \( L \). For states in level \( L \) to the states in level \( \ell + 1 \) we can compute \( b^*(a|s^1_{\ell+1}) \) by repeatedly applying Lemma [1]. Then we show that at the level \( \ell \) we get a similar recursive sampling proportion as stated in the theorem statement. The proof is given in Appendix [E].

4.2 MSE OF THE ORACLE

In this subsection, we derive the MSE that the oracle will incur when matching the action proportions given by Theorem [1]. The oracle is run for \( K \) episodes where each episode consist of \( L \) length trajectory of visiting state-action pairs. So the total budget is \( n = KL \). At the end of the \( K \)-th episode the MSE of the oracle is estimated which is shown in Proposition [2]. Before stating the proposition we introduce additional notation which we will use throughout the remainder of the paper. Let

\[
T^k_t(s, a) = \sum_{i=0}^{k-1} \mathbb{1}\{\{s^i_1, a^i_t\} = (s, a)\}, \forall t, s, a
\]

(5)
denote the total number of times that \( (s, a) \) has been observed in \( D \) (across all trajectories) up to time \( t \) in episode \( k \) and \( \mathbb{1}\{\cdot\} \) is the indicator function. Similarly let

\[
T^k_t(s, a, s') = \sum_{i=0}^{k-1} \mathbb{1}\{(s^i_1, a^i_t, s^i_{t+1}) = (s, a, s')\}, \forall t, s, a, s'
\]

(6)
denote the number of times action \( a \) is taken in \( s \) to transition to \( s' \). Finally we define the state sample \( T^k_t(s) = \sum_a T^k_t(s, a) \) as the total number of times any state is visited and an action is taken in that state.
Proposition 2. Let there be an oracle which knows the state-action variances and transition probabilities of the L-depth tree MDP \( \mathcal{T} \). Let the oracle take actions in the proportions given by Theorem 1. Let \( \mathcal{D} \) be the observed data over \( n \) state-action-reward samples such that \( n = KL \). Then the oracle suffers an MSE of

\[
\text{L}_n = \sum_{t=1}^{L} \left[ \frac{B^2(s_i^t)}{T_{L,K}^{*,K}(s_i^t)} \right] + \gamma^2 \sum_{a} \pi^2(a|s_i^t) \left[ \sum_{s_i^{t+1}} P(s_i^{t+1}|s_i^t, a) \frac{B^2(s_i^{t+1})}{T_{L,K}^{*,K}(s_i^{t+1})} \right].
\]

(7)

where, \( T_{L,K}^{*,K}(s_i^t) \) denotes the optimal state samples of the oracle at the end of episode \( K \).

The proof is given in Appendix B. From Proposition 2, we see that the MSE of the oracle goes to 0 as the number of episodes \( K \to \infty \), and \( T_{L,K}^{*,K}(s_i^t) \to \infty \) simultaneously for all \( s_i^t \in S \). Observe that if for every state \( s \) the total state counts \( T_{L,K}^{*,K}(s) = cn \) for some constant \( c > 0 \) then the loss of the oracle goes to 0 at the rate \( O(1/n) \).

4.3 REDUCED VARIANCE SAMPLING

The oracle data collection strategy provides intuition for optimal data collection for minimal-variance policy evaluation, however, it is not a practical strategy itself as it requires \( \sigma \) and \( P \) to be known. We now introduce a practical data collection algorithm – Reduced Variance Sampling (ReVar) – that is agnostic to \( \sigma \) and \( P \). Our algorithm follows the proportions given by Theorem 1 with the true reward variances replaced with an upper confidence bound and the true transition probabilities replaced with empirical frequencies. Formally, we define the desired proportion for action \( a \) in state \( s_i^t \) after \( t \) steps as \( \hat{b}_i^{k+1}(a|s_i^t) \)

\[
\hat{b}_i^{k+1}(a|s_i^t) = \frac{\bar{P}_i^{k}(s_i^{t+1}|s_i^t, a) B^2(s_i^{t+1})}{T_{L,K}^{*,K}(s_i^{t+1})}.
\]

(10)

where, \( T_{L}^{k}(s_i^t, a, s_i^{t+1}) \) is defined in (6). Further in (8), \( \hat{b}_i^{k+1}(s_i^t, a) \) is the plug-in estimate of \( B(s_i^{t+1}) \). Observe that for all of these plug-in estimates we use all the past history till time \( t \) in episode \( k \) to estimate these statistics.

Eq. (8) allows us to estimate the optimal proportion for all actions in any state. To match these proportions, rather than sampling from \( \hat{b}_i^{k+1}(a|s_i^t) \), ReVar takes action \( I_{t+1}^k \) at time \( t+1 \) in episode \( k \) according to:

\[
I_{t+1}^k = \arg \max_a \left\{ \hat{b}_i^k(a|s_i^t) \right\}.
\]

(11)

This action selection rule ensures that the ratio \( \hat{b}_i^k(a|s_i^t) / T_i(s_i^t, a) \approx 1 \). It is a deterministic action selection rule and thus avoids variance due to simply sampling from the estimated optimal proportions. Note that in the terminal states, \( s_i^T \), the sampling rule becomes

\[
I_{t+1}^k = \arg \max_a \left\{ \frac{\pi(a|s_i^t)\sigma_i^k(a|s_i^t)}{T_{L,K}^{*,K}(s_i^t, a)} \right\}
\]

which matches the bandit sampling rule of Carpenter and Munos (2011, 2012).

We give pseudocode for ReVar in Algorithm 1. The algorithm proceeds in episodes. In each episode we generate a trajectory from the starting state \( s_0^1 \) (root) to one of the terminal state \( s_i^L \) (leaf). At episode \( k \) and time-step \( t \) in some arbitrary state \( s_i^t \) the next action \( I_{t+1}^k \) is chosen based on (11). The trajectory generated is added to the dataset \( \mathcal{D} \). At the end of the episode we update the model parameters, i.e., we estimate the \( \hat{\sigma}_i^{k}(a|s_i^t) \) and \( \hat{\bar{P}}_i^{k}(s_i^{t+1}|s_i^t, a) \) for each state-action pair. Finally, we update \( \hat{b}_i^{k+1}(a|s_i^t) \) for the next episode using eq. (8).

Algorithm 1 Reduced Variance Sampling (ReVar)

1: Input: Number of trajectories to collect, \( K \).
2: Output: Dataset \( \mathcal{D} \).
3: Initialize \( \mathcal{D} = \emptyset, \hat{\bar{P}}_i^0(a|s_i^t) \) uniform over all actions in each state.
4: for \( k = 0, 1, \ldots, K \) do
5: Generate trajectory \( H^k := \{S_t, I_t, R(I_t)\}_{t=1}^L \) by selecting \( I_t \) according to (11).
6: \( \mathcal{D} \leftarrow \mathcal{D} \cup \{(H^k, \hat{\bar{P}}_i^k)\} \)
7: Update model parameters and estimate \( \hat{b}_i^{k+1}(a|s_i^t) \) for each \( (s_i^t, a) \).
8: Update \( \hat{\sigma}_i^{k+1}(a|s_i^t) \) from level \( L \) to 1 following (8).
9: Return Dataset \( \mathcal{D} \) to evaluate policy \( \pi \).
4.4 REGRET ANALYSIS

We now theoretically analyze ReVar by bounding its regret with respect to the oracle behavior policy. We analyze ReVar under the assumption that $P$ is known and so we are only concerned with obtaining accurate estimates of the reward means and variances. This assumption is only made for the regret analysis and is not a fundamental requirement of ReVar. Though somewhat restrictive, the case of known state transitions is still interesting as it arises in practice when state transitions are deterministic or we can estimate $P$ much easier than we can estimate the reward means.

We first define the notion of regret of an algorithm compared to the oracle MSE $\mathcal{L}_n$ in (7) as follows:

$$ R_n = \mathcal{L}_n - \mathcal{L}_n $$

where, $n$ is the total budget, and $\mathcal{L}_n$ is the MSE at the end of episode $K$ following the sampling rule in (8). We make the following assumption that rewards are bounded:

Assumption 2. The reward from any state-action pair has bounded range, i.e., $R(s, a) \in [-\eta, \eta]$ almost surely at every time-step $t$ for some fixed $\eta > 0$.

Note that this is a common assumption in the RL literature [Munos 2005, Agarwal et al., 2019]. The reward can also be multi-modal as long as it is bounded. Then the regret of ReVar over a $L$-depth deterministic tree is given by the following theorem.

Theorem 2. Let the total budget be $n = KL$ and $n \geq 4SA$. Then the total regret in a deterministic $L$-depth tree at the end of $K$-th episode when taking actions according to (8) is given by

$$ R_n \leq \left( \frac{B^2 \sqrt{\log(SAn^{1/2})}}{n^{3/2} b_{\min}^{3/2}(s)} \right) + \left( \frac{B^2 \sqrt{\log(SAn^{1/2})}}{n^{3/2} b_{\min}^{3/2}(s)} \right) $$

where, the $O$ hides other lower order terms and $B_{s_i}$ is defined in (4) and $b_{\min}^{3/2}(s) = \min_a b^{3/2}(a|s)$.

Note that if $L = 1$, $|S| = 1$, we recover the bandit setting and our regret bound matches the bound in Carpentier and Munos [2011]. Note that MSE using data generated by any policy decays at a rate no faster than $O(n^{-1})$, the parametric rate. The key feature of ReVar is that it converges to the oracle policy. This means that asymptotically, the MSE based on ReVar will match that of the oracle. Theorem 2 shows that the regret scales like $O(n^{-3/2})$ if we have the $b_{\min}^{3/2}(s)$ over all states $s \in S$ as some reasonable constant $O(1)$. In contrast, suppose we sample trajectories from a suboptimal policy, i.e., a policy that produces an MSE worse than that of the oracle for every $n$. This MSE gap never diminishes, so the regret cannot decrease at a rate faster than the oracle rate of $O(n^{-1})$. Finally, note that the regret bound in Theorem 2 is a problem dependent bound as it involves the parameter $b_{\min}^{3/2}(s)$.

Proof (Overview): We decompose the proof into several steps. We define the good event $\xi_k$ based on the state-action-reward samples $D$ that holds for all episode $k$ and time $t$ such that $|\hat{\sigma}^2_k(s, a) - \sigma^2(s, a)| \leq \epsilon$ for some $\epsilon > 0$ with probability $1 - \delta$ made explicit in Corollary 1. Now observe that MSE of ReVar is

$$ \mathcal{L}_n = \mathbb{E}[D] (Y_n(s_1^1) - v^*(s_1^1))^2 $$

where, the $\mathbb{E}[D]$ is known and so we are only interested in bounding the first term. For the second term we set the MSE at the end of $n$-depth deterministic tree is given by the following assumption:

$$ \mathcal{L}_n = \mathbb{E}[D] (Y_n(s_1^1) - v^*(s_1^1))^2 $$

Note that here we are considering a known transition function $P$. The first term in (12) can be bounded using

$$ \mathbb{E}[D] (Y_n(s_1^1) - v^*(s_1^1))^2 = \mathbb{E}[D] (Y_n(s_1^1) - v^*(s_1^1))^2 $$

where, $T^{(2)}, k(s_1^1, a)$ is a lower bound to $T^{(2)}, k(s_1^1, a)$ made explicit in Lemma 6 and $T^{(2)}, k(s_1^1, a)$ is a lower bound to $T^{(2)}, k(s_1^1, a)$ made explicit in Lemma 5. We can combine these two lower bounds and give an upper bound to MSE in a two depth tree which is shown Lemma 7. Finally, for the $L$ depth stochastic tree we can repeatedly apply Lemma 7 to bound the first term. For the second term we set the $\delta = n^{-2}$ and use the boundedness assumption in Assumption 2 to get the final bound. The proof is given in Appendix H.

5 OPTIMAL DATA COLLECTION BEYOND TREES

The tree-MDP model considered above allows us to develop a foundation for minimal-variance data collection in decision problems where actions at one state affect subsequent states. One limitation of this model is that, for any non-initial state, $s_i^t$, there is only a single state-action path that could have been taken to reach it. In a more general finite-horizon MDP, there could be many different paths to reach the same non-initial state. Unfortunately, the existence of multiple paths to a state introduces cyclical dependencies.
between states that complicate derivation of the minimal-variance data collection strategy and regret analysis. In this section, we elucidate this difficulty by considering the class of directed acyclic graph (DAG) MDPs.

In this section we first define a DAG $G \subset M$. An illustrative figure of a 3-depth 2-action DAG is in Figure 5 of Appendix I.

**Definition 2. (DAG MDP)** A DAG MDP follows the same definition as the tree MDP in Definition 1 except $P(s'|s, a)$ can be non-zero for any $s$ in layer $\ell$, $s'$ in layer $\ell+1$, and any $a$, i.e., one can now reach $s'$ through multiple previous state-action pairs.

**Proposition 3.** Let $G$ be a 3-depth, $A$-action DAG defined in Definition 2. The minimal-MSE sampling proportions $b^*(a|s_1^2), b^*(a|s_2^2)$ depend on themselves such that $b(a|s_1^2) \propto f(1/b(a|s_1^2))$ and $b(a|s_2^2) \propto f(1/b(a|s_2^2))$ where $f(\cdot)$ is a function that hides other dependencies on variances of $s$ and its children.

The proof technique follows the approach of Lemma 1 but takes into account the multiple paths leading to the same state. The possibility of multiple paths results in the cyclical dependency of the sampling proportions in level 1 and 2. Note that in $T$ there is a single path to each state and this cyclical dependency does not arise. The full proof is given in Appendix J. Because of this cyclical dependency it is difficult to estimate the optimal sampling proportions in $G$. However, we can approximate the optimal sampling proportion that ignores the multiple path problem in $G$ by using the tree formulation in the following way: At every time $t$ during a trajectory $T^t$ call the Algorithm $[\text{J}]$ in Appendix J to estimate $B_T(s) = \mathbb{E}[b_L(s) + b_R(s) + b_D(s) + b_U(s)]$ stores the expected standard deviation of the state $s$ at iteration $t$. After $L$ such iteration we use the value $B_0(s)$ to estimate $b(a|s)$ as follows:

$$b^*(a|s) \propto \sqrt{\pi^2(a|s) \sigma^2(s, a) + \gamma^2 \sum_{s'} P(s'|s, a) B_0^2(s')}. $$

Note that for a terminal state $s$ we have the transition probability $P(s'|s, a) = 0$ and then the $b(a|s) = \pi(a|s)\sigma(s, a)$. This iterative procedure follows from the tree formulation in Theorem 1 and is necessary in $G$ to take into account the multiple paths to a particular state. Also observe that in Algorithm 1 we use value-iteration for the episodic setting [Sutton and Barto, 2018] to estimate the the optimal sampling proportion iteratively.

6 EMPIRICAL STUDY

We next verify our theoretical findings with simulated policy evaluation tasks in both a tree MDP and a non-tree Gridworld domain. Our experiments are designed to answer the following questions: 1) can ReVar produce policy value estimates with MSE comparable to the oracle solution? and 2) does our novel algorithm lower MSE relative to on-policy sampling of actions? Full implementation details are given in Appendix J.

![Figure 2: (Left) Deterministic 4-depth Tree. (Right) Stochastic Gridworld. The vertical axis gives MSE and the horizontal axis is the number of episodes collected. Axes use a log-scale and confidence bars show one standard error.](image)

**Figure 2:** (Left) Deterministic 4-depth Tree. (Right) Stochastic Gridworld.

**Experiment 1 (Tree):** In this setting we have a 4-depth 2-action deterministic tree MDP $T$ consisting of 15 states. Each state has a low variance arm with $\sigma^2(s, 1) = 0.01$ and high target probability $\pi(1|s) = 0.95$ and a high variance arm with $\sigma^2(s, 1) = 20.0$ and low target probability $\pi(2|s) = 0.05$. Hence, the Onpolicy sampling which samples according to $\pi$ will sample the second (high variance) arm less and suffer a high MSE. The CB-Var policy is a bandit policy that uses an empirical Bernstein Inequality [Maurer and Pontil, 2009] to sample an action without looking ahead and suffers high MSE. The Oracle has access to the model and variances and performs the best. ReVar lowers MSE comparable to Onpolicy and CB-Var and eventually matches the oracle’s MSE.

**Experiment 2 (Gridworld):** In this setting we have a $4 \times 4$ stochastic gridworld consisting of 16 grid cells. Considering the current episode time-step as part of the state, this MDP is a DAG MDP in which there are multiple path to a single state. There is a single starting location at the top-left corner and a single terminal state at the bottom-right corner. Let $L, R, D, U$ denote the left, right, down and up actions in every state. Then in each state the right and down actions have low variance arms with $\sigma^2(s, R) = \sigma^2(s, D) = 0.01$ and high target policy probability $\pi(R|s) = \pi(D|s) = 0.45$. The left and top actions have high variance arms with $\sigma^2(s, L) = \sigma^2(s, U) = 0.01$ and low target policy probability $\pi(L|s) = \pi(U|s) = 0.05$. Hence, Onpolicy which goes right and down with high probability (to reach the terminal state) will sample the low variance arms more and suffer a high MSE. Similar to above, CB-Var fails to look ahead when selecting actions and thus suffers from high MSE. ReVar lowers MSE compared to Onpolicy and CB-Var and actually matches and then reduces MSE compared to the Oracle. We point out that the DAG structure of the Gridworld violates the tree-structure under which Oracle and ReVar were derived. Nevertheless, both methods lower MSE compared to Onpolicy.
7 CONCLUSION AND FUTURE WORKS

This paper has studied the question of how to take actions for minimal-variance policy evaluation of a fixed target policy. We developed a theoretical foundation for data collection in policy evaluation by deriving an oracle data collection policy for the class of finite, tree-structured MDPs. We then introduced a practical algorithm, ReVar, that approximates the oracle strategy by computing an upper confidence bound on the variance of the future cumulative reward at each state and using this bound in place of the true variances in the oracle strategy. We bound the finite-sample regret (excess MSE) of our algorithm relative to the oracle strategy. We also present an empirical study where we show that ReVar decreases the MSE of policy evaluation relative to several baseline data collection strategies including on-policy sampling. In the future, we would like to extend our derivation of optimal data collection strategies and regret analysis of ReVar to a more general class of MDPs, in particular, relaxing the tree structure and also considering infinite-horizon MDPs. Finally, real world problems often require function approximation to deal with large state and action spaces. This setting raises new theoretical and implementation challenges for ReVar where we intend to incorporate experimental design approaches [Pukelsheim 2006, Mason et al. 2021, Mukherjee et al. 2022]. Another interesting direction is to incorporate structure in the reward distribution of arms [Gupta et al. 2021, 2020].Addressing these challenges is an interesting direction for future work.

Acknowledgements: The authors will like to thank Kevin Jamieson from Allen School of Computer Science & Engineering, University of Washington for pointing out several useful references. This work was partially supported by AFOSR/AFRL grant FA9550-18-1-0166.

References


