Empirical Policy Evaluation With Supergraphs

Daniel Vial and Vijay Subramanian, *Senior Member, IEEE*

***Abstract*—We devise algorithms for the policy evaluation problem in reinforcement learning, assuming access to a simulator and certain side information called the *supergraph*. Our algorithms explore backward from high-cost states to find high-value ones, in contrast to approaches that work forward from all states. While several papers have demonstrated the utility of backward exploration empirically, we conduct rigorous analyses which show that our algorithms can reduce average-case sample complexity from $O(S \log S)$ to as low as $O(\log S)$. Analytically, we adapt tools from the network science literature to provide a new methodology for reinforcement learning problems.**

***Index Terms*—Reinforcement learning, Markov decision processes, policy evaluation, sample complexity, PageRank.**

I. Introduction

REINFORCEMENT learning (RL) [1], [2], [3] is a machine learning paradigm with applications in many domains. At a high level, RL studies agents interacting with uncertain environments – by taking actions, observing the effects of those actions, and incurring costs – in hopes of achieving some goal. Mathematically, this is often cast in the following Markov decision process (MDP) model [4]. Let $\mathcal{S}$ and $\mathcal{A}$ be finite sets of states and actions, respectively; for simplicity, we assume $\mathcal{S} = \{1, \dots, S\}$ for some $S \in \mathbb{N}$ throughout the paper. The uncertain environment is modeled by a controlled Markov chain with transition matrix P, i.e.,

$$\begin{aligned}&\mathbb{P}\big(S_{t+1} = s' | S_t = s, A_t = a\big)\\ &\quad = P\big(s'|s,a\big) \quad \forall\, s, s' \in \mathcal{S}, a \in \mathcal{A}\end{aligned}$$

where $\{S_t\}_{t=0}^{\infty}$ and $\{A_t\}_{t=0}^{\infty}$ are the random sequences of states and actions, respectively. State-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ incurs instantaneous cost $c(s,a) \in \mathbb{R}_+$. Mappings $\pi : \mathcal{S} \to \mathcal{A}$ are called *policies* and dictate the action taken at each state, i.e., $A_t = \pi(S_t)$. If the initial state is $s \in \mathcal{S}$ and the agent follows policy π, it incurs discounted cost

$$v_\pi(s) = \mathbb{E}_\pi\left[(1-\gamma)\sum_{t=0}^{\infty}\gamma^t c(S_t, A_t)\middle| S_0 = s\right]$$

$$= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P_\pi^t(s,\cdot)c_\pi \tag{1}$$

where $\mathbb{E}_\pi$ means $A_t = \pi(S_t)$ inside the expectation, $\gamma \in (0,1)$ is a *discount factor*, and $P_\pi(s,s') = P(s'|s,\pi(s))$ and $c_\pi(s) = c(s,\pi(s))$ are the transition matrix and cost vector induced by the policy π.

To find good policies – roughly, π for which v_π is small – one often needs to estimate v_π for a fixed π, when the MDP model is unknown but can be sampled from (we will soon make this precise). For example, the *empirical policy iteration* algorithm of [5] iteratively estimates v_π and greedily updates π. Moving forward, we focus on the former step, which we call *empirical policy evaluation* (EPE). The policy π will thus be fixed for the remainder of the paper, so we dispense with this subscript in (1) and (with slight abuse of notation) define our problem as follows. Let $\gamma \in (0,1)$ be a discount factor, $c \in \mathbb{R}_+^S$ a cost vector, and P an $S \times S$ row stochastic matrix. We seek an algorithm to estimate the *value function*

$$v = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P^t c = (1-\gamma)(I-\gamma P)^{-1}c. \tag{2}$$

We assume the algorithm has access to a *simulator*, i.e., P is unknown but the agent can sample random states distributed as $P(s,\cdot)$ (for any $s \in \mathcal{S}$) via interaction with the environment. Since this interaction can be costly in applications, we aim to estimate (2) with as few samples as possible. In contrast to some works, we also assume c is a known input to the algorithm. Thus, our algorithms are suitable for goal-oriented applications where one knows instantaneous costs *a priori* – for instance, which states correspond to winning or losing if the MDP models a game – and aims to estimate long-term discounted costs – for instance, how good or bad non-terminal configurations of the game are.

A. Forward vs. Backward Exploration

To contextualize our contributions, we contrast two approaches to EPE. The first is one of forward exploration, where v is estimated by sampling trajectories beginning at each state. We focus on a typical scheme employed in, e.g., [5], which we refer to as the *forward approach* for the remainder of the paper, and which proceeds as follows. First, let $\{W_t\}_{t=0}^{\infty}$ be a Markov chain with transition matrix P, fix $s \in \mathcal{S}$ and $T \in \mathbb{N}$, and rewrite (2) as

$$v(s) = (1-\gamma)\sum_{t=0}^{T-1}\gamma^t \mathbb{E}[c(W_t)|W_0 = s] + O\big(\|c\|_\infty \gamma^T\big). \tag{3}$$

Here the $O(\|c\|_\infty \gamma^T)$ bias can be made small if T is chosen large, and the first term can be estimated by simulating

Manuscript received October 15, 2020; revised April 8, 2021; accepted April 8, 2021. Date of publication April 16, 2021; date of current version June 21, 2021. This work was supported in part by NSF under Grant EPCN:1603861 and Grant CIF:AF:2008130. *(Corresponding author: Daniel Vial.)*

Daniel Vial was with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109 USA. He is now with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78701 USA (e-mail: dvial@utexas.edu).

Vijay Subramanian is with the Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: vgsubram@umich.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSAIT.2021.3073257, provided by the authors. The material includes appendices containing proofs and experimental details.

Digital Object Identifier 10.1109/JSAIT.2021.3073257

length-T trajectories. More specifically, let $\{W_t^{s,i}\}_{t=0}^{T-1}$ be a trajectory obtained as follows: set $W_0^{s,i} = s$ and, for $t \in \{1, \ldots, T-1\}$, sample $W_t^{s,i}$ from $P(W_{t-1}^{s,i}, \cdot)$. Letting $m \in \mathbb{N}$ and repeating this for $i \in \{1, \ldots, m\}$, we obtain an unbiased estimate of the first term in (3):

$$\frac{1}{m}\sum_{i=1}^{m}(1-\gamma)\sum_{t=0}^{T-1}\gamma^t c\Big(W_t^{s,i}\Big).$$

This forward approach is simple both algorithmically and analytically; see Appendix I of the supplementary material. However, since trajectories must be sampled starting at each state to estimate v, $\Omega(S)$ samples are fundamentally required, which may be prohibitive in practice.

The second approach we consider is one of backward exploration. This approach relies on the idea that if there are only a few high-cost states with only a few trajectories leading to them, it is more efficient to work backward along just these trajectories (or along a small set containing them) to identify high-value states (those s for which $v(s)$ is large). Put differently, if P and c are sparse, intuition suggests that backward exploration from high-cost states is more sample-efficient than forward exploration from all states. While intuitively reasonable, there are two issues that prevent backward exploration from reducing the linear sample complexity of the forward approach. First, the agent must identify high-cost states in order to explore backward from them, without visiting all states. Second, the agent must explore a small set of trajectories likely to lead to high-cost states, without starting at each state and filtering out trajectories that do not reach the high-cost set.

Several algorithmic and computational approaches have been proposed to combat these issues. For instance, [6] uses observed state-action-cost sequences to train a model that generates samples of state-action pairs likely to lead to a given state. This allows the agent to construct simulated trajectories that are guaranteed to lead to high-cost states, addressing the second issue; the observed sequences are also used to identify high-cost states, addressing the first issue. The authors of [7] similarly train a model that predicts which trajectories lead to high-cost states while assuming costs are known *a priori*. In a different vein, [8] considers physical tasks like a robot navigating a maze which have clear goal states, addressing the first issue. The state-action space is assumed to have a certain continuity – "small" actions (e.g., a robot moving a small distance) lead to "nearby" states (e.g., physically close locations) – addressing the second issue. (We do note the algorithms [6], [7], [8] operate on trajectories, whereas we require a simulator.)

Despite all of the promising results [6], [7], [8] observed in practice for backward exploration, theoretical understanding of its limits and the determination of any guarantees are missing in the literature. Furthermore, a framework within which one can develop backward exploration algorithms with provable guarantees while addressing the issues discussed above is also missing. Our approach to developing such a framework is as follows. First, as mentioned above, we assume the cost vector is known *a priori* (like [7] and similar to [8]). Second, we assume the agent is provided certain side information: $A \in \{0,1\}^{S\times S}$ satisfying the "absolute continuity" condition

$$A(s,s') = 0 \Rightarrow P(s,s') = 0 \quad \forall\, s, s' \in \mathcal{S}. \tag{4}$$

Note we can view A as the adjacency matrix for a graph whose edges are a superset of those in the graph induced by P; thus, we refer to this side information as the *supergraph*. The utility of the supergraph is that it allows the agent to determine which states may be "close" to high-cost states in the induced graph, which may allow for construction of trajectories leading to such states. In this work, we assume the supergraph is provided and do not address the important practical consideration of how to actually obtain it. However, we do note it can likely be obtained from domain knowledge. For instance, in a robot navigation task like [8], one-step transitions between physically distant states s and s' may be impossible, which would allow us to conclude $P(s,s') = 0$ *a priori* and set $A(s,s') = 0$. Unlike [8], however, our supergraph assumption does not depend on state-action continuity and thus should hold more generally; for example, if the MDP models a game, the game's rules may prevent transitions from s to s', so that $A(s,s') = 0$. We also emphasize that the reverse of the implication in (4) need not hold. Thus, one can always set $A(s,s') = 1\ \forall\ s, s'$ to ensure that (4) holds. Of course, there is a trade off; as will be seen, our algorithms are most efficient when A is sparse in a certain sense.

B. Our Contributions

In the remainder of the paper, we devise two backward exploration-based EPE algorithms that exploit the supergraph. Unlike [6], [7], [8], which only present empirical results, our algorithms are amenable to rigorous accuracy and sample complexity guarantees. Thus, beyond developing a framework within which to generate backward exploration algorithms, our main contribution is to offer theoretical evidence for the empirical success of backward exploration. More precisely, our contributions are as follows.

First, we devise an algorithm called `Backward-EPE` in Section II that uses the supergraph to discover high-value states while working backward from high-cost ones. We establish l_∞ accuracy and worst-case sample complexity $O(S\log S)$, equivalent to the average-case complexity of the forward approach. More notably, we show the average-case sample complexity of `Backward-EPE` is $O(\bar{d}(\|c\|_1/\|c\|_\infty)\log S)$, where $\bar{d}$ is the average degree in the supergraph. Note the problem-instance dependent sparsity parameter in this bound, $\bar{d}(\|c\|_1/\|c\|_\infty)$, precisely captures the intuition that backward exploration depends on how many high-cost states are present (the $\|c\|_1/\|c\|_\infty$ term) and how many trajectories lead to them (the $\bar{d}$ term, which is a function of the transition matrix). In the extreme case, $\bar{d}\|c\|_1/\|c\|_\infty = O(1)$, in which case `Backward-EPE` reduces sample complexity from $S\log S$ to $\log S$.

Next, we combine `Backward-EPE` with the forward approach for our second algorithm `Bidirectional-EPE` in Section III. We establish a (pseudo)-relative error guarantee for this algorithm, which we argue is useful in, e.g., empirical policy iteration. Analytically, we show `Bidirectional-EPE`

reduces the sample complexity of a plug-in method with the same accuracy guarantee; empirically, we show it is more efficient than using the backward or forward approach alone.

Both of our algorithms are inspired by methods that estimate *PageRank* [9], a node centrality measure from the network science literature that resembles the value function (2). Critically, however, the PageRank estimation literature assumes P is known, so the extension to EPE is non-trivial. Thus, another contribution of this work to adapt PageRank estimators to EPE. A further contribution of our work is methodological: we generalize a deterministic fixed-point equation (FPE) from the PageRank setting to a family of random FPEs in the EPE setting (see Lemma 1 in Section II-B and Lemma 2 in Appendix B of the supplementary material). These FPEs are distinct from the Bellman equation and provide new tools for RL problems like EPE; in particular, they allow us to derive sample complexity bounds for our algorithms. Moreover, we address several technical subtleties that arise when applying these FPEs (see Remarks 3, 4, and 5). Finally, we note that side information to guide backward exploration appears to be necessary to prove such FPEs in the RL setting. Thus, another methodological contribution is our supergraph construction, which enables derivation of FPEs while being intuitively grounded in, and motivated by, real-world settings like robots and games. To demonstrate the utility of these methodological contributions, we discuss other problems where we believe our approach can be used in Section IV, along with other related work.

Frequently-used notation: For a matrix B and any $t \in \mathbb{N}$, $B^t(s, s')$, $B^t(s, \cdot)$, and $B^t(\cdot, s')$ denote the (s, s')-th entry, s-th row, and s'-th column of B^t, respectively. We write $0_{n\times m}$ and $1_{n\times m}$ for the $n \times m$ matrices of zeroes and ones, respectively. Matrix transpose is denoted by $^\mathsf{T}$. We use $1(\cdot)$ for the indicator function, i.e., $1(E) = 1$ if statement E is true and $1(E) = 0$ otherwise. For $s \in \mathcal{S}$, e_s is the S-length vector with 1 in the s-th entry and 0 elsewhere, i.e., $e_s(s') = 1(s = s')$. Also for $s \in \mathcal{S}$, $N_{\text{in}}(s) = \{s' \in \mathcal{S} : A(s', s) = 1\}$ and $d_{\text{in}}(s) = |N_{\text{in}}(s)|$ are the incoming neighbors and in-degree of s in the supergraph. Average degree is denoted by $\bar{d} = (1/S)\sum_{s,s'=1}^{S} A(s, s') = (1/S)\sum_{s=1}^{S} d_{\text{in}}(s)$. For $\{a_n\}_{n\in\mathbb{N}}, \{b_n\}_{n\in\mathbb{N}} \subset [0, \infty)$, we use the standard asymptotic notation $a_n = O(b_n)$, $a_n = \Omega(b_n)$, $a_n = \Theta(b_n)$, and $a_n = o(b_n)$, resp., if $\limsup_{n\to\infty}(a_n/b_n) < \infty$, $\liminf_{n\to\infty}(a_n/b_n) > 0$, $a_n = O(b_n)$ and $a_n = \Omega(b_n)$, and $\lim_{n\to\infty}(a_n/b_n) = 0$, resp. All random variables are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with $\mathbb{E}[\cdot] = \int_\Omega \cdot \, d\mathbb{P}$ denoting expectation and *a.s.* meaning $\mathbb{P}$-almost surely.

II. Backward Empirical Policy Evaluation

A. Algorithm

Our first algorithm is called `Backward-EPE` and is based on the `Approx-Contributions` PageRank estimator from [10]. The latter algorithm restricts attention to the case $c = e_{s^*}$ for some $s^* \in \mathcal{S}$ and assumes P is known; our algorithm is a fairly natural generalization to the case $c \in \mathbb{R}_+^S$ and unknown P. For brevity, we restrict attention to `Backward-EPE` in this section. For transparency, Appendix A of the supplementary material discusses `Approx-Contributions` and clarifies which aspects of our analysis are borrowed from [10] and other existing work.

Algorithm 1: `Backward-EPE`

Input: Simulator for transition matrix P; cost vector c; discount factor γ; supergraph in-neighbors $\{N_{in}(s)\}_{s=1}^S$; termination parameter ε; per-state sample count n

$k = 0$, $\hat{v}_k = 0_{S\times 1}$, $r_k = c$, $U_k = \emptyset$, $\hat{P}_k = 0_{S\times S}$
while $\|r_k\|_\infty > \varepsilon$ **do**
  $k \leftarrow k + 1$, $s_k \sim \arg\max_{s\in\mathcal{S}} r_{k-1}(s)$ uniformly,
  $U_k = U_{k-1} \cup N_{in}(s_k)$
  for $s \in \mathcal{S}$ **do**
    if $s \in N_{in}(s_k) \setminus U_{k-1}$ **then** $\{X_{s,i}\}_{i=1}^n \sim P(s, \cdot)$,
    $\hat{P}_k(s, \cdot) = \frac{1}{n}\sum_{i=1}^n 1(X_{s,i} = \cdot)$; **else**
    $\hat{P}_k(s, \cdot) = \hat{P}_{k-1}(s, \cdot)$;
  end
  for $s \in \mathcal{S}$ **do**
    if $s = s_k$ **then** $\hat{v}_k(s) = \hat{v}_{k-1}(s) + (1 - \gamma)r_{k-1}(s)$,
    $r_k(s) = \gamma\hat{P}_k(s, s_k)r_{k-1}(s_k)$; **else** $\hat{v}_k(s) = \hat{v}_{k-1}(s)$,
    $r_k(s) = r_{k-1}(s) + \gamma\hat{P}_k(s, s_k)r_{k-1}(s_k)$;
  end
end

Output: Estimate $\hat{v}_{k_*} = \hat{v}_k$ of $v = (1 - \gamma)\sum_{t=0}^\infty \gamma^t P^t c$

`Backward-EPE` is defined in Algorithm 1. The algorithm takes as input cost vector c, discount factor γ, and desired accuracy ε, and initializes four variables: a value function estimate $\hat{v}_0 = 0_{S\times 1}$, a residual error vector $r_0 = c$, a set $U_0 = \emptyset$ we call the *encountered set*, and a transition matrix estimate $\hat{P}_0 = 0_{S\times S}$. Conceptually, the algorithm then works backward from high-cost states, iteratively pushing mass from residual vector to estimate vector so as to improve the estimate of v. More precisely, the first iteration proceeds as follows. First, a high-cost state s_1 is chosen ($s_1 \in \mathcal{S}$ such that $r_0(s_1) = c(s_1)$ is maximal) and its incoming supergraph neighbors $N_{\text{in}}(s_1)$ are added to the encountered set (first line in `while` loop). For $s \in U_1 = N_{\text{in}}(s_1)$ – i.e., s for which $P(s, s_1)$ may be nonzero by (4) – an estimate $\hat{P}_1(s, \cdot)$ of $P(s, \cdot)$ is computed using n samples (first `for` loop). The estimate $\hat{v}_1(s_1)$ is then incremented with the $(1 - \gamma)r_0(s) = (1 - \gamma)c(s)$ component of $v(s_1)$, and $\hat{P}_1(s, s_1)$ is used to estimate the $P(s, s_1)r_0(s_1) = P(s, s_1)c(s_1)$ component of $v(s)$ and to increment the corresponding residual $r_1(s)$ (second `for` loop).

In subsequent iterations k, the iterative update proceeds analogously, choosing s_k to maximize $r_{k-1}(s_k)$, adding $N_{\text{in}}(s_k)$ to the encountered set, incrementing $\hat{v}_k(s_k)$ by $(1-\gamma)r_{k-1}(s)$, and using an estimate of $P(s, s_k)r_{k-1}(s_k)$ to increment $r_k(s)$. The only distinction is that at iteration k, $P(s, \cdot)$ is estimated only for states $s \in U_k \setminus U_{k-1}$. Put differently, the first time we encounter state s – i.e., the first k for which $s \in N_{\text{in}}(s_k)$ – we estimate $P(s, \cdot)$; we then retain that estimate for the remainder of the algorithm. Thus, the encountered set U_k tracks the rows of P we have estimated up to and including iteration k. Alternatively, one could estimate $P(s, s_k)$ with independent samples at each iteration k for which $s \in N_{\text{in}}(s_k)$; we discuss the merits of this approach in Section IV.

Remark 1: In practice, one can maintain a single $\hat{P}$ matrix in Algorithm 1, updating rows $N_{\text{in}}(s_k) \setminus U_{k-1}$ in the first `for` loop and leaving other rows unchanged. Similarly, one can maintain $\hat{v}, r$ vectors and only modify rows $N_{\text{in}}(s_k) \cup \{s_k\}$ in the second `for` loop. In short, the computational complexity of iteration k is typically lower than S; it appears to be S in Algorithm 1 only because we distinguish $\hat{P}_k, \hat{v}_k, r_k$ across iterations k for analytical clarity.

B. Invariant

The manner in which we update the estimate and residual vectors may appear opaque, but it allows us to prove the following analogue of a fixed-point equation (FPE) from [10]. The FPE from [10], and others like it in the PageRank literature, are fundamental analytical tools for PageRank estimators. As will be seen, our analogue is similarly critical for deriving sample complexity bounds for `Backward-EPE` and (we believe) backward exploration approaches more generally (see Section IV-A). Also, the forthcoming proof suggests that our supergraph construct is the "correct" side information to enable such analyzes in the RL setting.

To explain our FPE, let $k_* = \inf\{k \in \mathbb{Z}_+ : \|r_k\|_\infty \leq \varepsilon\}$ denote the iteration at which `Backward-EPE` terminates, and let $\hat{\mu}_s$ denote the s-th row of $(1-\gamma)(I-\gamma\hat{P}_{k_*})^{-1}$, so that $\hat{\mu}_s c$ is the value function for $s \in \mathcal{S}$ defined on the final estimate $\hat{P}_{k_*}$ of P. (More precisely, we will complete the unestimated rows $\mathcal{S}\setminus U_{k_*}$ of $\hat{P}_{k_*}$ – in a manner to be discussed shortly – to obtain a row stochastic estimate matrix. This will ensure that $\hat{\mu}_s c$ is a well-defined value function.) Then the result (roughly) says that the FPE $\hat{v}_k(s) + \hat{\mu}_s r_k = \hat{\mu}_s c$ is preserved across iterations $k \in \{0, \ldots, k_*\}$. Conceptually, this means that if we run the algorithm until it terminates to obtain $\hat{P}_{k_*}$, and then look back at the sequence $\{\hat{v}_k, r_k\}_{k=0}^{k_*}$ generated by the algorithm, the FPE will have held at each k. This non-causality is somewhat unintuitive, yet crucial in our analysis.

Before we formally state the FPE result and prove it, we also highlight the difficulty in conceiving of such a result: at termination of the algorithm, the stochastic matrix P is partially estimated using $\hat{P}_{k_*}$ and a large class of stochastic matrices will agree with $\hat{P}_{k_*}$, but the FPE needs to hold irrespective of the particular stochastic matrix used as a completion of $\hat{P}_{k_*}$. The following lemma overcomes this issue by proving an FPE for every stochastic matrix which differs from $\hat{P}_{k_*}$ only in unestimated rows of P, i.e., rows indexed by $\mathcal{S} \setminus U_{k_*}$; in other words, simultaneously multiple FPEs hold.

Lemma 1: Let $Y_{s,i} \sim P(s,\cdot)\ \forall\ s \in \mathcal{S}, i \in [n]$, independent across s and i, and independent of the random variables in Algorithm 1. From $\{Y_{s,i}\}_{s\in\mathcal{S}, i\in[n]}$, define an offline estimate $\tilde{P}$ of P row-wise by $\tilde{P}(s,\cdot) = (1/n)\sum_{i=1}^n 1(Y_{s,i} = \cdot)$. Furthermore, define

$$\overline{P}(s,\cdot) = \begin{cases} \hat{P}_{k_*}(s,\cdot), & s \in U_{k_*} \\ \tilde{P}(s,\cdot), & s \in \mathcal{S}\setminus U_{k_*} \end{cases}$$

$$\overline{\mu_s} = (1-\gamma)e_s^\mathsf{T}\big(I-\gamma\overline{P}\big)^{-1}, \quad \overline{v}(s) = \overline{\mu_s}c \tag{5}$$

$$\underline{P}(s,\cdot) = \begin{cases} \hat{P}_{k_*}(s,\cdot), & s \in U_{k_*} \\ P(s,\cdot), & s \in \mathcal{S}\setminus U_{k_*} \end{cases}$$

$$\underline{\mu_s} = (1-\gamma)e_s^\mathsf{T}\big(I-\gamma\underline{P}\big)^{-1}, \quad \underline{v}(s) = \underline{\mu_s}c, \tag{6}$$

where $k_* = \inf\{k \in \mathbb{Z}_+ : \|r_k\|_\infty \leq \varepsilon\}$ is the iteration at which Algorithm 1 terminates. Then

$$\hat{v}_k(s) + \overline{\mu_s}r_k = \overline{v}(s), \quad \hat{v}_k(s) + \underline{\mu_s}r_k = \underline{v}(s)$$
$$\forall\ k \in \{0,\ldots,k_*\}, s \in \mathcal{S}\ a.s. \tag{7}$$

Remark 2: In words, $\overline{P}$ and $\underline{P}$ fill the unestimated rows of P with offline estimates and the true rows, respectively. More generally, an analogue of (7) holds whenever unestimated rows are consistent with the supergraph (4); see Lemma 2 in Appendix B of the supplementary material for details. We also emphasize the offline estimate $\tilde{P}$ is an analytical tool and does not affect sample complexity.

Proof of Lemma 1: We begin with the second identity in (7). We fix $s \in \mathcal{S}$ and use induction on k. For $k = 0$, (7) is immediate, since $\hat{v}_0 = 0_{S\times 1}$ and $r_0 = c$ in Algorithm 1. For $k \in [k_*]$, the iterative update of Algorithm 1 implies ($a.s.$)

$$\begin{aligned}
\hat{v}_k(s) + \underline{\mu_s}r_k &= \hat{v}_{k-1}(s) + (1-\gamma)r_{k-1}(s_k)1(s=s_k) \\
&\quad + \sum_{s'=1}^{S}\underline{\mu_s}(s')\Big(r_{k-1}(s')1\big(s'\neq s_k\big) \\
&\qquad + \gamma\hat{P}_k(s',s_k)r_{k-1}(s_k)\Big) \\
&= \hat{v}_{k-1}(s) + \underline{\mu_s}r_{k-1} \\
&\quad + r_{k-1}(s_k)\Big(-\underline{\mu_s}(s_k) + (1-\gamma)1(s=s_k) \\
&\qquad + \gamma\underline{\mu_s}\hat{P}_k(\cdot,s_k)\Big)
\end{aligned} \tag{8}$$

where for the second equality we added and subtracted $\underline{\mu_s}(s_k)r_{k-1}(s_k)$. Now since $\hat{v}_{k-1}(s) + \underline{\mu_s}r_{k-1} = \underline{v}(s)$ by the inductive hypothesis, and since by definition

$$\begin{aligned}
\underline{\mu_s}(s_k) - (1-\gamma)1(s=s_k) &= (1-\gamma)\sum_{t=1}^{\infty}\gamma^t\underline{P}^t(s,s_k) \\
&= \gamma(1-\gamma)\sum_{t=0}^{\infty}\gamma^t\underline{P}^t(s,\cdot)\underline{P}(\cdot,s_k) \\
&= \gamma\underline{\mu_s}\underline{P}(\cdot,s_k),
\end{aligned}$$

it suffices to show $\hat{P}_k(s',s_k) = \underline{P}(s',s_k)\ \forall\ s' \in \mathcal{S}$ (since then the term in parentheses in (8) will be zero). Towards this end, we fix $s' \in \mathcal{S}$ and consider two cases:

1) If $s' \in U_k$, Algorithm 1 implies $\hat{P}_k(s',s_k) = \hat{P}_{k_*}(s',s_k)$ (once we estimate $P(s',\cdot)$, our estimate remains unchanged). Moreover, $U_k \subset U_{k_*}$ in Algorithm 1 (the encountered set only grows), so $s' \in U_{k_*}$ and $\hat{P}_{k_*}(s',s_k) = \underline{P}(s',s_k)$ by definition of $\underline{P}$.
2) If $s' \notin U_k$, Algorithm 1 implies $\hat{P}_k(s',s_k) = 0$ (before encountering s', our estimate of $P(s',\cdot)$ is $0_{1\times S}$). On the other hand, $N_{\text{in}}(s_k) \subset U_k$, so $s' \notin N_{\text{in}}(s_k)$ and $A(s',s_k) = 0$ by definition of $N_{\text{in}}(s_k)$. Thus, $P(s',s_k) = 0$ by (4) and $\underline{P}(s',s_k) = 0$ by definition.

The proof of the first identity in (7) is identical, except in the very last step we must also use the fact that $P(s',s_k) = 0 \Rightarrow \tilde{P}(s',s_k) = 0$ by definition of $\tilde{P}$. ∎

Owing to the fact that (7) holds across iterations, we will refer to these identities as the $\overline{P}$-*invariant* and the $\underline{P}$-*invariant*, respectively. These invariants will be pivotal in the theorems

to come; interestingly though, only one invariant is useful for each theorem, while the other fails. This is due to technical subtleties discussed in Remarks 3, 4, and 5.

C. Results

Our first result is an accuracy guarantee for `Backward-EPE`.

Theorem 1: Fix $\varepsilon, \delta > 0$ and define

$$n^*(\varepsilon, \delta) = \frac{2\|c\|_\infty^2 \gamma^2}{\varepsilon^2 (1-\gamma)^2} \log\left(\frac{2S}{\delta}\left\lceil \frac{\log(4\|c\|_\infty/\varepsilon)}{1-\gamma}\right\rceil\right).$$

Then assuming $n \geq n^*(\varepsilon, \delta)$ in Algorithm 1, $\mathbb{P}(\|\hat{v}_{k_*} - v\|_\infty \geq 2\varepsilon) \leq \delta$.

Proof sketch: The full proof is deferred to Appendix C of the supplementary material but we sketch it here. First, by the $\overline{P}$-invariant (7), the triangle inequality, and the termination criteria of `Backward-EPE`,

$$\left\|\hat{v}_{k_*} - v\right\|_\infty \leq \left\|\hat{v}_{k_*} - \bar{v}\right\|_\infty + \|\bar{v} - v\|_\infty \leq \varepsilon + \|\bar{v} - v\|_\infty.$$

Now since $\bar{v}$ and v are the value functions corresponding to $\overline{P}$ and P, respectively, and since $\overline{P}$ is an unbiased estimate of P, we should expect the second term to be small for large n. However, this is not immediate, because $\bar{v}$ is a *biased* estimate of v in general. Thus, we further bound $\|\bar{v} - v\|_\infty$ by another random variable. First, for large T, similar to (3),

$$\|\bar{v} - v\|_\infty \leq (1-\gamma) \sum_{t=1}^{T-1} \gamma^t \left\|\left(\overline{P}^t - P^t\right)c\right\|_\infty + \frac{\varepsilon}{2}. \tag{9}$$

Second, using convexity of $\|\cdot\|_\infty$ and row stochasticity of $\overline{P}$, a simple calculation yields

$$\left\|\left(\overline{P}^t - P^t\right)c\right\|_\infty \leq \left\|\left(\overline{P}^{t-1} - P^{t-1}\right)c\right\|_\infty + \left\|\left(\overline{P} - P\right)P^{t-1}c\right\|_\infty.$$

Iterating this inequality and substituting into (9) gives a bound on $\|\bar{v} - v\|_\infty$ in terms of $\|(\overline{P} - P)P^{t-1}c\|_\infty$. Furthermore, this latter random variable has the same distribution as $\|(\tilde{P} - P)P^{t-1}c\|_\infty$, so we can bound $\|\bar{v} - v\|_\infty$ in terms of $\|(\tilde{P} - P)P^{t-1}c\|_\infty$ (see Remark 3). Finally, defining $d_{t-1} = P^{t-1}c$, the s-th entry of $\tilde{P}P^{t-1}c$ is

$$\sum_{s'=1}^{S} \tilde{P}(s, s') d_{t-1}(s') = \sum_{s'=1}^{S} \left(\frac{1}{n}\sum_{i=1}^{n} 1\left(Y_{s,i} = s'\right)\right) d_{t-1}(s') = \frac{1}{n}\sum_{i=1}^{n} d_{t-1}\left(Y_{s,i}\right),$$

and similarly, the s-th entry of $PP^{t-1}c$ is $\mathbb{E}d_{t-1}(Y_{s,i})$. Therefore,

$$\left\|\left(\tilde{P} - P\right)P^{t-1}c\right\|_\infty = \max_{s \in \mathcal{S}} \left|\frac{1}{n}\sum_{i=1}^{n}\left(d_{t-1}\left(Y_{s,i}\right) - \mathbb{E}d_{t-1}\left(Y_{s,i}\right)\right)\right|$$

which is bounded by ε with high probability by standard Chernoff bounds. ∎

Remark 3: It may seem wasteful that we use the $\overline{P}$-invariant instead of the $\underline{P}$-invariant, since $\underline{P}$ fills unestimated rows of $\hat{P}_{k_*}$ with the actual rows of P, and thus $\underline{v}$ should be a better estimate of v. We explain this choice as follows. First note that by the arguments in the proof sketch, bounding $\|\underline{v} - v\|_\infty$ amounts to bounding $\|(\underline{P} - P)P^{t-1}c\|_\infty$. It is tempting to use the union bound to bound such terms as

$$\mathbb{P}\left(\left\|\left(\underline{P} - P\right)P^{t-1}c\right\|_\infty \geq \eta \middle| U_{k_*}\right) \leq \sum_{s \in U_{k_*}} \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}\left(d_{t-1}\left(X_{s,i}\right) - \mathbb{E}d_{t-1}\left(X_{s,i}\right)\right)\right| \geq \eta \middle| U_{k_*}\right).$$

The issue with this approach is that there is a complicated dependence between $\{X_{s,i}\}_{i=1}^n$ and U_{k_*} in Algorithm 1. We also note that we replace $\|(\overline{P} - P)P^{t-1}c\|_\infty$ by $\|(\tilde{P} - P)P^{t-1}c\|_\infty$ in the proof sketch above owing to a similar issue.

Theorem 1 says that if we take $n \geq n^*(\varepsilon, \delta)$ samples per state encountered, the estimate $\hat{v}_{k_*}$ produced by `Backward-EPE` will be 2ε-accurate. Since `Backward-EPE` encounters $|U_{k_*}|$ states by definition, the total number of samples needed to ensure 2ε-accuracy is $n^*(\varepsilon, \delta)|U_{k_*}|$. Hence, our next goal is to bound $|U_{k_*}|$, in order to bound this overall complexity. By the backward exploration intuition discussed in Section I, we should expect a nontrivial bound $|U_{k_*}| = o(S)$ if the cost vector and supergraph are sufficiently sparse. However, even when both objects are maximally sparse, one can construct adversarial examples for which $U_{k_*} = \mathcal{S}$. For instance, suppose we restrict to c having a single high-cost state and the supergraph to having the minimal number of edges S. Then taking $c = [1\ 0\ \ldots\ 0]$ and $A = 1_{S\times 1}e_1^\mathsf{T}$ will satisfy this restriction, but will yield $U_{k_*} = \mathcal{S}$ (assuming $\varepsilon < 1$). Note the key issue in this example (and, we suspect, in most adversarial examples) is the interaction between the cost vector and the supergraph; in particular, if high-cost states have high in-degrees, $|U_{k_*}|$ will be large (even if there are few high-cost states and few edges overall).

In light of this, our best hope for a nontrivial bound on $|U_{k_*}|$ is an average-case analysis, i.e., bounding $\mathbb{E}|U_{k_*}|$ while randomizing over the inputs of `Backward-EPE`. As it turns out, we only need to randomize over the cost vector; roughly, by considering random cost vectors for which the expected cost of any given state does not dominate the average expected cost. For such cost vectors, the interaction between cost and in-degree discussed in the previous paragraph will "average out," and consequently the adversarial examples will not dominate in expectation. This is formalized in the following theorem.

Theorem 2: Let C be an $\mathbb{R}_+^S$-valued random vector s.t. $\mathbb{E}\|C\|_1 < \infty, \mathbb{E}C(s) \leq \beta\mathbb{E}\|C\|_1/S =: \bar{c}$ for some constant $\beta \in [1, \infty)$. Then if Algorithm 1 is initialized with cost vector C,

$$\mathbb{E}\left|U_{k_*}\right| \leq \frac{S\bar{c}\bar{d}}{\varepsilon(1-\gamma)}$$

where the expectation is with respect to C and the randomness in Algorithm 1.

Proof: As for Theorem 1, we exploit the $\overline{P}$-invariant (7) (note we proved Lemma 1 for fixed c but the same arguments hold for random C owing to their almost-sure nature). First

observe

$$\overline{v}(s) \geq \hat{v}_{k_*}(s) = (1-\gamma)\sum_{k=1}^{k_*} r_{k-1}(s)1(s=s_k) \geq \varepsilon(1-\gamma)\sum_{k=1}^{k_*} 1(s=s_k)$$

where the first inequality holds by the $\overline{P}$-invariant (7), the equality by Algorithm 1, and the second inequality by definition of k_*. On the other hand, we have

$$\left|U_{k_*}\right| = \left|\cup_{s=1}^{k_*} N_{\text{in}}(s_k)\right| \leq \sum_{k=1}^{k_*} d_{\text{in}}(s_k) = \sum_{k=1}^{k_*}\sum_{s=1}^{S} d_{\text{in}}(s)1(s=s_k) = \sum_{s=1}^{S} d_{\text{in}}(s)\sum_{k=1}^{k_*} 1(s=s_k).$$

Combining the previous two inequalities and taking expectation, we have therefore shown

$$\mathbb{E}\left|U_{k_*}\right| \leq \frac{1}{\varepsilon(1-\gamma)}\sum_{s=1}^{S} d_{\text{in}}(s)\mathbb{E}\overline{v}(s). \tag{10}$$

Now consider $\mathbb{E}\overline{v}(s)$. By definition (5),

$$\mathbb{E}\overline{v}(s) = \mathbb{E}\overline{\mu_s}C = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathbb{E}\overline{P}^t(s,\cdot)C = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathbb{E}\Big[\mathbb{E}\Big[\overline{P}^t(s,\cdot)|C\Big]C\Big].$$

Now after realizing C, we fill some rows of $\overline{P}$ with samples generated during the algorithm and other rows with samples generated offline; in contrast, all rows of $\tilde{P}$ are filled with offline samples. But in either case, these samples have the same distribution, so we can replace $\overline{P}$ by $\tilde{P}$ in the previous equation. Moreover, $\tilde{P}$ is independent of the random variables in Algorithm 1, including $r_0 = C$. In summary,

$$\mathbb{E}\Big[\overline{P}^t(s,\cdot)|C\Big] = \mathbb{E}\big[\tilde{P}^t(s,\cdot)|C\big] = \mathbb{E}\big[\tilde{P}^t(s,\cdot)\big]. \tag{11}$$

Combining the previous two equations and using the assumption on C, we obtain

$$\mathbb{E}\overline{v}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathbb{E}\big[\tilde{P}^t(s,\cdot)\big]\mathbb{E}[C] \leq (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathbb{E}\big[\tilde{P}^t(s,\cdot)\big]\bar{c}1_{S\times 1} = \bar{c}$$

where we also used row stochasticity of $\tilde{P}$. Substituting into (10) completes the proof. ∎

Remark 4: This proof fails if we use the $\underline{P}$-invariant instead of the $\overline{P}$-invariant. In particular, we cannot express $\mathbb{E}[\underline{P}^t(s,\cdot)|C]$ as deterministic in (11), since C influences which states are encountered during the algorithm and thus influences which rows of $\underline{P}$ are estimates and which are exact. This illustrates the utility of the $\overline{P}$-invariant: it allows us to "decorrelate" the estimated transition matrix from the cost vector, i.e., to obtain $\mathbb{E}[\overline{P}^t(s,\cdot)C] = \mathbb{E}[\tilde{P}^t(s,\cdot)]\mathbb{E}[C]$. In the current work, this is our only use of this decorrelation trick, but it may useful in analyzes of algorithms like `Backward-EPE` (see Section IV).

D. Discussion

We now return to interpret our results and derive `Backward-EPE`'s overall sample complexity, which (we recall) is $n^*(\varepsilon,\delta)|U_{k_*}|$. In the worst case, $|U_{k_*}| = \Omega(S)$, and thus the worst-case sample complexity for fixed c is $O(Sn^*(\varepsilon,\delta))$. Neglecting log log factors and constants, ignoring log terms for quantities that have polynomial scaling (e.g., writing $\log(1/(1-\gamma))/(1-\gamma)^2$ as simply $1/(1-\gamma)^2$), and assuming γ is either constant or grows to 1, Theorem 1 implies

$$Sn^*(\varepsilon,\delta) = O\Big(S\log(S/\delta)\|c\|_\infty^2\varepsilon^{-2}(1-\gamma)^{-2}\Big).$$

For comparison, the complexity of the forward approach is

$$O\Big(S\log(S/\delta)\|c\|_\infty^2\varepsilon^{-2}(1-\gamma)^{-3}\Big) \tag{12}$$

(see Appendix I of the supplementary material). Thus, in the worst case `Backward-EPE` has similar complexity to that of the forward approach, with a slightly improved dependence on the discount factor γ (see Section IV-E). (The extra $1/(1-\gamma)$ factor in (12) arises since $O(1/(1-\gamma))$-length trajectories must be sampled to make the bias in (3) small.)

In the average case, however, the sample complexity of `Backward-EPE` can be dramatically lower. In particular, Theorem 2 implies average-case sample complexity

$$\mathbb{E}\left|U_{k_*}\right| \times n^*(\varepsilon,\delta) = O\left(\frac{S\bar{c}\bar{d}}{\varepsilon(1-\gamma)} \times \frac{\log(S/\delta)\|C\|_\infty^2}{\varepsilon^2(1-\gamma)^2}\right) = O\left(\frac{\|C\|_1\bar{d}\log(S/\delta)\|C\|_\infty^2}{\varepsilon^3(1-\gamma)^3}\right).$$

(This argument is not precise, since $\|C\|_\infty$ is random in Theorem 2; we address this shortly.) Thus, if γ, δ, and $\|C\|_\infty/\varepsilon$ are constants, `Backward-EPE` has average case complexity

$$O\big((\|C\|_1/\|C\|_\infty) \times \bar{d} \times \log S\big). \tag{13}$$

Note (13) captures the intuition that backward exploration is efficient when the costs and supergraph are sufficiently sparse, since $\|C\|_1/\|C\|_\infty$ and $\bar{d}$ quantify cost and supergraph sparsity, respectively. We also note that when γ, δ, and $\|C\|_\infty/\varepsilon$ are constants, the forward approach's complexity (12) becomes simply $O(S\log S)$. In the extreme case, $\|C\|_1/\|C\|_\infty, \bar{d} = O(1)$ and `Backward-EPE` reduces sample complexity to $O(\log S)$.

We can make this argument rigorous with further assumptions on C. For example, the following corollary considers random binary cost vectors with H nonzero entries. Such cost vectors could arise, for example, in MDP models of games, where states corresponding to losing configurations of the game have unit cost and other states have zero cost.

Corollary 1: Fix $H \in \mathcal{S}$ and define $\mathcal{C}_H = \{\sum_{s=1}^S a_s e_s : a_s \in \{0,1\}\ \forall\ s \in \mathcal{S}, \sum_{s=1}^S a_s = H\}$ to be the set of binary vectors with H nonzero entries. Assume the cost vector C is chosen uniformly at random from $\mathcal{C}_H$ and $\gamma, \delta, \varepsilon$ are constants. Then to guarantee $\mathbb{P}(\|\hat{v}_{k_*} - v\|_\infty \geq 2\varepsilon|C) \leq \delta$ *a.s.*, `Backward-EPE` requires $O(\min\{Hd, S\}\log S)$ samples in expectation.

Proof: See Appendix D of the supplementary material. ∎

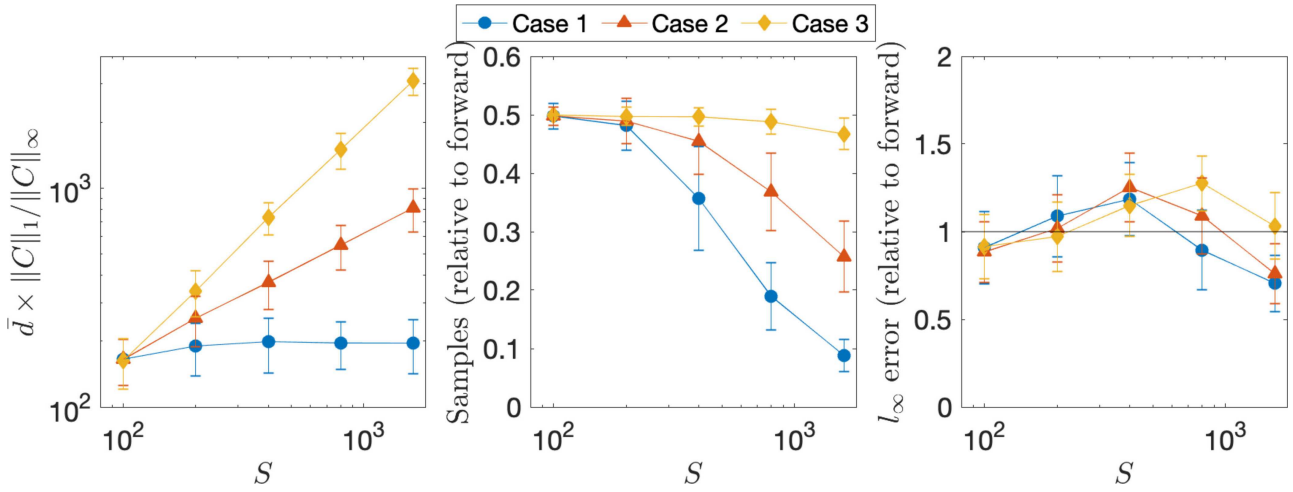

Fig. 1. Numerical illustration of `Backward-EPE`

E. Tightness of Our Bounds

At present, we lack lower bounds for the problem of EPE with supergraphs. In fact, even in the case where the cost vector has a single nonzero entry – i.e., the PageRank setting – the literature contains no tight lower bounds for the analogous PageRank problem, to the best of our knowledge. However, we next offer some speculative thoughts on this issue.

For simplicity, we restrict to the setting of Corollary 1. In this case, intuition suggests that any reasonable algorithm requires average sample complexity at least

$$\begin{aligned}&\mathbb{E}_C\left|\left\{s' \in \mathcal{S} : A(s', s) = C(s) = 1 \text{ for some } s \in \mathcal{S}\right\}\right| \\ &\quad = \mathbb{E}_C\left|\cup_{s\in\mathcal{S}:C(s)=1} N_{\text{in}}(s)\right| \end{aligned} \tag{14}$$

where the expectation is with respect to the random cost vector C from Corollary 1. Indeed, at a minimum, any algorithm should sample from $P(s', \cdot)$ whenever s' belongs to the set on the left side of (14), since these rows may contribute non-negligibly to $v(s')$.

Assuming our conjecture that (14) lower bounds sample complexity holds, we can compare it to the upper bound in Corollary 1 as follows:

$$\begin{aligned}\mathbb{E}_C\left|\cup_{s\in\mathcal{S}:C(s)=1} N_{\text{in}}(s)\right| &\le \sum_{s\in\mathcal{S}} d_{\text{in}}(s)\mathbb{P}_C(C(s)=1) \\ &= \sum_{s\in\mathcal{S}} d_{\text{in}}(s)H/S = H\bar{d}. \end{aligned} \tag{15}$$

Hence, the tightness of our upper bound is controlled by the inequality in (15), which is tight if the sets $\{N_{\text{in}}(s) : C(s) = 1\}$ have little "overlap". In other words, the gap (if any) depends on how much "double-counting" of incoming neighbors of high-cost states occurs (implicitly) in our analysis. As far as we are aware, sharpening the upper bound to account for this double-counting is an open problem even in the simpler PageRank case.

In summary, the gap in our upper bound arises from (15), but closing this gap is nontrivial even when $c = e_s$. Finally, we note that (15) *is* tight for certain instances of the supergraph. In Appendix E of the supplementary material, we provide such instances by showing the following: given H and d, there exists a supergraph with average degree $\bar{d}$ arbitrarily close to d such that (15) (and thus the upper bound of Corollary 1) is tight. See Claim 1 in Appendix E of the supplementary material for details.

F. Numerical Illustration

To conclude this section and to illustrate our analysis, we present empirical results in Figure 1. Here we generate random problem instances P, c in a manner that yields three different cases of the complexity factor $\bar{d}\|C\|_1/\|C\|_\infty$ identified above; roughly, $\Theta(1)$, $\Theta(\sqrt{S})$, and $\Theta(S)$ (left). In all cases, the sample complexity of `Backward-EPE` decays relative to that of the forward approach, suggesting sublinear complexity (middle). Moreover, the different scalings of $\bar{d}\|C\|_1/\|C\|_\infty$ reflect in different rates of decay in relative complexity, suggesting $\bar{d}\|C\|_1/\|C\|_\infty$ indeed determines sample complexity. We also note algorithmic parameters are chosen to ensure both algorithms yield similar l_∞ error (right). Error bars show standard deviation across problem instances. Further details can be found in Appendix H of the supplementary material.

III. Bidirectional Empirical Policy Evaluation

A. Algorithm

Our second algorithm is called `Bidirectional-EPE` and is inspired by the `Bidirectional-PPR` PageRank estimator from [11] (see Appendix A of the supplementary material for further discussion of this PageRank estimator). As will be seen, this algorithm is conducive to a stronger accuracy guarantee; namely, a (pseudo)-relative error guarantee. The utility of such a guarantee is that the resulting estimates tend to better preserve the ordering of the actual value function when compared to an l_∞ guarantee. Preserving this ordering is important in the problem of finding good policies; e.g., in the greedy update of policy iteration mentioned in Section I.

As its name suggests, `Bidirectional-EPE` proceeds in two stages: it first conducts backward exploration using `Backward-EPE`, then improves the resulting estimate via forward exploration. The analysis of this bidirectional approach relies on the $\underline{P}$-invariant (7). Similar to Theorem 1, we can make $|\underline{v}(s) - v(s)|$ small by taking n large in

`Backward-EPE`; when this holds, we have

$$v(s) \approx \underline{v}(s) = \hat{v}_{k_*}(s) + \underline{\mu_s} r_{k_*}. \tag{16}$$

Since $\underline{\mu_s}$ is a probability distribution over $\mathcal{S}$, the residual term in (16) satisfies

$$\underline{\mu_s} r_{k_*} = \mathbb{E}_{Z_s \sim \underline{\mu_s}} r_{k_*}(Z_s) \approx \frac{1}{n_F} \sum_{i=1}^{n_F} r_{k_*}\big(Z_{s,i}\big)$$

where in the approximate equality $\{Z_{s,i}\}_{i=1}^{n_F}$ are distributed as $\underline{\mu_s}$ and n_F is large. By (16),

$$v(s) \approx \hat{v}_{k_*}(s) + \frac{1}{n_F} \sum_{i=1}^{n_F} r_{k_*}\big(Z_{s,i}\big). \tag{17}$$

Intuitively, the right side of (17) is a more accurate estimate of $v(s)$ than $\hat{v}_{k_*}(s)$ alone; the only remaining question is how to generate $\{Z_{s,i}\}_{i=1}^{n_F}$. This can indeed be done in our model; namely, by generating Geometric$(1-\gamma)$-length trajectories on $\underline{P}$. More specifically, given $\underline{P}$, we first generate a Geometric$(1-\gamma)$ random variable $L_{s,i}$ and set $Z_{s,i}^0 = s$; we then sample $Z_{s,i}^t$ from $\underline{P}(Z_{s,i}^{t-1}, \cdot)$ for each $t \in [L_{s,i}]$; and finally we set $Z_{s,i} = Z_{s,i}^{L_{s,i}}$. Then conditioned on $\underline{P}$, $Z_{s,i}$ is distributed as $\underline{\mu_s}$. To see why, let $\mathbb{P}^{\underline{P}}$ denote probability conditioned on $\underline{P}$. Then

$$\begin{aligned}\mathbb{P}^{\underline{P}}\big(Z_{s,i} = s'\big) &= \sum_{t=0}^{\infty} \mathbb{P}^{\underline{P}}\big(Z_{s,i} = s' | L_{s,i} = t\big)\mathbb{P}^{\underline{P}}\big(L_{s,i} = t\big) \\ &= \sum_{t=0}^{\infty} \underline{P}^t(s,s')(1-\gamma)\gamma^t = \underline{\mu_s}(s').\end{aligned}$$

Thus, sampling from $\underline{\mu_s}$ amounts to sampling from $\underline{P}(s,\cdot)$. To do so, we either sample from $P(s,\cdot)$ (if $s \notin U_{k_*}$) or from $\hat{P}_{k_*}(s,\cdot)$ (if $s \in U_{k_*}$); the former is exactly what was done in `Backward-EPE`, and the latter can be done after running `Backward-EPE`. Put differently, to generate $Z_{s,i}$ we sample from $P(s,\cdot)$ unless we have already sampled from $P(s,\cdot)$ during `Backward-EPE`, in which case we sample from the empirical estimate $\hat{P}_{k_*}(s,\cdot)$ obtained during `Backward-EPE`.

The `Bidirectional-EPE` algorithm is formally defined in Algorithm 2. As above, write n_F for the per-state forward trajectory count; we also write n_B for the per-state sample count in the `Backward-EPE` subroutine. We denote the ultimate estimate of v by $\hat{v}_{BD}$.

Algorithm 2: `Bidirectional-EPE`

Input: Simulator for transition matrix P; cost vector c; discount factor γ; supergraph in-neighbors $\{N_{in}(s)\}_{s=1}^S$; termination parameter ε; per-state backward, forward sample counts n_B, n_F

1 Run `Backward-EPE` (Algorithm 1) with inputs P simulator, c, γ, $\{N_{in}(s)\}_{s=1}^S$, ε, n_B

2 Let $\hat{v}_{k_*}$, r_{k_*}, U_{k_*}, $\hat{P}_{k_*}$ be estimate vector, residual vector, encountered states, and P estimate at termination of `Backward-EPE`, and define $\underline{P}$ in (6)

3 **for** $s \in \mathcal{S}$ **do**

4 Generate samples $\{Z_{s,i}\}_{i=1}^n$ from $\underline{\mu_s}$, set $\hat{v}_{BD}(s) = \hat{v}_{k_*}(s) + \frac{1}{n_F}\sum_{i=1}^{n_F} r_{k_*}(Z_{s,i})$

5 **end**

Output: Estimate $\hat{v}_{BD}$ of $v = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P^t c$

B. Results

As alluded to above, `Bidirectional-EPE` is conducive to a pseudo-relative error guarantee. In particular, we have the following relative-plus-additive error bound. Note $v(s)$ can be arbitrarily small, so we should not expect a relative error guarantee for all states; however, for high-value states, the relative guarantee will dominate.

Theorem 3: Fix $\varepsilon_{rel} \in (0,1)$ and $\varepsilon_{abs}, \delta > 0$, and define

$$n_F^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta) = \frac{324\varepsilon \log(4S/\delta)}{\varepsilon_{rel}^2 \varepsilon_{abs}}$$

$$n_B^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta) = \frac{3\log\big(4S^2/\delta\big)}{\big(\log\big(1+\frac{\varepsilon_{rel}}{2}\big)\big)^2 \min_{i,j\in\mathcal{S}:P(i,j)>0} P(i,j)} \times \left\lceil \frac{\log(2\|c\|_\infty/\varepsilon_{abs})}{(1-\gamma)} \right\rceil^2.$$

Then assuming $n_F \geq n_F^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta)$ and $n_B \geq n_B^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta)$ in Algorithm 2, we have

$$\mathbb{P}\Big(\cup_{s=1}^S \big\{\big|\hat{v}_{BD}(s) - v(s)\big| > \varepsilon_{rel} v(s) + \varepsilon_{abs}\big\}\Big) \leq \delta. \tag{18}$$

Proof sketch: The proof separately treats errors arising from the backward and forward exploration stages; we briefly describe each stage here and defer details to Appendix F of the supplementary material. For the backward exploration stage, Lemma 3 in Appendix F of the supplementary material shows that if $n_B \geq n_B^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta)$,

$$\mathbb{P}\Big(\cup_{s=1}^S \Big\{\big|\underline{v}(s) - v(s)\big| > \frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2}\Big\}\Big) \leq \frac{\delta}{2} \tag{19}$$

with $\underline{v}$ defined as in (6). We prove (19) in two steps. First, we show that if $n_B \geq n_B^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta)$, then $|\underline{P}(s,s') - P(s,s')| \leq \lambda P(s,s')$ for some $\lambda \in (0,1)$ and all $s, s' \in \mathcal{S}$. This follows from standard Chernoff bounds after replacing $\underline{P}$ by $\tilde{P}$ (which is necessary for similar reasons as those discussed in Remark 3). Second, we show that if $|\underline{P}(s,s') - P(s,s')| \leq \lambda P(s,s')$, then $\underline{v}$ is close to v (in the sense of (19)). To illustrate the second step, note we can use the upper bound on $\underline{P}$ to write

$$\begin{aligned}(1-\gamma)\sum_{t=0}^{\bar{T}} \gamma^t \underline{P}^t(s,\cdot)c &\leq (1-\gamma)\sum_{t=0}^{\bar{T}} \gamma^t (1+\lambda)^t P^t(s,\cdot)c \\ &\leq (1+\lambda)^{\bar{T}}(1-\gamma)\sum_{t=0}^{\bar{T}} \gamma^t P^t(s,\cdot)c.\end{aligned}$$

For large enough $\bar{T}$, the left and right sides are within $\varepsilon_{abs}/2$-additive errors of $\underline{v}(s)$ and $(1+\lambda)^{\bar{T}}v(s)$, respectively; having chosen $\bar{T}$, we can choose λ to ensure $(1+\lambda)^{\bar{T}} \leq 1 + \varepsilon_{rel}/2$. It is from here that the relative-plus-additive guarantee of Theorem 3 arises.

For the forward exploration stage, we let $\mathcal{G}$ denote the σ-algebra generated by all the random variables from the `Backward-EPE` subroutine of `Bidirectional-EPE`, and we show that when $n_F \geq n_F^*(\varepsilon_{rel}, \varepsilon_{abs}, \delta)$,

$$\mathbb{P}\Big(\big|\hat{v}_{BD}(s) - \underline{v}(s)\big| \geq \frac{\varepsilon_{rel}}{2} v(s) + \frac{\varepsilon_{abs}}{2} \Big| \mathcal{G}\Big)$$

$$\times\ 1\left(\left|\underline{v}(s)-v(s)\right| \le \frac{\varepsilon_{rel}}{2}v(s)+\frac{\varepsilon_{abs}}{2}\right) \le \frac{\delta}{2S}. \quad (20)$$

Here we use probability conditioned on $\mathcal{G}$ so that the only randomness in $|\hat{v}_{BD}(s)-\underline{v}(s)|$ is that from the forward exploration. Moreover, we can bound this term by exploiting the $\underline{P}$-invariant to write $|\hat{v}_{BD}(s)-\underline{v}(s)|$ as the deviation of an empirical average from its mean (similar to the Theorem 1 proof sketch) and then use standard Chernoff bounds. Roughly speaking, this requires us to use the indicator function in (20) to replace $v(s)$ by $\underline{v}(s)$ in the probability term; we then separately address the cases of large $\underline{v}(s)$ and small $\underline{v}(s)$ using a modification of the approach from [11], which has a similar accuracy guarantee. ∎

Remark 5: While the choice of invariant used to prove Theorems 1 and 2 was subtle (see Remarks 3 and 4), choosing the $\underline{P}$-invariant for Theorem 3 is rather obvious, since we explicitly use the matrix $\underline{P}$ in Algorithm 2.

C. Discussion

We next discuss Theorem 3. To simplify notation, we restrict to the setting of Corollary 1; however, the key insights extend to the more general setting of Theorem 2. Also, we assume the relative error tolerance ε_{rel}, the discount factor γ, and inaccuracy probability δ are constants independent of S. Finally, we note Theorem 3 holds for random C; see Remark 9 in Appendix F of the supplementary material.

We begin by deriving the asymptotic sample complexity of `Bidirectional-EPE` in the setting of Corollary 1. For the backward exploration stage (i.e., the `Backward-EPE` subroutine), we require per-state sample complexity $n_B^*(\varepsilon_{rel},\varepsilon_{abs},\delta)$; note this is deterministic since $\|C\|_\infty = 1$ pointwise in Corollary 1. Thus, the average-case sample complexity is (by Corollary 1),

$$n_B^*(\varepsilon_{rel},\varepsilon_{abs},\delta)\mathbb{E}\left|U_{k_*}\right| = O\left(\frac{\log(S)\log(1/\varepsilon_{abs})}{\min_{i,j\in\mathcal{S}:P(i,j)>0}P(i,j)}\times\frac{H\bar{d}}{\varepsilon}\right). \quad (21)$$

For the forward exploration stage, we require $n_F^*(\varepsilon_{rel},\varepsilon_{abs},\delta) = O(\varepsilon\log(S)/\varepsilon_{abs})$ trajectories of expected length $\gamma/(1-\gamma)$ for each of S states. We are assuming γ is a constant, and thus the expected forward complexity is simply $O(\varepsilon S\log S/\varepsilon_{abs})$. Combined with (21), and writing K_{BD} for the overall expected sample of `Bidirectional-EPE` in the setting of Corollary 1,

$$K_{BD} = O\left(\frac{H\bar{d}\log(S)\log(1/\varepsilon_{abs})}{\varepsilon\min_{i,j\in\mathcal{S}:P(i,j)>0}P(i,j)}+\frac{\varepsilon S\log S}{\varepsilon_{abs}}\right). \quad (22)$$

Here the termination parameter ε for the `Backward-EPE` subroutine is a free parameter that can be chosen to minimize the overall sample complexity. For example,

$$\varepsilon = \Theta\left(\sqrt{\frac{H\bar{d}\varepsilon_{abs}}{S\min_{i,j\in\mathcal{S}:P(i,j)>0}P(i,j)}}\right)$$
$$\Rightarrow K_{BD} = O\left(\sqrt{\frac{SH\bar{d}}{\varepsilon_{abs}\min_{i,j\in\mathcal{S}:P(i,j)>0}P(i,j)}}\log S\right) \quad (23)$$

where for simplicity we wrote $\log(1/\varepsilon_{abs})/\sqrt{\varepsilon_{abs}}$ as simply $1/\sqrt{\varepsilon_{abs}}$ (note this choice of ε minimizes (22) if we also ignore the $\log(1/\varepsilon_{abs})$ term in that expression). To interpret (23), we consider a specific choice of ε_{abs}. To motivate this, we first observe that in the setting of Corollary 1,

$$\mathbb{E}v = (1-\gamma)\sum_{t=0}\gamma^t P^t\times\mathbb{E}C = (1-\gamma)\sum_{t=0}\gamma^t P^t\times\frac{H}{S}1_{S\times1} = \frac{H}{S}1_{S\times1}$$

i.e., the "typical" value is H/S. It is thus sensible to choose $\varepsilon_{abs} = \Theta(H/S)$, so that we obtain a relative guarantee for above-typical values and settle for the absolute guarantee for below-typical values. Substituting into (23), we conclude that `Bidirectional-EPE` requires

$$K_{BD} = O\left(\sqrt{\frac{\bar{d}}{\min_{i,j\in\mathcal{S}:P(i,j)>0}P(i,j)}}S\log S\right) \quad (24)$$

samples in order to guarantee (18) in the setting of Corollary 1.

It is interesting to compare `Bidirectional-EPE` to a plug-in estimator that lends itself to the same accuracy guarantee. For this plug-in estimator, we simply estimate v by computing $(1-\gamma)\sum_{t=0}^{\infty}\gamma^t\tilde{P}^tC$, where $\tilde{P}(s,\cdot) = (1/n)\sum_{i=1}^{n}1(Y_{s,i}=\cdot)$ with $Y_{s,i}\sim P(s,\cdot)$ as in Lemma 1. Then by the same argument following (47) in the proof of Theorem 3 in the supplementary material, the plug-in estimate will satisfy the guarantee (18) whenever $n \ge n_B^*(\varepsilon_{rel},\varepsilon_{abs},\delta)$. Consequently, the sample complexity of the plug-in estimator is, under the assumptions leading to (24),

$$Sn_B^*(\varepsilon_{rel},\varepsilon_{abs},\delta) = O\left(\frac{S\log S}{\min_{i,j\in\mathcal{S}:P(i,j)>0}P(i,j)}\right). \quad (25)$$

Comparing (24) and (25), we see `Bidirectional-EPE` is more efficient than the plug-in whenever $\bar{d} \le 1/\min_{i,j\in\mathcal{S}\,:\,P(i,j)>0}P(i,j)$. To interpret this inequality, first suppose the supergraph is the graph induced by P, i.e., $A(s,s')=0 \Leftrightarrow P(s,s')=0$. Then

$$\bar{d} = \frac{1}{S}\sum_{s\in\mathcal{S}}\sum_{s'\in\mathcal{S}:P(s,s')>0}\frac{P(s,s')}{P(s,s')} \le \frac{\frac{1}{S}\sum_{s\in\mathcal{S}}\sum_{s'\in\mathcal{S}:P(s,s')>0}P(s,s')}{\min_{i,j\in\mathcal{S}:P(i,j)>0}P(i,j)} = \frac{1}{\min_{i,j\in\mathcal{S}:P(i,j)>0}P(i,j)}$$

More generally, this suggests the complexity of `Bidirectional-EPE` is order-wise similar to that of the plug-in method whenever degrees in the supergraph and induced graph are order-wise similar. If additionally most transitions are roughly uniform (in the sense that $P(i,j)\approx P(i,j')$ if $P(i,j),P(i,j')>0$), then $1/\min_{i,j\in\mathcal{S}:P(i,j)>0}P(i,j) \approx \max_{s\in\mathcal{S}}d_{\text{out}}(s)$, where $d_{\text{out}}(s) = \sum_{s'=1}^{S}A(s,s')$ is the out-degree of s. Thus, $\bar{d} = o(1/\min_{i,j\in\mathcal{S}:P(i,j)>0}P(i,j))$ if the maximum out-degree dominates the average degree.

Remark 6: At a higher level, the dependence on $\min_{i,j\in\mathcal{S}:P(i,j)>0}P(i,j)$ is undesirable but seems to be

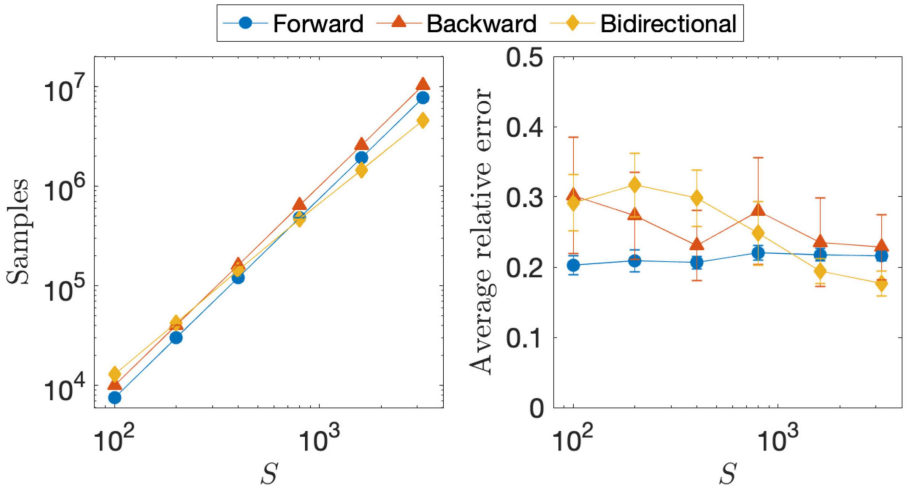

Fig. 2. Numerical illustration of `Bidirectional-EPE`

unavoidable for a relative error guarantee (see Appendix F-A of the supplementary material).

Generally, it is difficult to compare the sample complexity (24) to the bounds from Section II analytically, owing to the different error guarantees. Thus, we present an empirical comparison in Figure 2. Here we simulate all three algorithms using the case $\bar{d}\|C\|_1/\|C\|_\infty \approx \Theta(1)$ from Figure 1. We choose algorithmic parameters so that all algorithms maintain average relative error $(1/S)\sum_{s=1}^{S}(|\hat{v}(s) - v(s)|/v(s)) \approx 25\%$ across S (right). For these parameters, the sample complexities of the forward approach and `Backward-EPE` scale like S^2 (obtained via linear fits on a log-log scale, left). In contrast, the complexity of `Bidirectional-EPE` scales like $S^{1.7}$, suggesting a subquadratic sample complexity. Thus, as discussed above, `Bidirectional-EPE` appears more sample-efficient if one aims to bound relative error.

IV. Future Directions and Related Work

A. Adaptation of Other PageRank Algorithms

In this work, we adapted the PageRank estimators from [10], [11] to EPE. However, the PageRank literature contains many other algorithms either explicitly or conceptually related to these estimators, see, e.g., [12], [13], [14], [15], [16], [17]. Each of these algorithms rely on analyzes similar to that of [10], [11], which we extended to the EPE setting in this work. Thus, while we have focused on two specific algorithms in this paper, our analysis should be viewed as an example of how to extend a larger family of algorithms to EPE.

B. Finite Horizon Empirical Policy Evaluation

Another extension of this work is devising backward and bidirectional exploration-based EPE algorithms for the finite horizon cumulative cost value function

$$v(s) = \mathbb{E}\left[\sum_{t=0}^{T} c(Z_t)\middle| Z_0 = s\right] = \sum_{t=0}^{T} P^t(s,\cdot)c.$$

Here one aims to estimate multi-step transition distributions of the form $P^t(s,\cdot)$. Though our algorithms do not immediately apply, relevant analogues of `Approx-Contributions` exist in the case where P is known. In particular, [18] provides an algorithm to estimate $P^t(s,\cdot)$ when P is known. The algorithm is analogous to `Approx-Contributions` in that it explores backward from high-cost states. Moreover, [18] provides a bidirectional variant. Both of these algorithms could be adapted to EPE using our approach; this would yield analogues of `Backward-EPE` and `Bidirectional-EPE` for finite horizons.

C. Reusing Samples Versus Resampling

As mentioned in Section II, an alternative of `Backward-EPE` would take independent samples from $P(s,\cdot)$ for each $s \in N_{\text{in}}(s_k)$ and at each iteration k, rather than only sampling from $P(s,\cdot)$ when we first encounter s as in `Backward-EPE`. This alternative scheme is formally defined in Appendix G of the supplementary material. An interesting property is that, while the invariants of Lemma 1 fail, a related error process is a zero-mean martingale (see Appendix G of the supplementary material), and thus the ultimate estimate is unbiased. Analytically, this is an advantage over `Backward-EPE`, where the $\overline{P}$- and $\underline{P}$-invariants hold but the corresponding value functions $\overline{v}$ and $\underline{v}$ are biased estimates of v. The disadvantage of this alternative approach is that it may sample many times from each row of P, and thus the overall sample complexity may exceed that of the forward approach. Put differently, `Backward-EPE` is conservative in the sense that it performs no worse than the forward approach in the worst case (see Section II), but it sacrifices desirable properties that could perhaps improve performance in other cases. A useful avenue for future work would thus be to investigate this tradeoff.

D. PageRank Estimation With Limited Knowledge

A problem that has received little attention is PageRank estimation when the estimator has limited knowledge of P. For instance, consider a third party who wishes to identify influential Twitter users for advertising. Since PageRank serves as a measure of influence in networks, the third party may wish to identify high PageRank users, but the Twitter graph (as encoded by P) is not publicly available, and thus existing PageRank estimators do not apply. However, Twitter does

allow limited data requests [19], which may allow the third party to partially recover relevant entries of P. This setting could be abstracted as the follows: devise an algorithm that estimates PageRank while only sampling from P and minimizing sample complexity. This is similar to the problem we considered in this work, with one major difference: we justified the existence of a supergraph based on, for example, physical limitations that prevent transitions between states; if P represents Twitter, states (i.e., Twitter users) can be connected arbitrarily. Thus, we could perhaps replace knowledge of the supergraph with sampling of incoming neighbors, i.e., given a Twitter user, we can sample a random follower via data request. This would serve a similar purpose as the supergraph, and we suspect many of our ideas could be recycled.

E. Other Related Work

In addition to our discussion of [5] in Section I, we clarify some connections to other RL papers. First, recall from Section II-D that the complexity of `Backward-EPE` has a $(1-\gamma)^{-2}$ dependence on the discount factor γ. At first glance, this appears to violate a $(1-\gamma)^{-3}$ lower bound from [20]; see also [21] for matching upper bounds. The distinction here is that [20], [21] consider the problem of learning near-optimal policies, whereas `Backward-EPE` addresses the simpler task of learning the value function for a fixed policy. Next, we note the PageRank/RL connection was previously explored in [22]; however, [22] aimed to adapt RL algorithms to PageRank estimation, so our goals are complementary. Generally, we refer the reader to [1], [2], [3], [4] for general background on RL and MDPs.

Finally, we clarify distinctions between our work and others that exploit sparsity in reinforcement learning. Roughly speaking, these works presuppose that the value function can be approximated as $v \approx \Phi\theta$, where Φ is $S \times d$ feature matrix and θ is a $d \times 1$ parameter vector, with typically $d \ll S$. Here one can overparameterize by choosing d large – possibly larger than the number of observed transitions – and impose sparsity via ℓ_1 regularization to learn relevant features and avoid overfitting. See, for example, [23], [24], [25], [26] for policy evaluation, [27] for multi-task learning, and [28] for bandits. These papers are distinct from ours, which exploits sparsity in the transition matrix and cost vector and makes no assumption of a low-dimensional representation (of course, this means we do not benefit from such a representation either). Another distinction is that the aforementioned works assume the data is collected from a trajectory. Hence, while our work and [23], [24], [25], [26] all study policy evaluation, our formulation asks how data should be collected from the simulator, while [23], [24], [25], [26] ask how one can learn in feature space and select relevant features.

REFERENCES

[1] D. P. Bertsekas, *Reinforcement Learning and Optimal Control*. Belmont, MA, USA: Athena Sci., 2019.

[2] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA, USA: Athena Sci., 1996.

[3] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, hboxvol. 135. Cambridge, MA, USA: MIT Press, 1998.

[4] P. R. Kumar and P. Varaiya, *Stochastic Systems: Estimation, Identification, and Adaptive Control*. Philadelphia, PA, USA: SIAM, 2015.

[5] W. B. Haskell, R. Jain, and D. Kalathil, "Empirical dynamic programming," *Math. Oper. Res.*, vol. 41, no. 2, pp. 402–429, 2016.

[6] A. Goyal *et al.*, "Recall traces: Backtracking models for efficient reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–20.

[7] A. D. Edwards, L. Downs, and J. C. Davidson, "Forward-backward reinforcement learning," 2018. [Online]. Available: arXiv:1803.10227.

[8] C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel, "Reverse curriculum generation for reinforcement learning," in *Proc. Conf. Robot Learn.*, 2017, pp. 482–495.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the Web," Stanford InfoLab, Stanford, CA, USA, Rep. 1999-66, 1999.

[10] R. Andersen, C. Borgs, J. Chayes, J. Hopcroft, V. Mirrokni, and S.-H. Teng, "Local computation of PageRank contributions," *Internet Math.*, vol. 5, nos. 1–2, pp. 23–45, 2008.

[11] P. Lofgren, S. Banerjee, and A. Goel, "Personalized PageRank estimation and search: A bidirectional approach," in *Proc. 9th ACM Int. Conf. Web Search Data Min.*, 2016, pp. 163–172.

[12] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using PageRank vectors," in *Proc. IEEE Symp. Found. Comput. Sci.*, Berkeley, CA, USA, 2006, pp. 475–486.

[13] P. Berkhin, "Bookmark-coloring algorithm for personalized pagerank computing," *Internet Math.*, vol. 3, no. 1, pp. 41–62, 2006.

[14] G. Jeh and J. Widom, "Scaling personalized Web search," in *Proc. 12th Int. Conf. World Wide Web*, 2003, pp. 271–279.

[15] D. Vial and V. Subramanian, "On the role of clustering in personalized PageRank estimation," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 4, no. 4, p. 21, 2019.

[16] S. Wang, R. Yang, X. Xiao, Z. Wei, and Y. Yang, "FORA: Simple and effective approximate single-source personalized PageRank," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2017, pp. 505–514.

[17] H. Wang, Z. Wei, J. Gan, S. Wang, and Z. Huang, "Personalized PageRank to a target node, revisited," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2020, pp. 657–667.

[18] S. Banerjee and P. Lofgren, "Fast bidirectional probability estimation in Markov models," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2015, pp. 1423–1431.

[19] *Rate Limiting*, Twitter Develop. Doc., San Francisco, CA, USA. Accessed: Sep. 24, 2020. [Online]. Available: https://developer.twitter.com/en/docs/basics/rate-limiting

[20] M. G. Azar, R. Munos, and H. J. Kappen, "Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model," *Mach. Learn.*, vol. 91, no. 3, pp. 325–349, 2013.

[21] A. Sidford, M. Wang, X. Wu, L. F. Yang, and Y. Ye, "Near-optimal time and sample complexities for solving Markov decision processes with a generative model," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 5192–5202.

[22] V. S. Borkar and A. S. Mathkar, "Reinforcement learning for matrix computations: PageRank as an example," in *Proc. Int. Conf. Distrib. Comput. Internet Technol.*, 2014, pp. 14–24.

[23] M. Geist and B. Scherrer, "l_1-penalized projected Bellman residual," in *Proc. Eur. Workshop Reinforcement Learn.*, 2011, pp. 89–101.

[24] M. Ghavamzadeh, A. Lazaric, R. Munos, and M. Hoffman, "Finite-sample analysis of lasso-TD," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1177–1184.

[25] M. W. Hoffman, A. Lazaric, M. Ghavamzadeh, and R. Munos, "Regularized least squares temporal difference learning with nested l_2 and l_1 penalization," in *Proc. Eur. Workshop Reinforcement Learn.*, 2011, pp. 102–114.

[26] J. Z. Kolter and A. Y. Ng, "Regularization and feature selection in least-squares temporal difference learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 521–528.

[27] D. Calandriello, A. Lazaric, and M. Restelli, "Sparse multi-task reinforcement learning," *Intelligenza Artificiale*, vol. 9, no. 1, pp. 5–20, 2015.

[28] Y. Abbasi-Yadkori, D. Pal, and C. Szepesvari, "Online-to-confidence-set conversions and application to sparse stochastic bandits," in *Proc. 15th Int. Conf. Artif. Intell. Stat.*, 2012, pp. 1–9.

[29] P. Lofgren and A. Goel, "Personalized PageRank to a target node," 2013. [Online]. Available: arXiv:1304.4658.

[30] D. P. Dubhashi and A. Panconesi, *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2009.