Efficient, Low-Regret, Online Reinforcement Learning for Linear MDPs: Supplementary Material

ABSTRACT

CCS CONCEPTS

• **Theory of computation → Reinforcement learning**; *Unsupervised learning and clustering*.

ACM Reference Format:
. 2025. Efficient, Low-Regret, Online Reinforcement Learning for Linear MDPs: Supplementary Material. In . ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

A INTUITION FOR ALGORITHM 3

In this section we prove Proposition 8.

A.1 Some Useful Results

The following results will come handy in Appendix A.

Lemma 1. *If $x \in \mathbb{R}^d$ and $A, A' \succ 0 \in \mathbb{R}^{d\times d}$, then*

$$\left|\|x\|_A^2 - \|x\|_{A'}^2\right| \le \|x\|_2^2 \left\|A - A'\right\|_{2\to 2}.$$

Proof. We have that

$$\begin{aligned}\left|\|x\|_A^2 - \|x\|_{A'}^2\right| &= (x^\top A x) - (x^\top A' x)\\ &= x^\top (Ax) - x^\top (A'x)\\ &= x^\top ((A - A')x)\\ &\le \|x\|_2 \left\|(A - A')x\right\|_2\\ &\le \|x\|_2^2 \left\|A - A'\right\|_{2\to 2}.\end{aligned}$$

□

We require the following property of sub-sampling from a large set of independent samples.

Lemma 2. *Let $L := (v_1, \ldots, v_n)$ be a sequence of vectors sampled independently from $\mathcal{D}$, a distribution on $\mathbb{R}^d$. Let p be a distribution on $[n]$, and denote $\|p\|_2^2 := \sum_{i\in[n]} p(i)^2$. Then, with probability at least $1-\delta$, the distribution that arises from sampling $m \le \sqrt{\delta/\|p\|_2^2}$ vectors from L independently according to the distribution p is the same as the distribution that arises from sampling m vectors independently from the distribution $\mathcal{D}$.*

Proof. If we sample $z \in L$ with $\mathbf{Pr}[z = v_i] = p(i)$, then for any Borel set B of $\mathbb{R}^d$, we have

$$\begin{aligned}\mathbf{Pr}[z \in B] &= \sum_{i=1}^n \mathbf{Pr}[z \in B | z = v_i]\,\mathbf{Pr}[z = v_i]\\ &= \sum_{i=1}^n \mathbf{Pr}[v_i \in B] \cdot p(i) = \mathbf{Pr}[v_1 \in B],\end{aligned}$$

since the v_i's are identically distributed. This shows that z has the same distribution $\mathcal{D}$. If we sample z_1, z_2 from L independently according to p and get $z_1 = v_i$, $z_2 = v_j$ for $i, j \sim p$, the probability that $i = j$ is given by $\|p\|_2^2$. Hence, if we take m independent samples, the probability that there is at least one collision is upper bounded by $\binom{m}{2}\|p\|_2^2 \le m^2\|p\|_2^2$. So, for $m \le \sqrt{\delta/\|p\|_2^2}$, this probability is upper-bounded by δ. If no collisions happen in the m samples from p, then the samples are independent by construction of L. □

We now prove a version of Lemma 2 for continuous distributions.

Lemma 3. *Let $L := (v_i)_{i\in\mathcal{I}}$ be a sequence of vectors sampled independently from $\mathcal{D}$, a distribution on $\mathbb{R}^d$, where the index set $\mathcal{I}$ is* non-discrete*. Let p be a* continuous *distribution on $\mathcal{I}$. Then, with probability 1, the distribution that arises from sampling m vectors from L independently according to the distribution p is the same as the distribution that arises from sampling m vectors independently from $\mathcal{D}$.*

Proof. If we sample $z \in L$ by sampling $i \sim p$ and taking $z = v_i$, then for any Borel set B of $\mathbb{R}^d$, we have

$$\mathbf{Pr}[z \in B] = \int_{x\in\mathcal{I}} \mathbf{Pr}[z \in B | z = v_x] \cdot p(x)\,\mathrm{d}x = \mathbf{Pr}[v_1 \in B],$$

where the last step is valid since since $\mathbf{Pr}[z \in B | z = v_x]$ does not depend on x. This shows that z follows the distribution $\mathcal{D}$. If we sample $z_1, \ldots, z_m$ from L independently according to p, by the fact that p is continuous, we have that the probability of collisions between $z_1, \ldots, z_m$ is 0. In that case, the samples are independent by the construction of L. □

We require the following result by [3].

Theorem 4 ([1, 3]). *Consider a random vector X in $\mathbb{R}^d$. Let Σ be the covariance matrix of X and $\Sigma_N := \frac{1}{N}\sum_{i=1}^N w_i w_i^\top$ be the empirical covariance matrix corresponding to N i.i.d. samples $w_1, \ldots, w_N \sim \mathcal{N}(0, \Sigma)$. Then, with probability at least $1 - 2\exp\left(-O\left(\sqrt{d}\right)\right)$, it is the case that*

$$1 - O\left(\sqrt{\beta}\right) \le \lambda_{\min}(\Sigma_N) \le 1 + O\left(\sqrt{\beta}\right)$$

for $\beta := d/N$.

Drawing upon Theorem 4 we show the following lemma.

Lemma 5. *Let $w_1, \dots, w_k$ be drawn from $\mathcal{N}(0, \Sigma)$. Then the minimum eigenvalue of $\sum_{i=1}^{k} w_i w_i^\top$ is at least $k \cdot \lambda_{\min}(\Sigma)/100$, with probability at least $1 - 2\exp\left(-O\left(\sqrt{d}\right)\right)$.*

Proof. Let $\overline{\Lambda}_k := \frac{1}{k}\sum_{i=1}^{k} w_i w_i^\top$. It would suffice to show that $\lambda_{\min}(\overline{\Lambda}_k) \geq \lambda_{\min}(\Sigma)/100$ with high probability. Consider the Cholesky decomposition of Σ, that is, $\Sigma = UU^\top$ for some matrix U. We may now write $w_1, \dots, w_k$ as $Uz_1, \dots, Uz_k$, respectively, where $z_1, \dots, z_k \sim \mathcal{N}(0, I)$ are i.i.d. samples.

Then we have that

$$\begin{aligned}
\overline{\Lambda}_k &= \frac{1}{k}\sum_{i=1}^{k} w_i w_i^\top \\
&= \frac{1}{k}\sum_{i=1}^{k} U z_i z_i^\top U^\top \\
&= U\left(\frac{1}{k}\sum_{i=1}^{k} z_i z_i^\top\right) U^\top \\
&= U D_k U^\top,
\end{aligned}$$

whereby $D_k = \frac{1}{k}\sum_{i=1}^{k} z_i z_i^\top$. By Theorem 4, we get that

$$1 - O\left(\sqrt{\beta}\right) \leq \lambda_{\min}(D_k) \leq 1 + O\left(\sqrt{\beta}\right),$$

for $\beta := d/k$ with probability at least $1 - 2\exp\left(-O\left(\sqrt{d}\right)\right)$.

We shall now continue as follows. For a matrix A, the Rayleigh quotient definition of minimum eigenvalue states that $\lambda_{\min}(A) = \min_{x:\|x\|=1} x^\top A x$. Note that

$$\left\|U^\top x\right\|^2 = \left(U^\top x\right)^\top U^\top x = x^\top U U^\top x = x^\top \Sigma x \geq \lambda_{\min}(\Sigma).$$

Therefore

$$\begin{aligned}
\lambda_{\min}\left(\overline{\Lambda}_k\right) &= \lambda_{\min}(U D_k U^\top) \\
&= \min_{x:\|x\|=1} x^\top U D_k U^\top x \\
&= \min_{x:\|x\|=1} \left(U^\top x\right)^\top D_k U^\top x \\
&= \lambda_{\min}(\Sigma) \min_{x:\|x\|=1} \frac{\left(U^\top x\right)^\top}{\sqrt{\lambda_{\min}(\Sigma)}} D_k \frac{U^\top x}{\sqrt{\lambda_{\min}(\Sigma)}} \\
&\geq \lambda_{\min}(\Sigma) \min_{z:\|z\|=1} z^\top D_k z \\
&= \lambda_{\min}(D_k)\lambda_{\min}(\Sigma).
\end{aligned}$$

Since

$$\lambda_{\min}(\Sigma) - O(\lambda_{\min}(\Sigma))\, O\left(\sqrt{\beta}\right) \leq \lambda_{\min}(D_k)\, \lambda_{\min}(\Sigma)$$

by Theorem 4, we get that

$$\lambda_{\min}\left(\overline{\Lambda}_k\right) \geq \lambda_{\min}(\Sigma) - O(\lambda_{\min}(\Sigma))\, O\left(\sqrt{\beta}\right).$$

What is left is to show that $\lambda_{\min}(\Sigma) - O(\lambda_{\min}(\Sigma))\, O\left(\sqrt{\beta}\right)$ is at least $\lambda_{\min}(\Sigma)/100$ with high probability. The latter follows by the value of $\beta = d/k$ and the preceding discussion. □

A.2 Proof of Proposition 8

In the following, we shall assume that $H, \mathrm{Tr}(\Sigma), \left\|\Sigma^{-1}\right\|_{2\to 2} = O_d(1)$. Note that these are natural assumptions.

A.2.1 Convergence of $\Lambda_{h,k}^{-1}$ Implies Convergence in the Action-Value Function. Define

$$\Lambda_{h,k} := \sum_{i=1}^{k-1} \phi(s_{h,i}, a_{h,i})\phi(s_{h,i}, a_{h,i})^\top + \lambda I$$

for any h and k. We will say that *LSVI-UCB (Algorithm 1) converges at k_c* if, for all $k \ge k_c$,

$$\left\|\Lambda_{h,k}^{-1} - \Lambda_{h,k+1}^{-1}\right\|_{2\to 2} \le c/k^2,$$

for some $c > 0$ (that depends on d).

Let $w_{h,k}$ be the weight vector learned by LSVI-UCB at step h of episode k. We bound the difference between $w_{w,k}$ and $w_{h,k+1}$.

Lemma 6. *We have that*

$$\left\|w_{h,k} - w_{h,k+1}\right\|_2 \le O_d(1/k),$$

for all $k \ge \sqrt{K}$, with high probability.

Proof. Let

$$\psi_{h,k} := \sum_{i=1}^{k-1} \phi\left(s_{h,i}, a_{h,i}\right) r_h\left(s_{h,i}, a_{h,i}\right) + \sum_{i=1}^{k-1} \phi\left(s_{h,i}, a_{h,i}\right) \max_{a\in\mathcal{A}} Q_{h+1}\left(s_{h+1,i}, a\right)$$

and

$$z_{h,k} := r_h\left(s_{h,k}, a_{h,k}\right) + \max_{a\in\mathcal{A}} Q_{h+1}\left(s_{h+1,k}, a\right).$$

Note that $z_{h,k} \in [0, H+1]$ by the bounded reward assumption. We have that $w_{h,k} = \Lambda_{h,k}^{-1}\psi_{h,k}$, and

$$\begin{aligned} w_{h,k+1} - w_{h,k} &= \Lambda_{h,k+1}^{-1} \sum_{i=1}^{k} \phi\left(s_{h,i}, a_{h,i}\right) \left(r_h\left(s_{h,i}, a_{h,i}\right) + \max_a Q_{h+1}\left(s_{h+1,i}, a\right)\right) - w_{h,k} \\ &= \Lambda_{h,k+1}^{-1}\left(\psi_{h,k} + z_{h,k}\phi\left(s_{h,k}, a_{h,k}\right)\right) - w_{h,k} \\ &= \left(\Lambda_{h,k+1}^{-1} - \Lambda_{h,k}^{-1}\right)\psi_{h,k} + z_{h,k}\Lambda_{h,k+1}^{-1}\phi\left(s_{h,k}, a_{h,k}\right). \end{aligned}$$

Hence, using the triangle inequality and $0 \le z_{h,k} \le H+1$, we have

$$\left\|w_{h,k+1} - w_{h,k}\right\| \le \left\|\left(\Lambda_{h,k+1}^{-1} - \Lambda_{h,k}^{-1}\right)\psi_{h,k}\right\| + (H+1)\left\|\Lambda_{h,k+1}^{-1}\phi\left(s_{h,k}, a_{h,k}\right)\right\|.$$

To bound the first term, note that

$$\left\|\psi_{h,k}\right\|_2 \le (H+1)\cdot(k-1)\cdot O(\sqrt{\mathrm{Tr}(\Sigma)}),$$

and hence by our assumption we have that

$$\left\|\left(\Lambda_{h,k+1}^{-1} - \Lambda_{h,k}^{-1}\right)\psi_{h,k}\right\|_2 \le O_d\left(\frac{(H+1)\sqrt{\mathrm{Tr}(\Sigma)}}{k}\right).$$

To bound the second term, note that $\Lambda_{h,k+1}^{-1}$ is equal to

$$\left(\sum_{i=1}^{k} \phi\left(s_{h,i}, a_{h,i}\right)\phi\left(s_{h,i}, a_{h,i}\right)^\top + \lambda I\right)^{-1} \approx_{\mathrm{F}} \left(\sum_{i=1}^{k} \phi\left(s_{h,i}, a_{h,i}\right)\phi\left(s_{h,i}, a_{h,i}\right)^\top\right)^{-1} \approx_{\mathrm{F}} (k\Sigma)^{-1} = \frac{1}{k}\Sigma^{-1},$$

by setting $\lambda = o(1)$.

So we get that $\Lambda_{h,k+1}^{-1}\phi(s_{h,k}, a_{h,k}) \approx_{\mathrm{F}} \frac{1}{k}\Sigma^{-1}\phi(s_{h,k}, a_{h,k})$. Let us now take into account the fact that (with high probability) $\left\|\phi(s_{h,k}, a_{h,k})\right\|_2 \le O\left(\sqrt{\mathrm{Tr}(\Sigma)}\right)$ and the assumption that $\left\|\Sigma^{-1}\right\|_{2\to 2} = O_d(1)$. In this case, we have

$$\left\|\Lambda_{h,k+1}^{-1}\phi(s_{h,k}, a_{h,k})\right\|_2 \le \frac{1}{k}\cdot O_d\left(\sqrt{\mathrm{Tr}(\Sigma)}\right),$$

and so by the discussion above we get that

$$\left\|w_{h,k} - w_{h,k+1}\right\|_2 \le (H+1)\cdot O_d\left(\frac{\sqrt{\mathrm{Tr}(\Sigma)}}{k}\right) + (H+1)\cdot\frac{1}{k}\cdot O_d(\mathrm{Tr}(\Sigma)).$$

This concludes the proof as $H, \mathrm{Tr}(\Sigma) = O_d(1)$. □

The above yield a bound on $\left\|Q_{h,k} - Q_{h,k'}\right\|_\infty$, as follows.

Lemma 7. *If LSVI-UCB converges at $\sqrt{K}$, then for all $h \in [H]$ and $k' \ge k \ge \sqrt{K}$, we have that*

$$\left\|Q_{h,k} - Q_{h,k'}\right\|_\infty \le O_d\left(\log k' - \log k + \frac{\beta}{K^{1/4}}\right),$$

with high probability.

Proof. For any $k \in [K]$, $h \in [H]$ and $(s, a) \in \mathcal{S} \times \mathcal{A}$, we have

$$Q_{h,k}(s,a) := \min\left(w_{h,k}^{\top}\phi(s,a) + \beta \left\|\phi(s,a)\right\|_{\Lambda_{h,k}^{-1}}, H\right).$$

If both $Q_{h,k}(s,a)$ and $Q_{h,k+1}(s,a)$ are equal to H, the bound in the lemma is trivially satisfied for the (s,a) pair. Without loss of generality, we have that $Q_{h,k}(s,a) = w_{h,k}^{\top}\phi(s,a) + \beta\left(\phi(s,a)\,\Lambda_{h,k}^{-1}\phi(s,a)\right)^{1/2}$. Then we have

$$\begin{aligned}\left|Q_{h,k}(s,a) - Q_{h,k+1}(s,a)\right| &\leq \left|w_{h,k}^{\top}\phi(s,a) + \beta\left\|\phi(s,a)\right\|_{\Lambda_{h,k}^{-1}} - w_{h,k+1}^{\top}\phi(s,a) - \beta\left\|\phi(s,a)\right\|_{\Lambda_{h,k+1}^{-1}}\right| \\ &\leq \left|(w_{h,k} - w_{h,k+1})^{\top}\phi(s,a)\right| + \beta\left|\left\|\phi(s,a)\right\|_{\Lambda_{h,k}^{-1}} - \left\|\phi(s,a)\right\|_{\Lambda_{h,k+1}^{-1}}\right|. \end{aligned} \tag{1}$$

By Cauchy-Schwarz, the first term of the RHS of Equation (1) is at most

$$\left\|w_{h,k} - w_{h,k+1}\right\|_2 \left\|\phi(s,a)\right\|_2.$$

By Lemma 1, the second term of the RHS of Equation (1) is at most

$$\beta \cdot \left\|\phi(s,a)\right\|_2^2 \cdot \left\|\Lambda_{h,k}^{-1} - \Lambda_{h,k+1}^{-1}\right\|_{2\to 2}.$$

Therefore, by the assumptions of the lemma, we get

$$\left|Q_{h,k}(s,a) - Q_{h,k+1}(s,a)\right| \leq c_1\left\|\phi(s,a)\right\|_2/k + c_2\beta\left\|\phi(s,a)\right\|_2^2/k^2,$$

whereby the RHS on this inequality is $O_d(1/k + \beta/k^2)$, by taking into account the fact that $\|\phi(s,a)\|_2 = O(\mathrm{Tr}(\Sigma))$ with high probability, as $\phi(s,a)$ is drawn from the Gaussian distribution $\mathcal{N}(0,\Sigma)$, and the assumption that $\mathrm{Tr}(\Sigma) = O_d(1)$.

Therefore, we have

$$\left|Q_{h,k}(s,a) - Q_{h,k'}(s,a)\right| \leq \sum_{i=k}^{k'-1}\left|Q_{h,i}(s,a) - Q_{h,i+1}(s,a)\right| \sum_{i=k}^{k'-1} O_d\left(1/i + \beta/i^2\right).$$

Note that when $i \geq K^{3/4}$, we get that $\beta/i^2 \leq \beta/K^{1.5}$. Then, since $k \geq \sqrt{K}$ by assumption, we get

$$\sum_{i=k}^{K}\frac{\beta}{i^2} \leq K^{3/4}\frac{\beta}{K} + \frac{\beta}{\sqrt{K}} = O\left(\frac{\beta}{K^{1/4}}\right).$$

Thus,

$$\left|Q_{h,k}(s,a) - Q_{h,k'}(s,a)\right| \leq O_d\left(\log k' - \log k + \frac{\beta}{K^{1/4}}\right).$$

This concludes the proof. □

We define $\widehat{Q}$ to be the action-value function learned by Algorithm 3, whereas Q denotes the action-value function used by LSVI-UCB (Algorithm 1). Note that some of the values of $\widehat{Q}$ are identical to Q (in the learning intervals) and some might not be (in the rest of the episodes).

We use Lemma 7 to show the closeness Q and $\widehat{Q}$ in the non-learning intervals. If episode $k \in [K]$ is in a non-learning interval, let $k' < k$ denote the episode at the end of the last learning interval. By construction of Algorithm 3, we have $\widehat{Q}_{h,k} = \widehat{Q}_{h,k'} = Q_{h,k'}$. By applying Lemma 7, we get the following corollary.

Corollary 8. *If Algorithm 2 converges at $\sqrt{K}$, then for all $k > \sqrt{K}$, we have that*

$$\left\|Q_{h,k} - \widehat{Q}_{h,k}\right\|_{\infty} \leq \sum_{\ell=k'}^{k-1}\left\|Q_{h,\ell+1} - Q_{h,\ell}\right\|_{\infty} O_d\left(\log k - \log k' + \frac{\beta}{K^{1/4}}\right),$$

with high probability.

Finally, we bound the maximum difference between Q and $\widehat{Q}$.

Lemma 9. *For all ε, assuming $\left\|Q - \widehat{Q}\right\|_{\infty} \leq \varepsilon$ (with high probability), if it is the case that $(s', a') = \arg\max_{(s,a)} Q(s,a)$ and $(s'', a'') = \arg\max_{(s,a)} \widehat{Q}(s,a)$, then it is the case that $\left|Q(s',a') - \widehat{Q}(s'',a'')\right| \leq \varepsilon$ (with high probability).*

Proof. We have $\widehat{Q}(s',a') \in [Q(s',a') - \varepsilon, Q(s',a') + \varepsilon]$ by assumption, therefore $\widehat{Q}(s'',a'') \geq Q(s',a') - \varepsilon$ since (s'',a'') gives the maximum value of $\widehat{Q}$ by definition.

Similarly, $Q(s'',a'') \in \left[\widehat{Q}(s'',a'') - \varepsilon, \widehat{Q}(s'',a'') + \varepsilon\right]$, so that $Q(s',a') \geq \widehat{Q}(s'',a'') - \varepsilon$ if and only if $\widehat{Q}(s'',a'') \leq \widehat{Q}(s',a') + \varepsilon$. This concludes the proof. □

That is, we have shown that $\left\|\Lambda_{h,k}^{-1} - \Lambda_{h,k+1}^{-1}\right\|_{2\to 2} \le c/k^2$ (for sufficiently large k) implies that the action-value function of Algorithm 3, namely $\widehat{Q}$, is close to the action-value function of Algorithm 1, namely Q.

A.2.2 Convergence of $\Lambda_{h,k}^{-1}$ Under Some Assumptions. We show the following straightforward proposition.

Proposition 10. *With probability at least $1 - 1/K$, for all $h \in [H]$ and $k \in [K-1]$, we have*

$$\left\|\Lambda_{h,k}^{-1} - \Lambda_{h,k+1}^{-1}\right\|_{2\to 2} \le O_d(1/k^2).$$

Proof. By Assumption 6 and Assumption 7, using either Lemma 2 or Lemma 3 (depending on whether $\mathcal{S}$ is discrete or continuous), for any $h \in [H]$, with probability at least $1 - 1/T^2$, the values $\phi(s_{h,1}, a_{h,1}), \ldots, \phi(s_{h,K}, a_{h,K})$ are i.i.d samples from $\mathcal{N}(\mu, \Sigma)$. By a union bound, this holds for all h simultaneously with probability $1 - 1/K^2$.

By construction, $\Lambda_{h,k+1} = \Lambda_{h,k} + \phi(s_{h,k}, a_{h,k})\phi(s_{h,k}, a_{h,k})^\top$, so the Sherman-Morrison-Woodbury identity gives, for $u := \phi(s_{h,k}, a_{h,k})$ and $w := \Lambda_{h,k}^{-1}\phi(s_{h,k}, a_{h,k})$,

$$\begin{aligned}
\left\|\Lambda_{h,k}^{-1} - \Lambda_{h,k+1}^{-1}\right\|_{2\to 2} &= \left\|\frac{\Lambda_{h,k}^{-1} u u^\top \Lambda_{h,k}^{-1}}{1 + u^\top \Lambda_{h,k}^{-1} u}\right\|_{2\to 2} \\
&= \left\|\frac{w w^\top}{1 + u^\top \Lambda_{h,k}^{-1} u}\right\|_{2\to 2} \\
&= \frac{\left\|w w^\top\right\|_{2\to 2}}{\left\|1 + u^\top \Lambda_{h,k}^{-1} u\right\|_{2\to 2}} \\
&\le \left\|w w^\top\right\|_{2\to 2} \\
&= \|w\|_2^2 \\
&= \|\Lambda_{h,k}^{-1} \cdot \phi(s_{h,k}, a_{h,k})\|_2^2 \\
&\le \left\|\Lambda_{h,k}^{-1}\right\|_{2\to 2}^2 \left\|\phi(s_{h,k}, a_{h,k})\right\|_2^2,
\end{aligned}$$

by the fact that $\left\|1 + u^\top \Lambda_{h,k}^{-1} u\right\|_{2\to 2} \ge 1$.

What is left is to bound the quantities $\left\|\Lambda_{h,k}^{-1}\right\|_{2\to 2}^2, \left\|\phi(s_{h,k}, a_{h,k})\right\|_2^2$. To bound $\left\|\Lambda_{h,k}^{-1}\right\|_{2\to 2}^2$ our approach is to lower bound the smallest eigenvalue of $\Lambda_{h,k}$ by $\Omega(k)$, with high probability. By the definition of $\Lambda_{h,k}$ (see Algorithm 2) and Lemma 5, we get that the minimum eigenvalue of $\Lambda_{h,k}$ is at least $(k-1)\,\lambda_{\min}(\Sigma) + \lambda = \Omega(k)$ (by appropriately setting λ).

We now turn to bounding $\left\|\phi(s_{h,k}, a_{h,k})\right\|_2$. Since $\phi(s_{h,k}, a_{h,k}) \sim \mathcal{N}(\mu, \Sigma)$, it is a standard result that $\left\|\phi(s_{h,k}, a_{h,k})\right\|_2 = O\left(\sqrt{\mathrm{Tr}(\Sigma)}\right)$, with high probability. The result follows from the fact that $\mathrm{Tr}(\Sigma) = O_d(1)$. □

B RUNNING THE EXPERIMENTS

Our code can be found in the anonymized repository https://github.com/pseudonymousprocrastinator/efficient-low-regret-linear-mdp-rl.

B.1 Hardware and Software

We ran our experiments on cluster-machines with Intel Xeon Silver 4116 CPUs and obtained up to 24 CPU cores and 240 GB RAM in total (multiple experiments were run simultaneously). For the linearized environment experiments, we obtained a single NVIDIA Tesla T4 GPU. The machines run Ubuntu 22.04. The python version used was 3.10.12. The other packages in our environment, along with the versions, are listed in the requirements.txt file.

B.2 Generating the MDP Environments

B.2.1 Synthetic Environments. The file `linear_mdp.py` contains both the wrapper-class for a linear MDP environment as well as the code to generate a synthetic environment as described in Section 5. The arguments are, in order, num_states ($|\mathcal{S}|$), num_actions ($|\mathcal{A}|$), embedding_dim (d) and planning_horizon (H). The synthetic environment that we use in our experiments was generated with:

```
python linear_mdp.py 500 15 30 50
```

and can be found in `models/linear_mdp02.dat`.

B.2.2 Linearized Environments. The code to generate the linearized environments described in Section 5.1 is adapted from the implementation of [4] which we obtained from https://github.com/shelowize/lvrep-rl. We only use the contrastive representation learning part of the implementation. It can be invoked via `thirdparty/lvrep-rl/main_mod.py`. For generating our linearized environments, it was invoked as follows (for the Alien dataset):

```
python main_mod.py -env 'ALE/Alien-ram-v5' -horizon 100
    -start_timesteps 1000 -max_timesteps 50000
    -eval_freq 500 -hidden_dim 1024
    -feature_dim 512 -o alien_agent.dat
```

The command needs to be run in the `thirdparty/lvrep-rl` subdirectory. The crucial parameters are horizon (H), hidden_dim (number of neurons in the hidden layer of the neural net) and feature_dim (d).

The neural nets generated by the CTRLSAC implementation needs to be embedded in a LinearMDP class for use with our algorithms, as follows:

```
python linearize_interface.py -d alien_agent.dat
    -o models/linearized_alien.dat 100
```

This command needs to be run in the source code root directory. `-d` gives the file name of the saved neural net data, and `-o` gives the output file name. The last parameter, `100`, is the planning horizon H.

B.2.3 The Experiments. All the experiments can be run from `experiment_base.py`. The syntax is as follows:

```
python experiment_base.py -o <output-folder>
    -n <num-jobs> -k-min <K-min>
    -num-reps <num-reps> -chunk-size <op-chunk-size>
    <alg-specific-arguments> <alg> <K_max> <mdp_file>
```

The required arguments are: alg — which can be one of {basic, alt_fixed, alt_adaptive} — K_max — which is a number specifying the largest K (total number of episodes) to consider while measuring the regret-growth etc. — and mdp_file — which gives the file name of the serialized object holding the environment information (generated as described previously). The general optional arguments are:

-o The output folder name.
-n The number of worker processes (jobs) to use.
–k-min The smallest K (number of episodes) to start from. Default is 100.
–num-reps The number of repetitions for each possible K.
–chunk-size The number of entries in each output chunk.

The regret/process time/space usage growth will be measured by running the selected algorithm on the given environment for K episodes, choosing a sequence of values of K from –k-min (which we choose to be 100) to K_max (which we choose to be 500, 5, 000, 15, 000 etc. for the different experiments) in increments of 20.

For each choice of K, we run the experiments –num-reps many times and take the average (of the regret, space usage, etc.). For the synthetic environment experiments, we set –num-reps to be 5, whereas we use –num-reps=1 in the linearized environment experiments due to resource constraints. The output is produced as multiple CSV files, each containing –chunk-size entries (corresponding to a block of K values).

The baseline algorithm (LSVI-UCB), which can be selected using alg=basic, does not require any additional parameters for execution (as discussed in Section 5, the λ and β hyperparameters are hardcoded as per [2]). See Table 1.

Table 1: Execution parameters.

alt_fixed	–learn-iters-base-exp	$\rho \in (0, 1]$ such that the algorithm resets after every K^{ρ} steps.
alt_adaptive	–lookback-period	m, the number of previous episodes checked in the LEARN condition.
	–alt-threshold	$\tau_c > 0$, such that $\tau = \tau_c \cdot d^2$ in the LEARN condition.
	–learn-iters-budget-exp	$e \in (0, 1]$, such that Budget is set to K^e.
	–max-phase-len-exp	$\rho \in (0, 1]$ such that the algorithm resets after at most K^{ρ} steps.

We try various possible combinations of values for these parameters, as can be seen in Section 5.

REFERENCES

[1] Radosław Adamczak, Alexander E. Litvak, Alain Pajor, and Nicole Tomczak-Jaegermann. 2012. Sharp bounds on the rate of convergence of the empirical covariance matrix. arXiv:1012.0294 [math.PR]

[2] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. 2023. Provably Efficient Reinforcement Learning with Linear Function Approximation. *Math. Oper. Res.* 48, 3 (2023), 1496–1521.

[3] Roman Vershynin. 2010. How close is the sample covariance matrix to the actual covariance matrix? arXiv:1004.3484 [math.PR]

[4] Tianjun Zhang, Tongzheng Ren, Mengjiao Yang, Joseph Gonzalez, Dale Schuurmans, and Bo Dai. 2022. Making Linear MDPs Practical via Contrastive Representation Learning. In *ICML 2022 (Proceedings of Machine Learning Research, Vol. 162)*. ML Research Press, Cambdridge, MA, 26447–26466.