Supplementary material to "Optimistic planning for Markov decision processes": Proofs

Proof of main result

Recall that to prove Theorem 1, it must first be shown that the regret of the algorithm is related to the smallest α among expanded nodes (which will be done in Lemma 6), and then that the algorithm always works to decrease this smallest α (done in Lemma 7). A preliminary result is also needed.

Lemma 5. *The ν-values of the near-optimal policy classes increase over iterations: $\nu(H^*_{t+1}) \geq \nu(H^*_t)$, where $H^*_t \in \arg\max_{H \in \mathcal{T}_t} \nu(H)$.*

Proof. Consider first one policy class H, split by expanding some leaf node $s \in \mathcal{L}(\mathcal{T}_H)$. One child class H' is obtained for each action u, and we have $\mathcal{L}(\mathcal{T}_{H'}) = (\mathcal{L}(\mathcal{T}_H) \setminus \{s\}) \cup \mathcal{C}(s,u)$. By easy calculations, since the rewards are positive, the terms that nodes $\mathcal{C}(s,u)$ contribute to $\nu(H')$ add up to more than the term of s in $\nu(H)$, and the other terms remain constant. Thus $\nu(H') \geq \nu(H)$. Then, among the policy classes $H_t \in \mathcal{T}_t$, some are split in $\mathcal{T}_{t+1}$ and some remain unchanged. For the children of split classes ν-values are larger than their parents'; while ν-values of unchanged classes remain constant. Thus, the maximal ν-value increases across iterations. Note it can similarly be shown that $b(H^\dagger_{t+1}) \leq b(H^\dagger_t)$. □

Lemma 6. *Define $\alpha_t = \alpha(s_t)$, the α value of the node expanded at iteration t; and $\alpha^* = \min_{t=0,\dots,n-1} \alpha_t$. The regret after n expansions satisfies $\mathcal{R}_n \leq \frac{N}{\gamma}\alpha^*$.*

Proof. We will first bound, individually at each iteration t, the suboptimality of $\nu(H^*_t)$, by showing:

$$v^* - \nu(H^*_t) \leq \mathrm{diam}(H^\dagger_t) \leq \frac{N}{\gamma}\alpha_t \tag{7}$$

To this end, observe that:

$$\nu(H^\dagger_t) \leq \nu(H^*_t) \leq v^* \leq b(H^\dagger_t) \tag{8}$$

The inequality $\nu(H^*_t) \leq v^*$ is true by definition ($\nu(H^*_t)$ is a lower bound on the value of some policy, itself smaller than v^*). For the leftmost inequality, H^*_t maximizes the lower bound across all policy classes compatible with the current tree, so its lower bound is at least as large as that of the optimistic policy class $H^\dagger_t$. Similarly, for the rightmost inequality, since $H^\dagger_t$ maximizes the upper bound, its upper bound is immediately larger than the true optimal value. Using this string of inequalities, we get:

$$\begin{aligned} v^* - \nu(H^*_t) &\leq b(H^\dagger_t) - \nu(H^\dagger_t) \\ &= \mathrm{diam}(H^\dagger_t) = \sum_{s \in \mathcal{L}(\mathcal{T}_{H^\dagger_t})} c(s) \end{aligned} \tag{9}$$

We now investigate the relationship between this diameter and α_t. Consider the subtree $\mathcal{T}_{H^\dagger_t}$ of policy class $H^\dagger_t$, represented schematically in Figure 4 using a black continuous outline (this subtree has a branching factor of N). We are thus interested in finding an upper bound for $\sum_{s \in \mathcal{L}(\mathcal{T}_{H^\dagger_t})} c(s)$ as a function of α_t. Consider the tree $\mathcal{T}_{hs_t}$, as introduced earlier in the definition of $n(s)$, which is included in $\mathcal{T}_{H^\dagger_t}$ and is the same for any $h \in H^\dagger_t$. To see this, recall that s_t maximizes c among the leaves of $\mathcal{T}_{H^\dagger_t}$. Since additionally c strictly decreases along paths, any node with a contribution larger than $c(s_t)$ must be above these leaves, and this holds for any $h \in H^\dagger_t$.

Denote in this context $\mathcal{T}_{hs_t}$ more simply by $\mathcal{T}'$, shown in gray in the figure, and its leaves by $\mathcal{L}'$, shown as a gray outline. Denote the *children* of $\mathcal{L}'$ by $\mathcal{L}''$, shown as a dashed line.

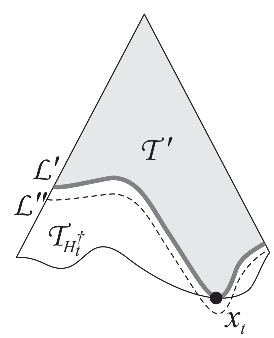

Figure 4: Tree of the optimistic policy class and various subtrees.

Recall that for any h and $s \in \mathcal{T}_h$, $\sum_{s' \in \mathcal{C}(s,h(s))} c(s') = \gamma c(s)$. This also means the sum of contributions for the leaves of any subtree of $\mathcal{T}_h$ having some s as its root is smaller than $c(s)$. Using these properties, we have:

$$\begin{aligned} \sum_{s \in \mathcal{L}(\mathcal{T}_{H^\dagger_t})} c(s) &\leq \sum_{s' \in \mathcal{L}'} c(s') = \frac{1}{\gamma} \sum_{s'' \in \mathcal{L}''} c(s'') \leq \frac{1}{\gamma} \sum_{s'' \in \mathcal{L}''} c(s_t) \\ &\leq \frac{1}{\gamma} N \left|\mathcal{L}'\right| c(s_t) \leq \frac{1}{\gamma} N n(s_t) c(s_t) = \frac{N}{\gamma}\alpha_t \end{aligned}$$

where we additionally exploited the facts that $c(s'') \leq c(s_t)$ (otherwise s'' would have been in $\mathcal{T}'$), that each node in $\mathcal{L}'$ has N children in $\mathcal{L}''$, and that by the definition of $n(s)$ $|\mathcal{L}'| \leq n(s_t)$. From this and also (9), the desired intermediate result (7) is obtained.

Using now (8) and (7), as well as Lemma 5, we have:

$$\begin{aligned}\mathcal{R}_n &= \max_u Q^*(x_0, u) - Q^*(x_0, H_n^*(s_0)) \\ &\leq v^* - \nu(H_n^*) \leq b(H_{t^*}^\dagger) - \nu(H_{t^*}^\dagger) \\ &= \operatorname{diam}(H_{t^*}^\dagger) \leq \frac{N}{\gamma}\alpha^*\end{aligned}$$

where $H_n^*(s_0)$ is the action chosen by OP at the root (i.e., in state x_0), and $t^* \in \arg\min_{t=0,\ldots,n-1} \alpha_t$. The first inequality is true because $\max_u Q^*(s_0, u) = v^*$ and $Q^*(s_0, H_n^*(s_0)) \geq \nu(H_n^*)$ (the return $Q^*(s_0, H_n^*(s_0))$ is obtained by choosing optimal actions below level 0, whereas H_n^* may make other suboptimal choices). The proof is complete. □

Lemma 7. *All nodes expanded by the algorithm belong to S_{α^*}, so that $n \leq |S_{\alpha^*}|$.*

Proof. We show first that $s_t \in S_{\alpha_t}$ at any iteration t. Condition (i) in the definition (5) of S_{α_t} is immediately true. For condition (ii), an $\frac{N}{\gamma}\alpha_t$-optimal policy h whose tree $\mathcal{T}_h$ contains s_t is needed. Choose any $h \in H_t^\dagger$, then $s_t \in \mathcal{T}_h$ and:

$$v^* - v(h) \leq b(H_t^\dagger) - \nu(H_t^\dagger) = \operatorname{diam}(H_t^\dagger) \leq \frac{N}{\gamma}\alpha_t$$

where we used some of the inequalities derived in the proof of Lemma 6. Thus $s_t \in S_{\alpha_t}$. Furthermore, $\alpha^* \leq \alpha_t$ implies $S_{\alpha_t} \subseteq S_{\alpha^*}$, and we are done. □

Proof of Theorem 1. Exploiting Lemma 7 in combination with (6):

- if $\beta > 0$, $n = \tilde{O}(\alpha^{*-\beta})$, thus for large n, $\alpha^* = \tilde{O}(n^{-\frac{1}{\beta}})$;
- if $\beta = 0$, $n \leq a\left(\log \frac{1}{\alpha^*}\right)^b$, thus $\alpha^* \leq \exp[-(\frac{n}{a})^{\frac{1}{b}}]$.

By Lemma 6, $\mathcal{R}_n \leq \frac{N}{\gamma}\alpha^*$ which immediately leads to the desired results. □

Proofs for values of β in special cases

Proof of Proposition 2 (uniform case). We study the size of S_ε. Due to the equal rewards all the policies are optimal, and condition (ii) in (5) does not eliminate any nodes. The contribution of a node is $c(s) = P(s)\frac{\gamma^{d(s)}}{1-\gamma} = (\frac{\gamma}{N})^{d(s)}\frac{1}{1-\gamma}$ since the probability of reaching a node at depth $d(s)$ is $(\frac{1}{N})^{d(s)}$. This also means that, for any policy h, the tree $\mathcal{T}_{hs}$ consists of all the nodes s' up to the depth of s. The number of leaves of this tree is $N^{d(s)}$ (recall that a policy tree has only branching factor N), and since this number does not depend on the policy, $n(s)$ is also $N^{d(s)}$. Therefore, $\alpha(s) = n(s)c(s) = \frac{\gamma^{d(s)}}{1-\gamma}$ and condition (i) eliminates nodes with depths larger than $D = \frac{\log \varepsilon(1-\gamma)}{\log \gamma}$. The remaining nodes in the whole tree, with branching factor NK, form S_ε, which is of size:

$$|S_\varepsilon| = O((NK)^D) = O((NK)^{\frac{\log \varepsilon(1-\gamma)}{\log \gamma}}) = O(\varepsilon^{-\frac{\log NK}{\log 1/\gamma}})$$

yielding for β the value: $\beta_{\text{unif}} = \frac{\log NK}{\log 1/\gamma}$. So, for large n the regret $\mathcal{R}_n = \tilde{O}(n^{-\frac{\log 1/\gamma}{\log NK}})$. In fact, as can be easily checked by examining the proof of Theorem 1, the logarithmic component disappears in this case and $\mathcal{R}_n = O(n^{-\frac{\log 1/\gamma}{\log NK}})$. □

Proof of Proposition 3 (structured rewards). Since $\alpha(s)$ depends only on the probabilities, condition (i) leads to the same $D = \frac{\log \varepsilon(1-\gamma)}{\log \gamma}$ as in the uniform case. However, now condition (ii) becomes important, so to obtain the size of S_ε, we must only count *near-optimal* nodes up to depth D.

Consider the set of nodes in $\mathcal{T}_\infty$ which do not belong to the optimal policy, but lie below nodes that are at depth d' on this policy. An example is enclosed by a dashed line in Figure 3, where $d' = 1$. All these nodes are sub-optimal to the extent of the loss incurred by not choosing the optimal action at their parent, namely: $(\frac{\gamma}{N})^{d'}\frac{1}{1-\gamma}$. Note these nodes *do* belong to a policy that is near-optimal to this extent, one which makes the optimal choices everywhere except at their parent. Looking now from the perspective of a given depth d, for any $m \leq d$ there are $N^d K^m$ nodes at this depth that are $(\frac{\gamma}{N})^{d-m}\frac{1}{1-\gamma}$-optimal. Condition (ii), written $(\frac{\gamma}{N})^{d-m}\frac{1}{1-\gamma} \leq \frac{N}{\gamma}\frac{\gamma^d}{1-\gamma}$, leads to $m \leq d\frac{\log N}{\log N/\gamma} + 1$. Then:

$$|S_\varepsilon| \leq \sum_{d=0}^{D} N^d K^{d\frac{\log N}{\log N/\gamma}+1} \leq K \sum_{d=0}^{D} \left(NK^{\frac{\log N}{\log N/\gamma}}\right)^d$$

If $N > 1$:

$$\begin{aligned}|S_\varepsilon| &= O\big((NK^{\frac{\log N}{\log N/\gamma}})^D\big) = O\big((NK^{\frac{\log N}{\log N/\gamma}})^{\frac{\log \varepsilon(1-\gamma)}{\log \gamma}}\big) \\ &= O\big(\varepsilon^{-\frac{\log N}{\log 1/\gamma}(1+\frac{\log K}{\log N/\gamma})}\big)\end{aligned}$$

yielding the desired value of $\beta_{\text{rew}} = \frac{\log N}{\log 1/\gamma}(1+\frac{\log K}{\log N/\gamma})$.

If $N = 1$ (deterministic case), $\beta_{\text{rew}} = 0$ and:

$$\begin{aligned}|S_\varepsilon| &= \sum_{d=0}^{D} 1 \cdot K = (D+1)K = \left(\frac{\log \varepsilon(1-\gamma)}{\log \gamma} + 1\right) K \\ &\leq a \log 1/\varepsilon\end{aligned}$$

for small ε and some constant a, which is of the form (6) for $b = 1$. From Theorem 1, the regret is $O(\exp(-\frac{n}{a}))$. □

Proof of Proposition 4 (structured probabilities). We will show that the quantities of nodes with sizable contributions on the subtree of one policy, and respectively on the whole tree, satisfy:

$$n(\lambda) = |\{s \in \mathcal{T}_\infty \mid c(s) \geq \lambda\}| = \tilde{O}(\lambda^{-\delta})$$
$$n_h(\lambda) = |\{s \in \mathcal{T}_h \mid c(s) \geq \lambda\}| = \tilde{O}(\lambda^{-\delta_h})$$

for constants δ_h and δ; and we will find values for these constants. (Note $n_h(\lambda)$ is not a function of h, since all policies have the same probability structure.) Then, since condition (ii) always holds and nodes in S_ε only have to satisfy condition (i):

$$\begin{aligned}|S_\varepsilon| &= |\{s \in \mathcal{T}_\infty \mid n(s)c(s) \geq \varepsilon\}| \\ &\leq |\{s \in \mathcal{T}_\infty \mid n_h(c(s))c(s) \geq \varepsilon\}| \\ &\leq \left|\left\{s \in \mathcal{T}_\infty \;\middle|\; a[\log 1/c(s)]^b c(s)^{1-\delta_h} \geq \varepsilon\right\}\right| \\ &= \tilde{O}(\varepsilon^{-\frac{\delta}{1-\delta_h}})\end{aligned}$$

where we used $n(s) \leq n_h(c(s))$ and $n_h(c(s)) = \tilde{O}(c(s)^{-\delta_h})$. Thus $\beta = \frac{\delta}{1-\delta_h}$.

Consider now $n_h(\lambda)$. The nodes at each depth d correspond to a binomial distribution with d trials, so there are C_d^m nodes with contribution $c(s) = p^{d-m}(1-p)^m \frac{\gamma^d}{1-\gamma}$, for $m = 0, 1, \ldots, d$. Since these contributions decrease monotonically with d, as well as with m at a certain depth, condition $c(x) \geq \lambda$ eliminates all nodes above a certain maximum depth D, as well as at every depth d all nodes above a certain $m(d)$, where:

$$\frac{(p\gamma)^d}{1-\gamma} \geq \lambda \quad \Rightarrow \quad d \leq \frac{\log 1/(\lambda(1-\gamma))}{\log 1/(p\gamma)} = D$$
$$m \leq \frac{\log 1/(\lambda(1-\gamma))}{\log p/(1-p)} - d\frac{\log 1/(p\gamma)}{\log p/(1-p)} = m(d)$$

Note in the condition for D we set $m = 0$ to obtain the largest probability. So, $m(d)$ decreases linearly with d, so that up to some depth m^*, $m(d) \geq d$ and we count all the nodes up to $m = d$; while above m^*, $m(d) < d$ and we count fewer nodes. The depth m^* is obtained by solving $m(d) = d$, leading to $m^* = \frac{\log 1/(\lambda(1-\gamma))}{\log 1/(\gamma(1-p))} = \frac{\log 1/(p\gamma)}{\log 1/(\gamma(1-p))} D = \eta D$ with the notation $\eta = \frac{\log 1/(p\gamma)}{\log 1/(\gamma(1-p))}$. The structure of the subtree satisfying $c(s) \geq \lambda$ is represented in Figure 5.

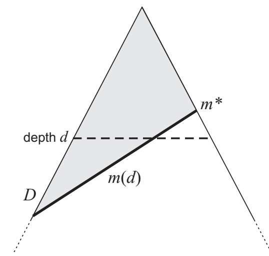

Figure 5: Schematic representation of the subtree satisfying $c(s) \geq \lambda$, shown in gray. Nodes with larger probabilities are put to the left. The thick line represents the fringe $m(d)$ where nodes stop being counted.

Now:

$$\begin{aligned}n_h(\lambda) &= \sum_{d=0}^{D} \sum_{m=0}^{\min\{m(d),d\}} C_d^m \leq \sum_{d=0}^{D} \sum_{m=0}^{\min\{m(d),d\}} \left(\frac{de}{m}\right)^m \\ &\leq \sum_{d=0}^{D} \sum_{m=0}^{m^*} \left(\frac{De}{m^*}\right)^{m^*} = Dm^* \left(\frac{De}{m^*}\right)^{m^*} \\ &= \eta D^2 \left(\frac{e}{\eta}\right)^{\eta D} = \tilde{O}(\left(\frac{e}{\eta}\right)^{\eta D})\end{aligned}$$

where we used $C_d^m \leq \left(\frac{de}{m}\right)^m$ as well as $\left(\frac{de}{m}\right)^m \leq \left(\frac{De}{m}\right)^m \leq \left(\frac{De}{m^*}\right)^{m^*}$. The latter inequality can be shown by noticing that $\left(\frac{De}{m}\right)^m$, as a function of m, increases up to $m = D$, and $m^* \leq D$ is on the increasing part. Denoting now $\eta' = \left(\frac{e}{\eta}\right)^\eta$ and continuing:

$$n_h(\lambda) = \tilde{O}(\eta'^D) = \tilde{O}(\eta'^{\frac{\log 1/(\lambda(1-\gamma))}{\log 1/(p\gamma)}}) = \tilde{O}(\lambda^{-\frac{\log \eta'}{\log 1/(p\gamma)}})$$

leading to the value for $\delta_h = \frac{\log \eta'}{\log 1/(p\gamma)}$.[3]

Similarly, it is shown that $n(\lambda) = \tilde{O}(\lambda^{-\frac{\log K\eta'}{\log 1/(p\gamma)}})$ and thus $\delta = \frac{\log K\eta'}{\log 1/(p\gamma)}$, where the extra K comes from the fact we count the nodes corresponding to all K^d policies rather than just one.

The desired result is immediate: $\beta_{\text{prob}} = \frac{\delta}{1-\delta_h} = \frac{\log K\eta'}{\log 1/(p\gamma\eta')}$. Note throughout, we silently used the fact that p is close to 1; indeed, this is required for some of the steps to be meaningful, such as having $\log 1/(p\gamma\eta') > 0$. □

[3] The definition of $n(s)$ in fact only requires counting the *leaves* of the subtree corresponding to $n_h(\lambda)$ (thick line in Figure 5), while we counted all the nodes (gray area). Exploiting this property is unlikely to be helpful, however, since in the upper bound derived for $n_h(\lambda)$ the inner term in the sum (corresponding to C_d^m, the number of nodes having a certain probability) is dominant. The fact that the whole tree is taken into account only enters the logarithmic component of the bound.