HOW TO CHOOSE THE STATE RELEVANCE WEIGHT OF THE APPROXIMATE LINEAR PROGRAM?

YANN LE TALLEC AND THEOPHANE WEBER

Abstract. The linear programming approach to approximate dynamic programming was introduced in [1]. Whereas the state relevance weight (i.e. the cost vector) of the linear program does not matter for exact dynamic programming, it is not the case for approximate dynamic programming. In this paper, we address the issue of selecting an appropriate state relevant weight in the case of approximate dynamic programming. In particular, we want to choose c so that there is a practical control of the approximate policy performance by the capability of the approximation architecture. We present here some theoretical results and more practical guidelines to select a good state relevance vector.

1. Introduction

The linear programming approach to approximate dynamic programming was introduced in [1], and it is reviewed quickly in Section 2. Whereas the state relevance weight (i.e. the cost vector) of the linear program does not matter for exact dynamic programming, it is not the case for approximate dynamic programming. There are no guidelines in the literature to select an appropriate state relevant weight in the case of approximate dynamic programming. In Section 3, we propose to use available performance bounds on the suboptimal policy based on the approximate linear program to build a criterion for choosing the state relevance weight c. We characterize appropriate state relevance weights as solutions of an optimization problem (P). However, (P) cannot be solved easily so that we look for suboptimal solutions in Section 4, in particular we prove in Section 5 that under some technical assumptions we can choose c as a probability distribution. Finally, we establish some practical necessary conditions to choose c; one of them suggesting to reinforce the linear program for approximate dynamic programming.

1.1. **Finite Markov decision process framework.**

In this paper, we consider finite Markov decision process (MDP): they have a finite state space $\mathcal{S}$ and a finite control space $U(x)$ for each state x in $\mathcal{S}$. Let $g_u(x)$ be the expected immediate cost of applying control u in state x. $P_u(x,y)$ denotes the transition probability from state x to state y under control $u \in U(x)$. The objective of the controller is to minimize the α-discounted cost $E\left[\sum_{t\geq 0} \alpha^t g_{u(t)}(x_t)|x_0\right]$.

First, observe that it is possible to transform any finite Markov decision process with finitely many controls in another one where the immediate cost of an action is the same for all actions. Indeed, consider the MDP comprising the original MDP states plus one state for each state-action pair. In this MDP, the controller first chooses a control and the system moves in the corresponding state-action pair.

Date: May 15, 2004.

From there, the system incurs the cost corresponding to the state-action pair and follows the original dynamics to the next state. Figure 1 provides a simple example of the transformation of an MDP into another one with same immediate cost at each state.

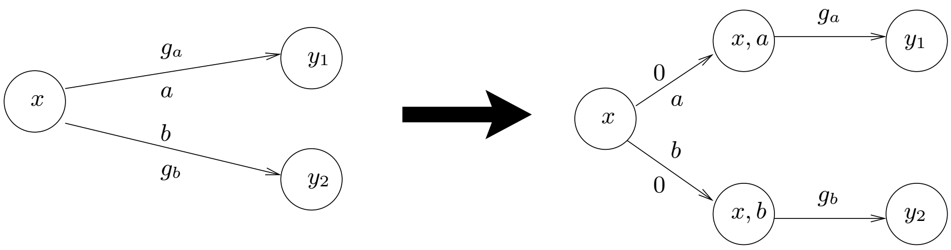

FIGURE 1. A simple example of an MDP transformation. The original MDP starts from state x and moves to state y_1 and incurs a cost g_a if a is chosen, or moves to y_2 and incurs g_b if the other action, b, is chosen.

As a result, we can make without loss of generality the following assumption.

Assumption 1.1. *For all x in $\mathcal{S}$, $g_u(x) = g(x)$ is independent of the control $u \in U(x)$. Furthermore, we can assume $g(x) \geq 0$, $\forall x \in \mathcal{S}$.*

2. The linear programming approach for dynamic programming

Let T be the usual dynamic programming operator for discounted problem: $TJ(x) := \min_{u \in U(x)}\{g(x)+\alpha \sum_{y \in \mathcal{S}} P_u(x,y)J(y)\}$. It is well-known that T is monotonic ($J \leq J' \Rightarrow TJ \leq TJ'$) and that for all J,

$$\lim_{k \to +\infty} T^k J = J^*,$$

where J^* is the optimal cost-to-go vector. Moreover, J^* is the unique solution to Bellman equations $J^* = TJ^*$. As a result, if $J \leq TJ$, then $J \leq TJ \leq T^k J \leq J^*$. In other words, J^* is the biggest vector of $\mathbb{R}^{|\mathcal{S}|}$ verifying $J \leq TJ$. The following proposition, which states that J^* can be computed by a linear program, is an immediate consequence of this remark

Proposition 2.1. *For all $c > 0$, J^* is the unique optimal solution to the following linear program*

$$(LP): \qquad \min_{J \in \mathbb{R}^{|\mathcal{S}|}} c^T J$$
$$J(x) \leq g(x) + \alpha \sum_{y \in \mathcal{S}} P_u(x,y)J(y),\ \forall x \in \mathcal{S},\ \forall u \in U(x)$$

Unfortunately, the linear program (LP) is often enormous with as many variables as the cardinality of the state space $\mathcal{S}$ and as many constraints as there are state-action pairs. Hence, (LP) is often intractable. Moreover, even storing a cost-to-go vector J as a lookup table might not be amenable for large state space.

One approach to deal with this curse of dimensionality is to approximate $J^*(x) \approx \Phi(x)r$, where $r \in \mathbb{R}^m$ (usually $m \ll |\mathcal{S}|$) and $\Phi(x) = (\phi_1(x), \ldots, \phi_m(x))$ are given feature vectors.

Inspired by the form of (LP), it is natural to consider the approximation of $J^* \approx \Phi\tilde{r}(c)$, where $\tilde{r}(c)$ is an optimal solution of the approximate linear program:

$$(ALP): \qquad \min_{r \in \mathbb{R}^m} c^T \Phi r$$

$$\Phi(x)r \leq g(x) + \alpha \sum_{y \in \mathcal{S}} P_u(x,y)\Phi(y)r, \; \forall x \in \mathcal{S}, \; u \in U(x)$$

Notice that (ALP) has only m variables, but still as many constraints as (LP). Hence, some large scale optimization technics are needed to solve (ALP), or alternatively [3] showed that constraints sampling is a viable approach to solve this problem.

On the contrary to the case of the "exact" linear program (LP), there is no guarantee that the optimal solution $\tilde{r}(c)$ is independent of the choice of $c > 0$. The objective of this paper is to provide a methodology to choose c, but to motivate our criterion to select c, we first need to introduce two performance bounds.

3. Two performance bounds

3.1. A general performance bound.

First, let us relate J^* to the cost-to-go of the policy, which is greedy with respect to J, where J is any approximation of J^*.

Let $J \in \mathbb{R}^{|\mathcal{S}|}$ and u_J be the greedy policy with respect to J, i.e.

$$u_J(x) = argmin_{u \in U(x)} \left\{ g_u(x) + \alpha \sum_{y \in \mathcal{S}} P_u(x,y)J(y) \right\}.$$

In the case of equality between controls $\{u_1, \ldots, u_q\}$, then u_J chooses randomly one of them with equal probability $1/q$.

Theorem 3.1 (Theorem 3.1 in [1])**.** *For all $J \in \mathbb{R}^{|\mathcal{S}|}$ such that $TJ \geq J$,*

$$\|J_{u_J} - J^*\|_{1,\nu} \leq \frac{1}{1-\alpha}\|J - J^*\|_{1,\mu_{\nu,u_J}}, \tag{3.1}$$

where $\mu_{\nu,u_J} := (1-\alpha)\nu^T(I - \alpha P_{u_J})^{-1}$ and $P_{u_J}(x,y)$ denotes the probability of the system going to state y under the policy u_J given it is in state x.

Notice that $\mu_{\nu,u}$ is well-defined even for some randomized policy u.

Hence, if J is a good approximation of J^* in the sense of a certain weighted l_1-norm, a policy greedy with respect to J will perform closely to optimal.

3.2. Approximate linear program approximation bound.

Here we reproduce a bound from [1], which says that solutions to (ALP) produce approximations to the optimal cost-to-go function J^* that are close to the best possible approximation within the architecture that is linear in Φ.

Theorem 3.2 (Theorem 4.2 in [1])**.** *Let $\tilde{r}(c)$ be an optimal solution of the approximate linear program with the state relevance weight c. Then for any $v \in \mathbb{R}^m$ such that $\Phi(x)v > 0$ and $\alpha \max_{u \in U(x)} \sum_{y \in \mathcal{S}} P_u(x,y)\Phi(y)v < \Phi(x)v$ for all $x \in \mathcal{S}$,*

(3.2)

$$\|J^* - \Phi\tilde{r}(c)\|_{1,c} \leq \frac{2c^T\Phi v}{1-\beta_{\Phi v}} \min_{r \in \mathbb{R}^m} \|J^* - \Phi r\|_{\infty,1/\Phi v} = \frac{2c^T\Phi v}{1-\beta_{\Phi v}}\|J^* - \Phi r^*\|_{\infty,1/\Phi v},$$

where r^* be the projection of J^* on the surface $\{\Phi r | r \in \mathbb{R}^m\}$ with respect to the norm $\|.\|_{\infty,1/\Phi v}$.

The greedy policy with respect to $\Phi\tilde{r}(c)$, $u_{\Phi\tilde{r}(c)}$, is called the ALP policy associated with the state relevance weight c.

3.3. **Proposed criterion to choose the state relevance weight c.**

We would like to link the two bounds (3.1) and (3.2) by controlling $\|J^* - \Phi\tilde{r}(c)\|_{1,\mu_{\nu,u_{\Phi\tilde{r}(c)}}}$ with $\|J^* - \Phi\tilde{r}(c)\|_{1,c}$ in order to bound the performance loss of the ALP policy by the architecture capability to approximate J^*. Hence, we would like to find a state relevance weight $c > 0$ that makes this guarantee as sharp as possible.

In other words, the state relevance weight should be chosen so that

$$\begin{aligned} (P): \qquad & \min_{c>0} c^T \Phi v \\ & \mu^T_{\nu,u_{\Phi\tilde{r}(c)}} := (1-\alpha)\nu^T (I - \alpha P_{u_{\Phi\tilde{r}(c)}})^{-1} \leq c^T, \end{aligned}$$

where $u_{\Phi\tilde{r}(c)}$ depends on c.

Then, for any feasible c of (P), we can write

$$\|J_{u_{\Phi\tilde{r}(c)}} - J^*\|_{1,\nu} \leq \frac{2c^T\Phi v}{1-\beta_{\Phi v}} \|J^* - \Phi r^*\|_{\infty,1/\Phi v}, \tag{3.3}$$

by combining the bounds (3.1) and (3.2), and (P) tries to make the factor of the right-hand side as small as possible.

Recall that $\tilde{r}(c)$ depends on c so that we have a circular dependence between c and $\tilde{r}(c)$.

$$\begin{tikzcd} c \arrow[r] & \tilde{r}(c) \arrow[d] \\ \mu_{\nu,u_{\Phi\tilde{r}(c)}} \arrow[u] & u_{\Phi\tilde{r}(c)} \arrow[l] \end{tikzcd}$$

(P) is a difficult problem because of the complex constraint $(1-\alpha)\nu^T(I - \alpha P_{u_{\Phi\tilde{r}(c)}})^{-1} \leq c^T$. In this paper, we try only to obtain feasible points of (P). Still, there is a special case of ν where (P) can be solved exactly.

Proposition 3.3. *If $\nu \geq 0$ happens to be chosen as the steady-state probability distribution of $P_{u_{\Phi\tilde{r}(c)}}$ (when it exists), i.e. $\nu^T = \nu^T P_{u_{\Phi\tilde{r}(c)}}$, (**??**) yields*

$$c^T \geq \nu^T.$$

Proof. Indeed, let $x \neq 0$ be a left eigenvector of an invertible matrix A associated with the eigenvalue λ ($\lambda \neq 0$).

$$\begin{aligned} & x^T A = \lambda x^T \\ \Leftrightarrow \quad & x^T A^{-1} = \lambda^{-1} x^T \end{aligned}$$

If $\nu^T = \nu^T P_{u_{\Phi\tilde{r}(c)}}$, then $\nu^T = \frac{1}{1-\alpha}\nu^T(I - \alpha P_{u_{\Phi\tilde{r}(c)}})$. Hence, $\nu^T = (1-\alpha)\nu^T(I - \alpha P_{u_{\Phi\tilde{r}(c)}})^{-1} \leq c^T$, and the optimal solution of (P) is $c = \nu$. □

In the following section, we give derive some simple feasible points but their performance with respect to the objective of (P) can be very poor. Then in Section 5, we try to obtain better feasible points of (P), namely probability distributions.

4. Two simple choices for c

4.1. A trivial bound.

It is well-known [1] that $\mu_{\nu,u}$ is a probability distribution over $\mathcal{S}$ for all initial distributions ν and all policies u in the sense that $0 \leq \mu_{\nu,u}(x) \leq 1,\ \forall x \in \mathcal{S}$ and $\mu_{\nu,u}^T \mathbf{1} = 1$. Moreover, by definition of weighted l_1-norms, $c \geq \mu_{\nu,u} \geq 0 \Rightarrow \| \bullet \|_{1,c} \geq \| \bullet \|_{1,\mu_{\nu,u}}$ so that we have the following proposition.

Proposition 4.1. *By choosing $c^T = (1, \ldots, 1) = \mathbf{1}$ in (ALP), we get the bound*

$$\|J_{u_{\Phi\tilde{r}(c)}} - J^*\|_{1,\nu} \leq \frac{2\mathbf{1}^T \Phi v}{1 - \beta_{\Phi v}} \|J^* - \Phi r^*\|_{\infty,1/\Phi v}$$

This bound is poor. c does not depend on the problem characteristics. Furthermore, there is a factor $\sum_{x\in\mathcal{S}} \Phi(x) v$ on the right-hand side, which becomes very large in large scale system or in system where some states have a high value for the Lyapunov function.

4.2. A simple algorithm.

Notice that if $c > 0$ is scaled by a positive factor $\gamma > 0$ the optimal solutions of (ALP) are unchanged. This remark allows us to devise the following scheme.

(1) Pick any $c > 0$ and find an optimal solution $\tilde{r}(c)$ of (ALP)
(2) Given ν, compute $\mu_{\nu,u_{\Phi\tilde{r}(c)}}$.
(3) Let $\gamma > 0$ be the smallest scalar such that $0 \leq \mu_{\nu,u_{\Phi\tilde{r}(c)}} \leq \gamma c$.

If $\gamma < +\infty$,

$$\|J_{u_{\Phi\tilde{r}(c)}} - J^*\|_{1,\nu} \leq \frac{2\gamma c^T \Phi v}{1 - \beta_{\Phi v}} \|J^* - \Phi r^*\|_{\infty,1/\Phi v}.$$

Unfortunately, there is no guarantee that γ will be small, even when it is finite, so that the bound is practical.

5. Finding probability distribution feasible for (P)

If the state relevance weight c of the ALP could be chosen such that

$$c^T = \mu_{\nu,u_{\Phi\tilde{r}(c)}} := (1-\alpha)\nu^T (I - \alpha P_{u_{\Phi\tilde{r}(c)}})^{-1}, \tag{5.1}$$

(3.3) would hold. Indeed, c verifies (5.1) if and only if c is a probability distribution that is feasible for (P). We hope that in this case, the bound (3.3) is practical.

5.1. A theoretical algorithm.

5.1.1. *A naive algorithm.*

A naive algorithm is as follows.

Algorithm A

(1) Start with $k = 0$ and any vector $c_0 \geq 0$ such that $\sum_{x\in\mathcal{S}} c(x) = 1$.
(2) Solve (ALP) for c_k and let $\tilde{r}(c_k)$ be any optimal solution.
(3) Compute $\mu_k := \mu_{\nu,u_k}$, where $u_k := u_{\Phi\tilde{r}(c_k)}$ is greedy with respect to $\Phi\tilde{r}(c_k)$.
(4) Set $c_{k+1} = \mu_k$, do $k = k+1$ and go back to 2

Equivalently, the algorithm may be represented by

$c \overset{\tilde{r}}{\to} r(c) \overset{F}{\to} \Phi r(c) \overset{U}{\to} u_{\Phi r(c)} \overset{M}{\to} \mu_{\nu,u_{\Phi r(c)}}$, or, in a more compact fashion: $c \overset{M}{\to} \mu_{\nu,u_{\Phi r(c)}}$.

Definition 5.1. Define $P = \{p \in \mathbb{R}^{|\mathcal{S}|} |\ p(x) \geq 0,\ \sum_{x \in \mathcal{S}} p(x) = 1\}$ as the space of probabilities distributions. It is a compact, convex set.

Notice that μ is a mapping from P in itself, where P is the set of probability distribution over $\mathcal{S}$, i.e.

Lemma 5.2. *c is a fixed point of $\mu \iff c$ verifies (5.1)*

However, it is not clear that the algorithm A has a fixed point, and whether c_k converges. If the mapping μ was continuous from P to P, Brouwer's theorem would guarantee the existence of a fixed point.

However, in the chain $c \xrightarrow{\tilde{r}} r(c) \xrightarrow{F} \Phi r(c) \xrightarrow{U} u_{\Phi r(c)} \xrightarrow{M} \mu_{\nu, u_{\Phi r(c)}}$, some of the functions may not be continuous so that M needs not be continuous.

- The function F is just a matrix multiplication. Therefore it is continuous.
- The function $M\ :\ \ (P_u, g_u) \rightarrow \mu(u) = (1-\alpha)\nu^\top(I - \alpha P_u)^{-1} = (1-\alpha)\nu^\top \sum\limits_t \alpha^t P_u^t$ is also continuous.
- However, it is well-known that the functions $\tilde{r}$ and U are not necessarily continuous.

In the following part, we define a randomized version of A making $\tilde{r}$ and U continuous so that Brouwer's theorem will guarantee the existence of a fixed point to the randomized algorithm.

5.1.2. *A randomized algorithm.*

Defining and smoothing $\tilde{r}$

First, we show that (ALP) has an optimal solution thanks to the following lemma.

Lemma 5.3. *The feasible set of (ALP), $\{r \in \mathbb{R}^m |\ \Phi r \leq T \Phi r\}$ is nonempty and bounded.*

Proof. Since we assumed $g(x) \geq 0$, the feasible set contains $r = 0$, and is thus nonempty.

The matrix Φ is full rank, hence $\Phi'\Phi$ is invertible (because it is symmetric definite positive).

Therefore, $(\Phi'\Phi)^{-1}$ has a maximum norm which is denoted $M_1 = \|(\Phi'\Phi)^{-1}\|_\infty$ For all r, $\|(\Phi'\Phi)^{-1} r\|_\infty \leq \|(\Phi'\Phi)^{-1}\|_\infty \|r\|_\infty$, and using this property with $r' = (\Phi'\Phi) r$, we get $\|r\|_\infty \leq M_1 . \|(\Phi'\Phi) r\|_\infty$

Now, the matrix Φ' also has some maximum norm M_2, and using sub-multiplicative property, we get $\|r\|_\infty \leq M_1 . M_2 \|\Phi r\|_\infty$

If we consider r feasible for the ALP, we have $\Phi r \leq T \Phi r \leq T^2 \Phi r \leq .. \leq J^*$ which gives : $\|\Phi r\|_\infty \leq \|J^*\|_\infty$.

Combining the two inequalities yield $\|r\|_\infty \leq M_1 . M_2 \|J^*\|_\infty$ □

Hence, from the theory of Linear Programming, there is always a solution of the LP that is an extreme point of the feasible set. Usually, there is a unique optimal solution to a linear program with a cost vector c, and it is an extreme point of the feasible polyhedron. In this case, $\tilde{r}(c)$ is clearly defined. When there are multiple optimal solutions, we can define $\tilde{r}$ arbitrarily because we will see that it happens with probability 0 in our algorithm. Furthermore, it can be showed that the function $\tilde{r}$ defined above is piecewise constant [5].

The following lemma is well-known.

Lemma 5.4. *Let f be a bounded piecewise constant mapping from some vector space E to F. Let g be a continuous function from F to $\mathbb{R}$ that has finite integral.*

Then, the function f' : $x \to \int\limits_F f(x+y).g(y)dy$ is a well defined, continuous function from E to F.

f' is a smoothed version of the initial piecewise constant mapping f.

This lemma suggests to randomize the cost vector c with some noise in order to smooth $\tilde{r}$.

Proposition 5.5. *Let $\widetilde{c}$ be a random vector defined by $\widetilde{c} = c + \delta c$, where δc is a Gaussian vector, which covariance matrix C is equal to v.I. Then, the function $c \to E[\tilde{r}[\widetilde{c}]]$ is continuous in c.*

Proof. $c \overset{\tilde{r}}{\to} E[\tilde{r}[\widetilde{c}]]$ can also be written $c \to \int\limits_{\mathbb{R}^N} \tilde{r}(c+c_0).g(c_0)dc_0$ and $\tilde{r}$ is a piecewise constant, bounded function. Using the previous lemma gets the result. □

Smoothing U

The function U is also discontinuous, in the same fashion as the function r. We therefore use the same "trick", but in a slightly different way. Instead of using deterministic greedy policies, we use randomized, δ-greedy policies

Definition 5.6. Let $\delta > 0$. The δ -greedy policy with respect to J is a randomized policy u_δ for which the action a is chosen in state x with probability

$$u_J^\delta(u,x) = \frac{exp[-(g_u(x) + \alpha P_u(x)J)/\delta]}{\sum_{a \in U(x)} exp[-(g_a(x) + \alpha P_a(x)J)/\delta]}.$$

[4] provides various continuity results for the $\delta - greedy$ policies, which we will use.

Proposition 5.7. $\underset{\delta \downarrow 0}{\lim} \underset{J,x}{\sup} |T_\delta J(x) - TJ(x)| = 0$

This proposition states the fact that T_δ approaches uniformly T as $\delta \downarrow 0$.

Proposition 5.8. *T_δ and u_δ are continuous in J.*

Randomized algorithm $A(v,\delta)$

We now define the randomized version of the algorithm A.

Definition 5.9. The randomized function $\mu(v,\delta)$ is defined by the following chain of functions:

$c \overset{\tilde{r}}{\to} \widetilde{r}(c) \overset{F}{\to} \Phi\widetilde{r}(c) \to u_\delta(\Phi\widetilde{r}(c)) \overset{\mu(v,\delta)}{\to} \mu_{\nu,u_\delta(\Phi\widetilde{r}(c))}$, or, in a compact fashion: $c \overset{\mu(v,\delta)}{\to} \mu_{\nu,u_\delta(\Phi\widetilde{r}(c))}$

Proposition 5.10. *$\mu(v,\delta)$ is continuous from P to P.*

Definition 5.11. Let v and δ be some positive numbers. The algorithm $A(v,\delta)$ is:

1) Start from some c_0 in P, and set $k = 0$.

2) Do $c_{k+1} = \mu(v,\delta)\ (c_k)$

3) Set $k = k+1$ and go to 2.

Theorem 5.12. *$\mu(v,\delta)$ has at least one fixed point $c(v,\delta) \in P$*

Proof. $\mu(v,\delta)$ is a continuous function on a compact, convex set. By application of Brouwer's theorem, it has a fixed point. □

Remark 5.13. Saying that $\mu(v,\delta)$ has at least one fixed point is equivalent to saying that $A(v,\delta)$ has at least one fixed point

However, the c_k produced by the algorithm may still fail to converge so that $A(v,\delta)$ does not provide the value of a fixed point.

5.1.3. *Existence of fixed point for the original algorithm A.*. In this part, we will use the previous theorem asserting the existence of a fixed point to the algorithm that holds for all variance $v > 0$ and all $\delta > 0$ to show that there exists a fixed point to the original algorithm A.

Theorem 5.14. *For any pair (v_k, δ_k) in $\mathbb{R}^2$ with $(v_k,\delta_k) > 0$, denote C_k the set of fixed points of the algorithm $A(v_k,\delta_k)$, which is not empty by Theorem 5.12.*

If there is a sequence $(v_k,\delta_k)_{k\geq 0}$ of such pairs with $(v_k,\delta_k) \to (0,0)$, such that there is an accumulation point c of the set C_k that yields a unique optimum if used as a state relevance vector in (ALP), then $c \geq 0$ is a probability distribution that verifies

$$c^T = \mu_{\nu, u_{\Phi\tilde{r}(c)}} := (1-\alpha)\nu^T(I - \alpha P_{u_{\Phi\tilde{r}(c)}})^{-1} \tag{5.2}$$

Proof. Without loss of generality, let $c_k \in C_k$ such that $lim_{k\to+\infty} c_k = c$. By definition,

$$c_k = \mu_{\nu, u^{\delta_k}_{\Phi\tilde{r}(c_k)}} == (1-\alpha)\nu^T(I - \alpha P_{u^{\delta_k}_{\Phi\tilde{r}(c_k)}})^{-1}. \tag{5.3}$$

Note $\Pi \in \mathbb{R}^m$ the polyhedron that is the feasible set of (ALP). By assumption, there is a unique $\tilde{r}(c)$ verifying $\tilde{r}(c) \in \Pi$ (ALP feasibility) and $c^T\Phi\tilde{r}(c) > c^T\Phi r$ for all $r \in \Pi$. Hence, $\tilde{r}(c)$ stays the unique optimal solution of (ALP) for state relevance weight close enough to c. Since $c_k \to c$, there is K such that $k \geq K \Rightarrow \tilde{r}(c_k) = \tilde{r}(c)$. In particular, $u_{\Phi\tilde{r}(c)} = u_{\Phi\tilde{r}(c_k)}$, $\forall k \geq K$, and (5.3) becomes for $k \geq K$

$$c_k = (1-\alpha)\nu^T(I - \alpha P_{u^{\delta_k}_{\Phi\tilde{r}(c)}})^{-1} \tag{5.4}$$

Recall that a δ-greedy policy u^δ_J with respect to $J \in \mathbb{R}^{|\mathcal{S}|}$ chooses control u in state x with probability

$$u^\delta_J(u,x) = \frac{exp[-(g_u(x) + \alpha P_u(x)J)/\delta]}{\sum_{a\in U(x)} exp[-(g_a(x) + \alpha P_a(x)J)/\delta]}. \tag{5.5}$$

Lemma 5.15. *Assume that $\underline{U} = \{u_1,\ldots,u_q\} \subset U(x)$ is the set of minimizers of $g_u(x) + \alpha P_u(x)J$. Then,*

$$\lim_{\delta\downarrow 0} u^\delta_J(a,x) = u_J(a,x) = \begin{cases} 1/q \; if \; a \in \underline{U} \\ 0 \; otherwise \end{cases}$$

For $J = \Phi\tilde{r}(c)$, the lemma yields $\lim_{k\to+\infty} u^{\delta_k}_{\Phi\tilde{r}(c)} = u_{\Phi\tilde{r}(c)}$.

Combining this results with (5.4), we have

$$c = \lim_{k\to+\infty} c_k = \lim_{k\to+\infty} (1-\alpha)\nu^T(I - \alpha P_{u^{\delta_k}_{\Phi\tilde{r}(c)}})^{-1} = (1-\alpha)\nu^T(I-\alpha P_{u_{\Phi\tilde{r}(c)}})^{-1}. \tag{5.6}$$

□

5.2. **Necessary conditions.**

Although the theoretical algorithm presented above shows the existence of a probability distribution in the feasible set of (P), it is not practical. Now, we would like to obtain practical guidelines for the choice of c. In particular, we derive in this section some necessary conditions on the state relevance weight. One of them yields a reinforced approximate linear program.

5.2.1. *A condition on c depending on the Lyapunov function and the initial distribution.*

Proposition 5.16. *If c verifies (5.1), then*

$$\nu^T \Phi v \leq (1-\alpha)^{-1} c^T (I - \alpha P_{u_v}) \Phi v$$

Proof. Assume (5.1) holds, or equivalently

$$(1-\alpha)\nu^T = c^T (I - \alpha P_{u_{\Phi \tilde{r}(c)}}).$$

Then multiplying by Φv on the right and noting u_v the greedy policy with respect to the Lyapunov function Φv ($P_{u_v} \Phi v \leq P_u \Phi v, \ \forall u$, as we modified the Markov Chain so that all policies has the same cost vector), we have

$$\begin{aligned} \nu^T \Phi v &= (1-\alpha)^{-1} c^T (I - \alpha P_{u_{\Phi \tilde{r}(c)}}) \Phi v \\ \Rightarrow \nu^T \Phi v &\leq (1-\alpha)^{-1} c^T (I - \alpha P_{u_v}) \Phi v \end{aligned}$$

□

Notice that the spectrum of $(1-\alpha)^{-1}(I - \alpha P_{u_v})$ is of the form $(1-\alpha\lambda)/(1-\alpha)$, where λ is an eigenvalue of P_{u_v}.

5.2.2. *Reinforced approximate linear program.*

A possible approach to obtain (5.1) is to enforce this constraint in the ALP and hope there is a solution for a given c. That is to try to solve the following non-linear program:

$$(RANLP): \qquad \begin{aligned} &\max_{r \in \mathbb{R}^m} c^T \Phi r \\ &T\Phi r \geq \Phi r \\ &c^T = (1-\alpha)\nu^T (I - \alpha P_{u_{\Phi \tilde{r}(c)}}) \end{aligned}$$

The last constraints are hard to deal with, but we can derive more tractable necessary conditions. In particular, the next proposition shows that they imply a system of linear equations.

Proposition 5.17. *If c verifies (5.1), then the following system of linear equations holds*

$$(1-\alpha)\nu^T \Phi \tilde{r}(c) \geq c^T (I - \alpha P_u) \Phi R, \ \forall u \in U(x) \tag{5.7}$$

Proof. By definition of $u_{\Phi \tilde{r}(c)}$ given Assumption 1.1, we have

$$P_{u_{\Phi \tilde{r}(c)}} \Phi \tilde{r}(c) \leq P_u \Phi \tilde{r}(c), \ \forall u \in U(x) \tag{5.8}$$

Hence,

$$\begin{aligned} & -\alpha P_{u_{\Phi\tilde{r}(c)}}\Phi\tilde{r}(c) \geq -\alpha P_u \Phi\tilde{r}(c),\ \forall u \in U(x) \\ \Leftrightarrow\quad & (I-\alpha P_{u_{\Phi\tilde{r}(c)}})\Phi\tilde{r}(c) \geq (I-\alpha P_u)\Phi\tilde{r}(c),\ \forall u \in U(x) \\ \Leftrightarrow\quad & \Phi\tilde{r}(c) \geq (I-\alpha P_{u_{\Phi\tilde{r}(c)}})^{-1}(I-\alpha P_u)\Phi\tilde{r}(c),\ \forall u \in U(x) \end{aligned}$$

The last equivalence follows from $(I-\alpha P_{u_{\Phi\tilde{r}(c)}})^{-1} = \sum_{t\geq 0}\alpha^t P^t_{u_{\Phi\tilde{r}(c)}} \geq 0$. Multiplying both sides of the last equation by $(1-\alpha)\nu^T$,

$$(1-\alpha)\nu^T\Phi\tilde{r}(c) \geq \underbrace{(1-\alpha)\nu^T(I-\alpha P_{u_{\Phi\tilde{r}(c)}})^{-1}}_{\mu_{\nu,u_{\Phi\tilde{r}(c)}}}(I-\alpha P_u)\Phi\tilde{r}(c),\ \forall u \in U(x)$$

□

As a result, it is natural to consider a reinforced linear program (RALP) to approximate J^* by a linear combination of Φ.

$$\begin{aligned} (RALP):\qquad & \max_{r\in\mathbb{R}^m} c^T\Phi r \\ & T\Phi r \geq \Phi r \\ & (1-\alpha)\nu^T\Phi r \geq c^T(I-\alpha P_u)\Phi r,\ \forall u \in U(x) \end{aligned}$$

Notice that the last constraint enforces the equality $\mu_{\nu,u_{\Phi\tilde{r}(c)}} = c$ only on the subspace $\{(I-\alpha P_u)\Phi\tilde{r}(c),\ u \in U\}$, whereas we need this condition to hold for $J^*-\Phi\tilde{r}(c)$, $\tilde{r}(c)$ being an optimal solution of (RALP) so that $\|J^*-\Phi\tilde{r}(c)\|_{1,\mu_{\nu,\tilde{r}(c)}} = \|J^*-\Phi\tilde{r}(c)\|_{1,c}$.

6. Conclusion

We presented some new results for the choice of the state relevance weight c in the approximate linear program. The criterion to choose c hinges on two performance bounds that control the suboptimality of the ALP policy. However, these results remain preliminary, in particular how to tailor the state relevance weight to the problem setting remains an open question.

7. appendix

7.1. Insights on $\mu_{\nu,u}$.

By definition,

$$\mu^T_{\nu,u} := (1-\alpha)\nu^T(I-\alpha P_u)^{-1} = (1-\alpha)\sum_{t\geq 0}\alpha^t\nu^T P^t_u. \tag{7.1}$$

Hence, $\mu_{\nu,u}$ is a geometric average of the presence probability over the state space after t transitions under policy u starting from the distribution ν. When P_u irreducible, $lim_{t\to+\infty}\nu^T P^t_u = \pi^T_u$, where π_u is the steady-state distribution of P_u, i,e, $\pi^T_u = \pi^T_u P_u$. Thus we can wonder how far is π_u from $\mu_{\nu,u}$. We show now that in general $\mu_{\nu,u}$ is further away from π_u than ν.

Since π_u is also a left eigenvalue of $(1-\alpha)(I-\alpha P_u)^{-1}$, we have

$$\mu^T_{\nu,u} - \pi^T_u = (\nu-\pi_u)^T(1-\alpha)(I-\alpha P_u)^{-1}.$$

When P_u irreducible, the eigenvalue of P_u have a modulus smaller than one by Perron-Frobenius theorem. Hence, the eigenvalues of $(I - \alpha P_u)^{-1}$ have a modulus greater than 1. As a result, the previous equation yields

$$\|\mu_{\nu,u} - \pi_u\|_2 \geq \|(\nu - \pi_u)\|_2$$

References

1. D. P. de Farias and B. Van Roy, *The Linear Programming Approach to Approximate Dynamic Programming*, Operations Research, Vol. 51, No. 6, 2003.
2. D. P. de Farias, *The Linear Programming Approach to Approximate Dynamic Programming: Theory and Application*, Ph.D. Thesis, Stanford University, June 2002.
3. D. P. de Farias and B. Van Roy, *On Constraint Sampling for the Linear Programming Approach to Approximate Dynamic Programming*, to appear in Mathematics of Operations Research, submitted August, 2001.
4. D. P. de Farias and B. Van Roy, *On the Existence of Fixed Points for Approximate Value Iteration and Temporal-Difference Learning, Journal of Optimization Theory and Applications, Vol. 105, No. 3, June, 2000.*
5. Dimitris Bertsimas and John N. Tsitsiklis, *Introduction to Linear Optimization, Athena Scientific.*