1 Outline

- The Linear Programming (LP) Approach to Dynamic Programming (DP)
- Approximate Linear Programming (ALP)
- State-Relevance Weights

2 The Linear Programming Approach to Dynamic Programming

In previous lectures, we discussed solving Bellman’s equation via value iteration and policy iteration. We then introduced approximate dynamic programming (ADP), along with approximation architectures, as a means of solving real-world problems and overcoming Bellman’s curse of dimensionality. In this lecture, we will discuss an alternative approach - linear programming - to solving ADP problems.

We begin with a lemma that will become useful in later proofs.

Lemma 1 If \( J \leq T J \), then \( J \leq J^* \).

Proof Suppose that \( J \leq T J \). Due to the monotonicity property of the operator \( T \), applying \( T \) to both sides repeatedly \( k - 1 \) times yields the following sequence of inequalities

\[
J \leq T J \leq T^2 J \cdots \leq T^k J.
\]

Since \( \lim_{k \to \infty} T^k J = J^* \), we conclude that \( J \leq J^* \).

We now turn to the LP approach to DP. Consider the following nonlinear programming problem:

\[
\begin{align*}
\max_J & \quad c^T J \\
\text{s.t.} & \quad T J \geq J
\end{align*}
\]

(1)

where \( c \) is a vector with positive components, which we refer to as state-relevance weights. These state-relevance weights play an important role in the LP approach to ADP, and will be discussed in more detail in the next subsection. Notice that the above program is nonlinear because \( T \) is a nonlinear operator, taking the minimum over a set of actions. However, after noting that each constraint \( T J(x) \geq J(x) \), i.e.,

\[
\min_{a \in \mathcal{A}_x} g_a(x) + \alpha \sum_{y \in \mathcal{S}} P_a(x, y) J(y) \geq J(x),
\]

we can make a standard transformation of (1) to arrive at what we call the exact LP:

\[
\begin{align*}
\max_J & \quad c^T J \\
\text{s.t.} & \quad g_a(x) + \alpha \sum_{y \in \mathcal{S}} P_a(x, y) J(y) \geq J(x), \quad \forall x \in \mathcal{S}, \forall a \in \mathcal{A}_x
\end{align*}
\]

(2)
Lemma 2 If \( c > 0 \), then \( J^* \) is the unique optimal solution to (1).

Proof First, note that \( J^* \) is a feasible solution to (1) since \( J^* = T J^* \leq T J^* \). Let \( J^* \) be an optimal solution and let \( c > 0 \). Suppose that \( \bar{J} \) is also an optimal solution (1) and that there exists a state \( \hat{x} \) such that \( \bar{J}(\hat{x}) < J^*(\hat{x}) \).

Then,

\[
\sum_{x \in S} c(x) \bar{J}(x) = c(\hat{x}) \bar{J}(\hat{x}) + \sum_{x \in S: x \neq \hat{x}} c(x) \bar{J}(x) < c(\hat{x}) J^*(\hat{x}) + \sum_{x \in S: x \neq \hat{x}} c(x) J^*(x)
\]

\[
= \sum_{x \in S} c(x) J^*(x)
\]

This contradicts the assumption that \( \bar{J} \) is also an optimal solution. Hence, \( J^* \) is unique. \( \square \)

MPD Example Consider a simple three state example in which we can only control the departure probability out of state 1. [Picture to come]

The corresponding exact LP for this problem is:

\[
\max_J c^T J
\]

s.t. \( g_1(1) + \alpha[(1 - \delta)J(1) + \delta J(2)] \geq J(1) \)

\( g_2(1) + \alpha[\delta J(1) + (1 - \delta)J(2)] \geq J(1) \)

\( g(2) + \alpha[\delta J(1) + (1 - 2\delta)J(2) + \delta J(3)] \geq J(2) \)

\( g(3) + \alpha[\delta J(2) + (1 - \delta)J(3)] \geq J(3) \)

The problem with the exact LP formulation is that it requires an overwhelming number of constraints, one for each state-action pair, and a variable for each state. As a result, for practical problems, we would like to try an alternative approach, and so we now turn to applying linear programming to approximate dynamic programming problems.

3 Approximate Linear Programming

We now consider the linear programming approach to approximate dynamic program. Suppose we have pre-selected set of basis functions \( \phi_k, k = 1, \ldots, K \). We are interested in computing a weight vector \( \tilde{r} \in \mathbb{R}^K \) so that the approximation \( \Phi r \) is close to \( J^* \). After replacing \( J \) with its approximation \( \Phi r \), one could suggest the following optimization model:

\[
\max_r c^T r
\]

s.t. \( T r \geq \Phi r \)

[Picture from [1]]

Note that after tranforming this problem to a linear program, the number of variables will most likely decrease, while the number of constraints remains the same (one for each state-action pair).
4 State-Relevance Weights

As we discussed for value and policy iteration, for large-scale problems, it may be too time-consuming to wait until the max norm error is below some tolerance. More importantly, it seems unreasonable to demand convergence in states that are rarely visited for a given policy. This intuition leads us to consider state-relevance weights that give more weight to highly frequented states and less weight to those visited infrequently.

Lemma 3 A vector \( \tilde{r} \in \mathbb{R}^K \)

\[
\text{max}_{r} \quad c^T \Phi r \\
\text{s.t.} \quad T \Phi r \geq \Phi r 
\]

if and only if it solves

\[
\text{min}_{r \in \mathbb{R}^p} \quad \| J^* - \Phi r \|_{1,c} \\
\text{s.t.} \quad T \Phi r \geq \Phi r
\]

Proof From Lemma 1, it follows that for any \( J \) with \( J \leq TJ \), we have

\( J \leq TJ \leq T^2J \leq \cdots \leq J^* \).

Hence, any \( r \) that is feasible to the optimization problems of interest satisfies \( \Phi r \leq J^* \). It follows that

\[
\| J^* - \Phi r \|_{1,c} = \sum_{x \in S} c(x)|J^*(x) - (\Phi r)(x)| \\
= c^T J^* - c^T \Phi r,
\]

and maximizing \( c^T \Phi r \) is therefore equivalent to minimizing \( \| J^* - \Phi r \|_{1,c} \).

Let

\[
E_{X \sim v}[J_u(X) - J^*(X)] = \| J_u - J^* \|_{1,v}
\]

Let \( \mu_{u,v}^T = (1-\alpha)v^T \sum_{t=0}^{\infty} \alpha^t P_{u,v}^t \). The measure \( \mu_{u,v} \) captures the expected frequency of visits to each state when the system operates according to policy \( u \), given that the initial state follows the probability distributed \( v \).

Theorem 1 Let \( J \leq TJ \). Then

\[
\| J_{u,j} - J^* \|_{1,v} \leq \frac{1}{1-\alpha} \| J - J^* \|_{1,\mu_{u,j,v}}.
\]

Proof

\[
J_{u,j} - J = (I - \alpha P_{u,j})^{-1} g_{u,j} - J \\
= (I - \alpha P_{u,j})^{-1}[g_{u,j} - (I - \alpha P_{u,j})J] \\
= (I - \alpha P_{u,j})^{-1} g_{u,j} + \alpha P_{u,j} - J \\
= (I - \alpha P_{u,j})^{-1}(TJ - J)
\]
Because $J \leq TJ$, we have $J \leq TJ \leq J^* \leq J_{uJ}$. Hence,

\[
\|J_{uJ} - J^*\|_{1,v} = v^T(J_{uJ} - J^*) \\
\leq v^T(J_{uJ} - J) \\
= v^T(I - \alpha P_{uJ})^{-1}(TJ - J) \\
= \frac{1}{1 - \alpha} \mu_{uJ,v}^T(TJ - J) \\
\leq \frac{1}{1 - \alpha} \mu_{uJ,v}^T(J^* - J) \\
= \frac{1}{1 - \alpha} \|J - J^*\|_{1,\mu_{uJ,v}}.
\]

\[\square\]

References