6.231 DYNAMIC PROGRAMMING

LECTURE 21

LECTURE OUTLINE

• Review of approximate policy iteration
• Projected equation methods for policy evaluation
• Issues related to simulation-based implementation
• Multistep projected equation methods
• Bias-variance tradeoff
• Exploration-enhanced implementations
• Oscillations
• For a fixed policy \( \mu \) to be evaluated, consider the corresponding mapping \( T \):

\[
(TJ)(i) = \sum_{i=1}^{n} p_{ij} (g(i, j) + \alpha J(j)), \quad i = 1, \ldots, n,
\]

or more compactly, \( TJ = g + \alpha PJ \)

• Approximate Bellman’s equation \( J = TJ \) by \( \Phi_r = \Pi T(\Phi_r) \) or the matrix form/orthogonality condition \( Cr^* = d \), where

\[
C = \Phi' \Xi (I - \alpha P) \Phi, \quad d = \Phi' \Xi g.
\]
PROJECTED EQUATION METHODS

- Matrix inversion: \( r^* = C^{-1}d \)

- Iterative Projected Value Iteration (PVI) method:

\[
\Phi r_{k+1} = \Pi T(\Phi r_k) = \Pi (g + \alpha P \Phi r_k)
\]

Converges to \( r^* \) if \( \Pi T \) is a contraction. True if \( \Pi \) is projection w.r.t. steady-state distribution norm.

- Simulation-Based Implementations: Generate \( k+1 \) simulated transitions sequence \( \{i_0, i_1, \ldots, i_k\} \) and approximations \( C_k \approx C \) and \( d_k \approx d \):

\[
C_k = \frac{1}{k+1} \sum_{t=0}^{k} \phi(i_t)(\phi(i_t) - \alpha\phi(i_{t+1}))' \approx \Phi' \Xi(I - \alpha P) \Phi
\]

\[
d_k = \frac{1}{k+1} \sum_{t=0}^{k} \phi(i_t)g(i_t, i_{t+1}) \approx \Phi' \Xi g
\]

- LSTD: \( \hat{r}_k = C_k^{-1}d_k \)

- LSPE: \( r_{k+1} = r_k - G_k(C_k r_k - d_k) \) where

\[
G_k \approx G = (\Phi' \Xi \Phi)^{-1}
\]

Converges to \( r^* \) if \( \Pi T \) is contraction.
ISSUES FOR PROJECTED EQUATIONS

- Implementation of simulation-based solution of projected equation $\Phi r \approx J_\mu$, where $C_k r = d_k$ and
  
  $$C_k \approx \Phi' \Xi (I - \alpha P) \Phi, \quad d_k \approx \Phi' \Xi g$$

- Low-dimensional linear algebra needed for the simulation-based approximations $C_k$ and $d_k$ (of order $s$; the number of basis functions).

- Very large number of samples needed to solve reliably nearly singular projected equations.

- Special methods for nearly singular equations by simulation exist; see Section 7.3 of the text.

- Optimistic (few sample) methods are more vulnerable to simulation error

- Norm mismatch/sampling distribution issue

- The problem of bias: Projected equation solution $\neq \Pi J_\mu$, the “closest” approximation of $J_\mu$

- Everything said so far relates to policy evaluation. How about the effect of approximations on policy improvement?

- We will next address some of these issues
MULTISTEP METHODS

• Introduce a multistep version of Bellman’s equation $J = T^{(\lambda)} J$, where for $\lambda \in [0, 1)$,

$$T^{(\lambda)} = (1 - \lambda) \sum_{\ell=0}^{\infty} \lambda^{\ell} T^{\ell+1}$$

Geometrically weighted sum of powers of $T$.

• $T^\ell$ is a contraction with mod. $\alpha^\ell$, w. r. to weighted Euclidean norm $\| \cdot \|_\xi$, where $\xi$ is the steady-state probability vector of the Markov chain.

• Hence $T^{(\lambda)}$ is a contraction with modulus

$$\alpha_\lambda = (1 - \lambda) \sum_{\ell=0}^{\infty} \alpha^{\ell+1} \lambda^\ell = \frac{\alpha(1 - \lambda)}{1 - \alpha \lambda}$$

Note $\alpha_\lambda \to 0$ as $\lambda \to 1$ - affects norm mismatch.

• $T^\ell$ and $T^{(\lambda)}$ have the same fixed point $J_\mu$ and

$$\| J_\mu - \Phi r_\lambda^* \|_\xi \leq \frac{1}{\sqrt{1 - \alpha_\lambda^2}} \| J_\mu - \Pi J_\mu \|_\xi$$

where $\Phi r_\lambda^*$ is the fixed point of $\Pi T^{(\lambda)}$.

• $\Phi r_\lambda^*$ depends on $\lambda$. 
BIAS-VARIANCE TRADEOFF

Subspace \( S = \{ \Phi_r | r \in \mathbb{R}^s \} \)

Solution of projected equation \( \Phi_r = \Pi T^{(\lambda)}(\Phi_r) \)

- From \( \| J_\mu - \Phi r_{\lambda,\mu} \|_\xi \leq \frac{1}{\sqrt{1 - \alpha_\lambda^2}} \| J_\mu - \Pi J_\mu \|_\xi \)

  error bound

- As \( \lambda \uparrow 1 \), we have \( \alpha_\lambda \downarrow 0 \), so error bound (and quality of approximation) improves:

  \[
  \lim_{\lambda \uparrow 1} \Phi r_{\lambda,\mu} = \Pi J_\mu
  \]

- But the simulation noise in approximating

  \[
  T^{(\lambda)} = (1 - \lambda) \sum_{\ell=0}^{\infty} \lambda^\ell T^{\ell+1}
  \]

  increases

- Choice of \( \lambda \) is usually based on trial and error
MULTISTEP PROJECTED EQ. METHODS

- The multistep projected Bellman equation is
  \[ \Phi r = \Pi T^{(\lambda)}(\Phi r) \]
- In matrix form: \( C^{(\lambda)} r = d^{(\lambda)} \), where
  \[
  C^{(\lambda)} = \Phi' \Xi (I - \alpha P^{(\lambda)}) \Phi, \quad d^{(\lambda)} = \Phi' \Xi g^{(\lambda)},
  \]
  with
  \[
  P^{(\lambda)} = (1 - \lambda) \sum_{\ell=0}^{\infty} \alpha^\ell \lambda^\ell P^{\ell+1}, \quad g^{(\lambda)} = \sum_{\ell=0}^{\infty} \alpha^\ell \lambda^\ell P^\ell g
  \]
- The LSTD(\(\lambda\)) method is \( (C_k^{(\lambda)})^{-1} d_k^{(\lambda)} \), where \( C_k^{(\lambda)} \) and \( d_k^{(\lambda)} \) are simulation-based approximations of \( C^{(\lambda)} \) and \( d^{(\lambda)} \).
- The LSPE(\(\lambda\)) method is
  \[
  r_{k+1} = r_k - \gamma G_k (C_k^{(\lambda)} r_k - d_k^{(\lambda)})
  \]
  where \( G_k \) is a simulation-based approx. to \((\Phi' \Xi \Phi)^{-1}\)
- TD(\(\lambda\)): An important simpler/slower iteration [similar to LSPE(\(\lambda\)) with \( G_k = I \) - see the text].
MORE ON MULTISTEP METHODS

- The simulation process to obtain $C_k^{(\lambda)}$ and $d_k^{(\lambda)}$ is similar to the case $\lambda = 0$ (single simulation trajectory $i_0, i_1, \ldots$, more complex formulas)

$$C_k^{(\lambda)} = \frac{1}{k+1} \sum_{t=0}^{k} \phi(i_t) \sum_{m=t}^{k} \alpha^{m-t} \lambda^{m-t} \left( \phi(i_m) - \alpha \phi(i_{m+1}) \right)'$$

$$d_k^{(\lambda)} = \frac{1}{k+1} \sum_{t=0}^{k} \phi(i_t) \sum_{m=t}^{k} \alpha^{m-t} \lambda^{m-t} g_{i_m}$$

- In the context of approximate policy iteration, we can use optimistic versions (few samples between policy updates).

- Many different versions (see the text).

- Note the $\lambda$-tradeoffs:
  - As $\lambda \uparrow 1$, $C_k^{(\lambda)}$ and $d_k^{(\lambda)}$ contain more "simulation noise", so more samples are needed for a close approximation of $r_{\lambda,\mu}$
  - The error bound $\| J_\mu - \Phi r_{\lambda,\mu} \|_\xi$ becomes smaller
  - As $\lambda \uparrow 1$, $\Pi T^{(\lambda)}$ becomes a contraction for arbitrary projection norm
1st major issue: **exploration**. Common remedy is the off-policy approach: Replace \( P \) of current policy with

\[
\overline{P} = (I - B)P + BQ,
\]

where \( B \) is a diagonal matrix with \( \beta_i \in [0, 1] \) on the diagonal, and \( Q \) is another transition matrix.

Then LSTD and LSPE formulas must be modified ... otherwise the policy associated with \( \overline{P} \) (not \( P \)) is evaluated (see the textbook, Section 6.4).

Alternatives: Geometric and free-form sampling

Both of these use multiple short simulated trajectories, with random restart state, chosen to enhance exploration (see the text).

Geometric sampling uses trajectories with geometrically distributed number of transitions with parameter \( \lambda \in [0, 1) \). It implements LSTD(\( \lambda \)) and LSPE(\( \lambda \)) with exploration.

Free-form sampling uses trajectories with more generally distributed number of transitions. It implements method for approximation of the solution of a generalized multistep Bellman equation.


**APPROXIMATE PI ISSUES - OSCILLATIONS**

- Define for each policy $\mu$

$$R_\mu = \{ r \mid T_\mu(\Phi r) = T(\Phi r) \}$$

- These sets form the greedy partition of the parameter $r$-space

For a policy $\mu$, $R_\mu$ is the set of all $r$ such that policy improvement based on $\Phi r$ produces $\mu$

- Oscillations of nonoptimistic approx.: $r_\mu$ is generated by an evaluation method so that $\Phi r_\mu \approx J_{\mu}$
MORE ON OSCILLATIONS/CHATTERING

• For optimistic PI a different picture holds

\[ R_{\mu_1} \quad R_{\mu_2} \quad R_{\mu_3} \]

• Oscillations are less violent, but the “limit” point is meaningless!

• Fundamentally, oscillations are due to the lack of monotonicity of the projection operator, i.e., \( J \leq J' \) does not imply \( \Pi J \leq \Pi J' \).

• If approximate PI uses policy evaluation

\[ \Phi r = (WT_{\mu})(\Phi r) \]

with \( W \) a monotone operator, the generated policies converge (to an approximately optimal limit).

• The operator \( W \) used in the aggregation approach has this monotonicity property.