Practice Course 3
Theory and Computation Methods of Approximate DP II

Mengdi Wang

July 6th, 2012

Laboratory for Information and Decision Systems, M.I.T.

1 Theory of ADP: A Review

2 Options Pricing Problem

The Option Model

Exercise 1: Q-Learning

Exercise 2: Approx. PI

Theory of Approximate DP

Infinite-Horizon DP Problem

Minimize over policies

$$\pi = \{\mu_0, \mu_1, \ldots\}$$

the objective cost function

$$J_\pi(x_0) = \lim_{N \to \infty} \mathbf{E}_{w_k, k=0,1,\ldots} \left\{ \sum_{k=0}^{N-1} \alpha^k g\left(x_k, \mu_k(x_k), w_k\right) \right\}$$

How to Approximate DP

- Approximation: parameterize policies/cost vectors, aggregation, etc.
- Simulation: Use simulation-generated trajectories $\{x_k\}$ to calculate DP quantities, without knowing the system

Markovian Decision Process

Assume the system is an n-state (controlled) Markov chain

Change to Markov chain notation

- States $i = 1, \ldots, n$ (instead of x)
- Transition probabilities $p_{i_k i_{k+1}}(u_k)$ [instead of $x_{k+1} = f(x_k, u_k, w_k)$]
- Cost per stage $g(i, u, j)$ [instead of $g(x_k, u_k, w_k)$]
- Cost of a policy $\pi = \{\mu_0, \mu_1, \ldots\}$

$$J_\pi(i) = \lim_{N\to\infty} \mathbf{E}_{\substack{w_k \\ k=0,1,\ldots}} \left\{ \sum_{k=0}^{N-1} \alpha^k g\left(i_k, \mu_k(i_k), i_{k+1}\right) \mid i_0 = i \right\}$$

MDP Continued

The optimal cost vector satisfies the Bellman equation for all i

$$J^*(i) = \min_{u \in U} \sum_{j=1}^{n} p_{ij}(u)(g(i,u,j) + \alpha J^*(j)),$$

or in matrix form

$$J^* = \min_{\mu:\{1,\ldots,n\} \mapsto U} \{g_\mu + \alpha P_\mu J^*\}.$$

Shorthand notation for DP mappings

$$(TJ)(i) = \min_{u \in U(i)} \sum_{j=1}^{n} p_{ij}(u)\big(g(i,u,j) + \alpha J(j)\big), \quad i = 1, \ldots, n,$$

$$(T_\mu J)(i) = \sum_{j=1}^{n} p_{ij}\big(q(i)\big)\big(g\,(i, \mu(i), j) + \alpha J(j)\big), \quad i = 1, \ldots, n$$

Approximation Architecture

Approximation in Policy Space

Parameterize the set of policies μ using a vector r, and then optimize over r.

Approximation in Value Space

Approximate J^* and J_μ from a family of functions parameterized by r, e.g., a linear approximation

$$J \approx \Phi r, \qquad J(i) \approx \phi(i)' r.$$

Approximate DP Algorithms: A Roadmap

Approximate PI

- Approximate Policy Evaluation $\tilde{J}_{\mu_t} \approx T_{\mu_t}\tilde{J}_{\mu_t}$
  - Direct approach; temporal difference methods
- Approximate Policy Improvement $T_{\mu_{t+1}}\tilde{J}_{\mu_t} \approx T\tilde{J}_{\mu_t}$

Aggregation

- Use aggregation states to define a smaller DP problem.

$$\tilde{J} = DT\Phi\tilde{J} = \hat{T}\tilde{J}$$

- D has rows as disaggregation probability distribution
- Φ has columns as aggregation distributions.
- Solve the small aggregate DP problem exactly (VI/PI).

Approximate DP Algorithms: Roadmap Continued

Approximate J^* and Q^*

Solve $J^* = TJ^*$ or $Q^* = FQ^*$ directly by simulation, e.g.,

- Q- Learning: solve $Q^* = FQ^*$ by sampling and stochastic approximation.
- Bellman Error Minimization: solve the following least squares by sampling

$$\min_r \sum_{i=1}^n \|\tilde{J}(i,r) - T\tilde{J}(i,r)\|^2$$

- LP approach

Q-Factors in Discounted MDP

Definition of Q-Factors

$Q^*(i,u) = \sum_{j=1}^n p_{ij}(u)\left[g(i,u,j) + \alpha J^*(j)\right]$

Q^* and J^* imply each other.

Three Equivalent Forms of Bellman Equations

$$J^*(i) = \min_u \sum_{j=1}^n p_{ij}(u)\left[g(i,u,j) + \alpha J^*(j)\right]$$

$$J^*(i) = \min_u Q^*(i,u), \qquad Q^*(i,u) = \sum_{j=1}^n p_{ij}(u)\left[g(i,u,j) + \alpha J^*(j)\right]$$

$$Q^*(i,u) = \sum_{j=1}^n p_{ij}(u)\left[g(i,u,j) + \alpha \min_v Q^*(j,v)\right]$$

Q-Learning

Q-learning is simulation-based VI for Q-factors.
Solve the Bellman equation for Q-factors directly, by using samples:

$$Q^* = FQ^* \quad \Leftrightarrow \quad Q^*(i,u) = \sum_{j=1}^{n} p_{ij}(u)\left[g(i,u,j) + \alpha \min_v Q^*(j,v)\right]$$

Q-Learning Algorithm (approximation of $Q_{k+1} = FQ_k$)

- Generate $\{(i_k, u_k, j_k)\}$: sample (i_k, j_k) according to the system using control u_k.
- Update for each (i_k, u_k, j_k) with stepsize $\gamma_k > 0$:

$$\begin{aligned} Q_{k+1}(i_k,u_k) &= (1-\gamma_k)Q_k(i_k,u_k,j_k) + \gamma_k \text{ Sample}(FQ_k) \\ &= (1-\gamma_k)Q_k(i_k,u_k) \\ &\quad + \gamma_k\left(g(i_k,u_k,j_k) + \alpha \min_v Q_k(j_k,v)\right) \end{aligned}$$

Q-Learning for Optimal Stopping Problem

Stopping problem:

- $C(i)$: cost of stopping at state i
- $Q(i)$: short notation for $Q(i, HOLD)$.
- $g(i, HOLD, j) = 0$.
- Bellman equation:

$$Q^* = FQ^* \;\Leftrightarrow\; Q^*(i) = \sum_{j=1}^{n} p_{ij}(HOLD)\left(\alpha \min\{C(j), Q^*(j)\}\right)$$

Q-Learning Algorithm (approximation of $Q_{k+1} = FQ_k$)

- Generate $\{(i_k, j_k)\}$ according to the stochastic system
- Update for each i_k by using a stepsize $\gamma_k > 0$:

$$Q_{k+1}(i_k) = (1 - \gamma_k)Q_k(i_k) + \alpha\gamma_k \min\{C(j_k), Q_k(j_k)\}$$

1. Theory of ADP: A Review

2. Options Pricing Problem
   - The Option Model
   - Exercise 1: Q-Learning
   - Exercise 2: Approx. PI

Call Options

A call option gives the buyer of the option the right to buy the stock at a fixed price (strike price or K).

Valuing American Call Options

Valuing American options requires the solution of an optimal stopping problem:

$$\text{Option Price} = \mathbf{E}\left[S(t^*) - K \mid \text{Option eventually exercised }\right]$$

where

$$t^* = \text{optimal exercising time.}$$

If the option writers do not solve t^* correctly, the option buyers will have an arbitrage opportunity to exploit the option writers.

Infinite-Horizon DP Formulation

Assume that:

- Dynamics of underlying asset $S_{t+1} = f(S_t, w_t)$
- State: S_t, price of the underlying asset
- Control: $u_t \in \{\text{Exercise}, \text{Hold}\}$
- Transition cost: $g_t(\text{HOLD}) = 0$, $g_t(\text{Exercise}) = S_t - K$.
- The option never expires.
- Once exercised, no more control and cost.
- There exists a discount factor $\alpha \in (0, 1)$

Bellman Equation

Let $J_t(S)$ be the option price at the tth day when the current stock price is S

$$J(S_t) = \max\{S_t - K, \alpha \mathbf{E}\left[J(S_{t+1})\right]\}.$$

Binomial Model

For simplicity, consider a model with a finite number of states:

$$S_{t+1} = \begin{cases} \min\{U, uS_t\} & \text{with probability p} \\ \max\{D, dS_t\} & \text{with probability 1-p} \end{cases}$$

The Bellman equation is $J = TJ$ where

$$TJ(S) = \max\Big\{S - K, \\ \alpha\left[pJ(\min\{U, uS_t\}) + (1-p)J(\max\{D, dS_t\})\right]\Big\},$$

or $Q = FQ$ where

$$FQ(S) = \alpha\Big(p \max\{S - K, Q(\min\{U, uS_t\})\} \\ + (1-p)\max\{S - K, Q(\max\{D, dS_t\})\Big),$$

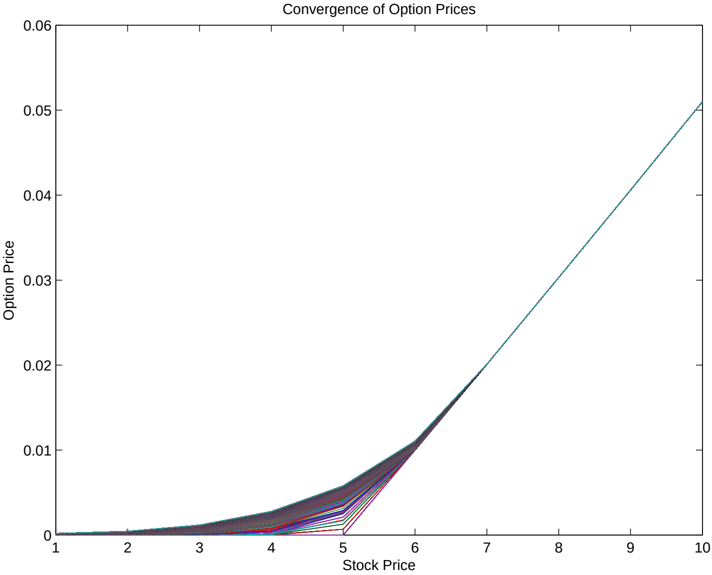
Q-Learning

Exercise 1

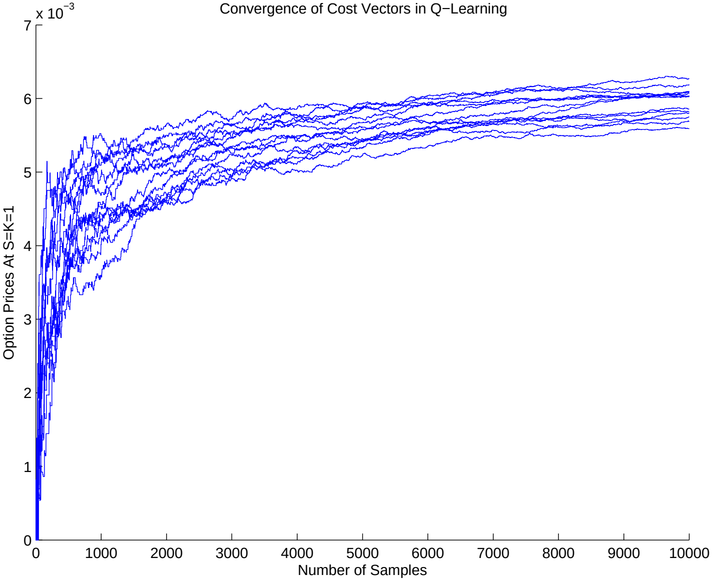
Use Q-learning to evaluate an American call option.

- Construct a simulator that generates trajectories of $\{(i_k, j_k)\}$.
- For each (i_k, j_k), choose an appropriate stepsize γ_k.
- Update the Q-factors by using each sample (i_k, j_k).
- Plot the results.

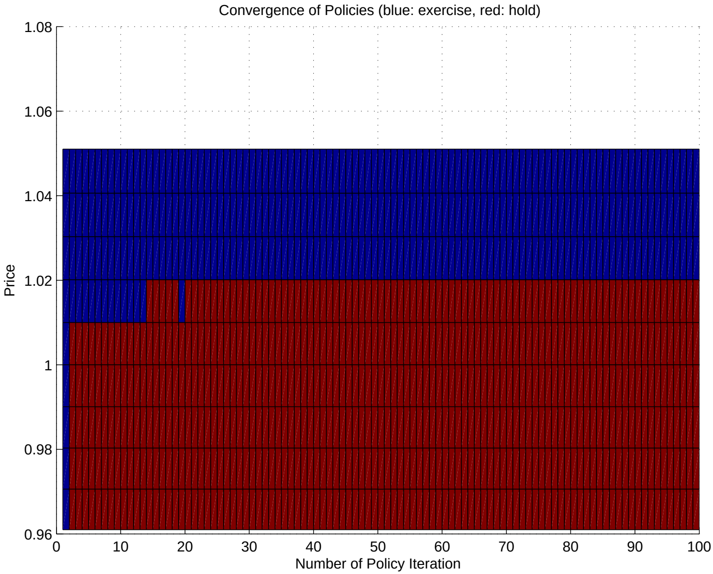
Convergence of Option Prices



Convergence of Option Prices



Convergence of Exercising Policies



Use Approximate PI to Evaluate Options

Exercise 2 (same as in last class)

Use approximate PI to price an American call option. The program should be a function of S_0, T, p, u, d, K.

Suggestions:

- Start with a randomly generated policy $\mu_0 : \{1, \dots, n\} \mapsto \{HOLD, EXERCISE\}$.
- Use approximate policy evaluation (Exercise 1 from last class) to evaluate J_{μ_t} and Q_{μ_t} for a given policy μ_t.
- Plot the trajectories of μ_t.

Features

We will approximate the option prices J^*, J_μ using two set of features, each consisting of 3 features/basis functions.

Simple Polynomial

$$L_0(S) = 1, \quad L_1(S) = S, \quad , L_2(S) = S^2.$$

Laguerre Polynomial

$$L_0(S) = \exp(-S), \quad L_1(S) = \exp(-S)(1 - S),$$
$$L_2(S) = \exp(-S)(1 - S + S^2/2).$$

The basis matrix Φ is an $n \times 3$ matrix.

Policy Iteration for Option Pricing

Algorithm (starts with any μ_0)

- Policy evaluation:
  - Evaluate $J_{\mu_t} \approx \Phi r_\mu$ by approximate policy evaluation: use the program of Exercise 1 to compute r_μ
  - Evaluate the Q-values. For example, for $i_t \in [2, n-1]$,

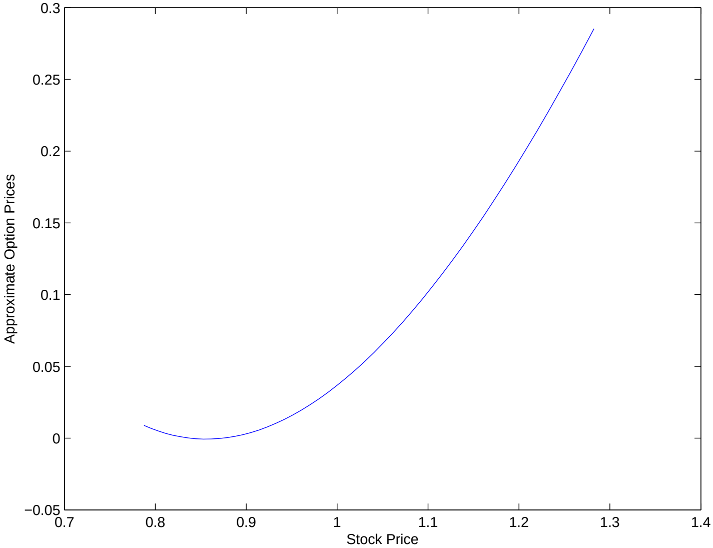
$$\begin{aligned} Q_{\mu_t}(i_t) &= \alpha \mathbf{E}\left[J_{\mu_t}(i_{t+1})\right] \approx \alpha \mathbf{E}\left[\tilde{J}_{\mu_t}(i_{t+1})\right] \\ &= \alpha \left(p\tilde{J}_{\mu_t}(i_t+1) + (1-p)\tilde{J}_{\mu_t}(i_t-1) \right). \end{aligned}$$

  Note $\tilde{J}_\mu(i) = \phi(i)' r_\mu$.

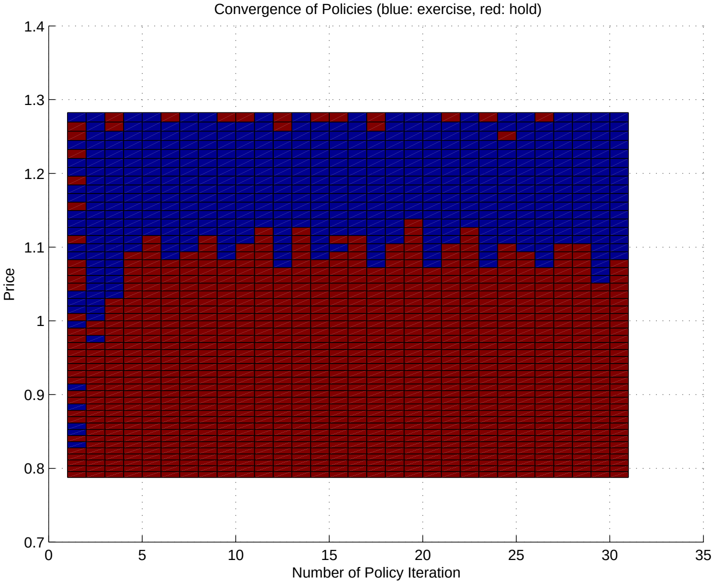
- Policy improvement:

$$\mu_{t+1}(i) = \begin{cases} \text{HOLD} & \text{if } S(i) - K \le Q_{\mu_t}(i), \\ \text{EXERCISE} & \text{Otherwise.} \end{cases}$$

Computation Results - Option Prices



Convergence of Exercising Policies



Online Approximate PI for Q Factors

Exercise 3 (same as in last class)

Modify the program of Exercise 2, so that the policy improvement step uses approximate evaluation of Q-factors (instead of exact Q values calculated using known p).

- For each state i, calculate

$$Q(i) = \mathbf{E}\left[\alpha \tilde{J}(i_{k+1}) \mid i_k = i\right]$$

by averaging the samples obtained from the trajectory

$$Q(i) \approx \frac{\sum_{k=0}^{k=N} \mathbf{1}(i_k = i)\alpha \tilde{J}(i_{k+1})}{\sum_{k=0}^{k=N} \mathbf{1}(i_k = i)}$$

- Note $J(i_{k+1}) = \phi(i_{k+1})'r$.

The end

Thank You Very Much!

Any Question is Welcome :-)