Topics in Reinforcement Learning: AlphaZero, ChatGPT, Neuro-Dynamic Programming, Model Predictive Control, Discrete Optimization
Arizona State University
Course CSE 691, Spring 2024

Links to Class Notes, Videolectures, and Slides at
http://web.mit.edu/dimitrib/www/RLbook.html

Dimitri P. Bertsekas
dpbertsekas@gmail.com

Lecture 8
Rollout and Approximation in Value Space for Stochastic and Continuous Spaces Problems

Outline

Stochastic Rollout: A Special Case of Approximation in Value Space

At State x_k

$$\min_{u_k \in U_k(x_k)} E\Big\{ g_k(x_k, u_k, w_k) + J_{k+1,\pi}\big(f_k(x_k, u_k, w_k)\big)\Big\}$$

Rollout Control $\tilde{u}_k$

Base Policy Cost

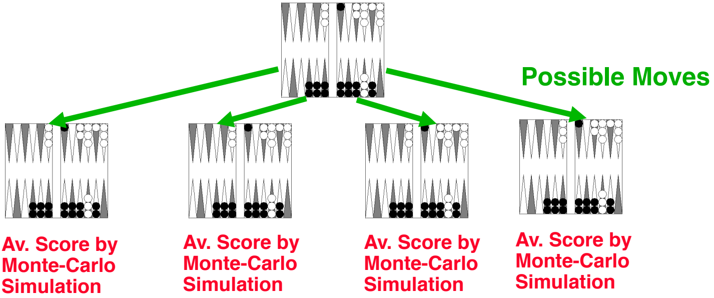
$J_{k+1,\pi}$ is the cost function of some policy π

- The policy π used for rollout is called base policy
- The policy $\tilde{\pi}$ obtained by lookahead minimization is called rollout policy
- Cost improvement property: $J_{k,\tilde{\pi}}(x_k) \leq J_{k,\pi}(x_k)$ for all x_k and k
- It holds because π is a legitimate policy (hence has a sequential consistency property)

Approximate variants: Try to approximate $J_{k+1,\pi}(x_{k+1})$

- Possibility of truncated rollout or other (use pruning, certainty equivalence, etc)
- Use limited simulation

Stochastic Rollout for Backgammon (Tesauro, 1996)



- States are the (board position, dice roll) pairs, actions are the different ways to play the dice roll, stochastic disturbance is the dice roll.
- The 1996 version of TD-Gammon uses truncated rollout with cost function approximation provided by a neural network.
- The neural network is trained off-line by a form of approximate policy iteration that used a temporal differences algorithm for policy evaluation.
- The truncated rollout program (1996) plays better than the one without rollout, and better than any human.
- It is too slow for on-line play due to the excessive on-line Monte Carlo simulation.

Cost Improvement Property of the Nontruncated Version of Rollout: $J_{k,\tilde{\pi}}(x_k) \le J_{k,\pi}(x_k)$ for all x_k and k

We prove this inequality by induction. Clearly it holds for $k = N$, since $J_{N,\tilde{\pi}} = J_{N,\pi} = g_N$. Assuming that it holds for index $k+1$, we have for all x_k,

$$\begin{aligned}
J_{k,\tilde{\pi}}(x_k) &= E\Big\{g_k\big(x_k, \tilde{\mu}_k(x_k), w_k\big) + J_{k+1,\tilde{\pi}}\Big(f_k\big(x_k, \tilde{\mu}_k(x_k), w_k\big)\Big)\Big\} \\
&\le E\Big\{g_k\big(x_k, \tilde{\mu}_k(x_k), w_k\big) + J_{k+1,\pi}\Big(f_k\big(x_k, \tilde{\mu}_k(x_k), w_k\big)\Big)\Big\} \\
&= \min_{u_k \in U_k(x_k)} E\Big\{g_k(x_k, u_k, w_k) + J_{k+1,\pi}\big(f_k(x_k, u_k, w_k)\big)\Big\} \\
&\le E\Big\{g_k\big(x_k, \mu_k(x_k), w_k\big) + J_{k+1,\pi}\Big(f_k\big(x_k, \mu_k(x_k), w_k\big)\Big)\Big\} \\
&= J_{k,\pi}(x_k),
\end{aligned}$$

where:

- The first equality is the DP equation for the rollout policy $\tilde{\pi}$.
- The first inequality holds by the induction hypothesis.
- The second equality holds by the definition of the rollout algorithm.
- The final equality is the DP equation for the base policy π.

Implementation by Simulation (Assuming a Finite Control Space)

- Given x_k, we compute for each $u_k \in U_k(x_k)$ the Q-factor

$$Q_{k,\pi}(x_k, u_k) = E\Big\{g_k(x_k, u_k, w_k) + J_{k+1,\pi}\big(f_k(x_k, u_k, w_k)\big)\Big\}$$

  and minimize over u_k (equivalently compare Q-factor differences).

- This requires that for each u_k, we generate many sample disturbance trajectories $(w_k, w_{k+1}, \ldots, w_{N-1})$ and we obtain the Q-factor as their average cost.
- In practice the number of samples is finite, so the calculated values $\hat{Q}_{k,\pi}(x_k, u_k)$ are approximate and involve stochastic variance.
- We should aim to reduce the variance of the calculated Q-factor differences

$$\hat{Q}_{k,\pi}(x_k, u_k) - \hat{Q}_{k,\pi}(x_k, u_k')$$

  for control pairs (u_k, u_k').

- For variance reduction purposes, it is often best to use the same sample disturbance trajectories $(w_k, w_{k+1}, \ldots, w_{N-1})$ for all u_k (see the "Course in RL" textbook).
- Example: Calculate the difference $q_1 - q_2$ by subtracting two simulation samples $s_1 = q_1 + w_1$ and $s_2 = q_2 + w_2$. $\mathrm{Var}(s_1 - s_2)$ decreases as correlation of w_1 and w_2 increases (it is zero when $w_1 = w_2$).

Stochastic Rollout with Multistep Lookahead

At State x_k
DP minimization

First ℓ Steps — "Future"

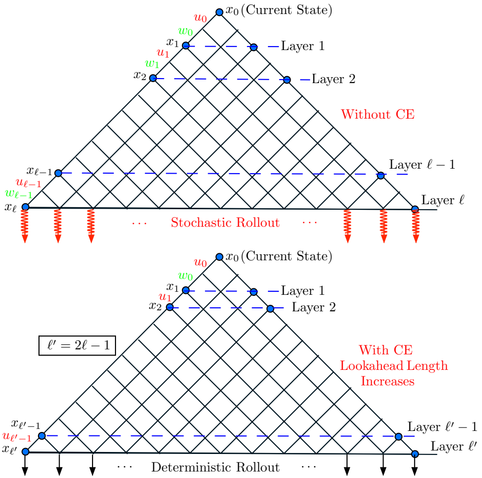
$$\min_{u_k,\mu_{k+1},\ldots,\mu_{k+\ell-1}} E\left\{ g_k(x_k,u_k,w_k) + \sum_{i=k+1}^{k+\ell-1} g_i\big(x_i,\mu_i(x_i),w_i\big) + \tilde J_{k+\ell}(x_{k+\ell}) \right\}$$

Rollout Control $\tilde u_k$
Rollout Policy $\tilde \mu_k$

Lookahead Minimization — **Base Policy Cost**

- Additional cost improvement is obtained with longer lookahead
- But the necessary simulation increases rapidly with the length of the lookahead
- The big issue: How do we save in simulation effort?
  - Truncated rollout
  - Use a certainty equivalence approximation (fix $w_{k+1},\ldots,w_{N-1}$ to nominal values)
  - Monte Carlo Tree Search (MCTS)
- Not much we can say about truncation ... except that it maintains the Newton step character of approximation in value space, and tends to improve the performance and stability of $\tilde\mu$ (relative to approximation in value space without rollout)
- We will next discuss certainty equivalence and MCTS
- Certainty equivalence maintains the Newton step character, MCTS does not

Certainty Equivalence Approximation (Requires Much Less Simulation)



- Fix $w_{k+1}, \ldots, w_{N-1}$ at some nominal values $\overline{w}_{k+1}, \ldots, \overline{w}_{N-1}$, and Monte Carlo average many trajectories of the form $(w_k, \overline{w}_{k+1}, \ldots, \overline{w}_{N-1})$.
- A two-fold benefit: Deterministic rather than stochastic simulation, and fewer applications of the base policy.

Monte Carlo Tree Search - Approximation of Lookahead Minimization
Motivation is to Save Simulation Effort

We assumed equal effort for evaluation of Q-factors of all controls at a state x_k

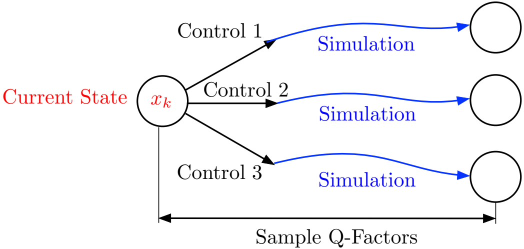
Drawbacks:

- Some controls may be clearly inferior to others and may not be worth as much sampling effort.
- Some controls that appear to be promising may be worth exploring better through more sampling and also multistep lookahead.

Monte Carlo Tree Search (MCTS) is a form of approximate lookahead minimization that tries to economize in simulation time

- MCTS involves adaptive simulation (simulation effort adapted to the perceived quality of different controls).
- Aims to balance exploitation (extra simulation effort on controls that look promising) and exploration (adequate exploration of the potential of all controls).
- MCTS (like simplified rollout and pruning) does not directly improve performance; it just tries to save in simulation effort. But the saving allows longer lookahead for a given computational budget.

MCTS - One-Step Approximation in Value Space



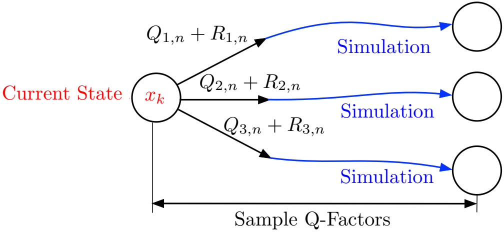
MCTS provides an economical sampling policy to estimate the Q-factors

$$\tilde{Q}_k(x_k, u_k) = E\Big\{ g_k(x_k, u_k, w_k) + \tilde{J}_{k+1}\big(f_k(x_k, u_k, w_k)\big) \Big\}, \qquad u_k \in U_k(x_k)$$

Simulation scheme: Pick a control u (in some way) and generate a single sample of its Q-factor

- After the nth sample we have $Q_{u,n}$, the empirical mean of the Q-factor of each control u (total sample value divided by total number of samples corresponding to u). We can view $Q_{u,n}$ as an exploitation index (a measure of quality of u).
- We could use the estimates $Q_{u,n}$ to select the control to sample next ... but how do we make sure that we do not overlook some less explored controls.

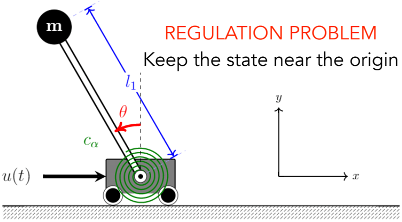
MCTS Based on Statistical Tests

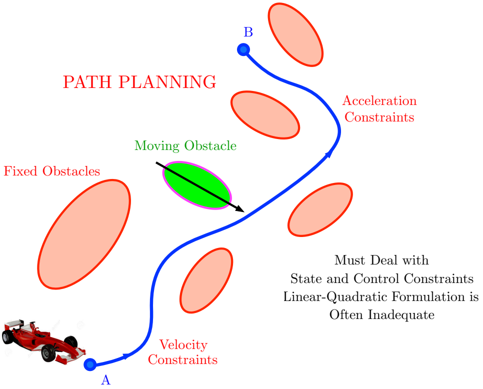


Main idea: To balance exploitation (sample controls that seem most promising, i.e., a small $Q_{u,n}$) and exploration (sample controls with small sample count).

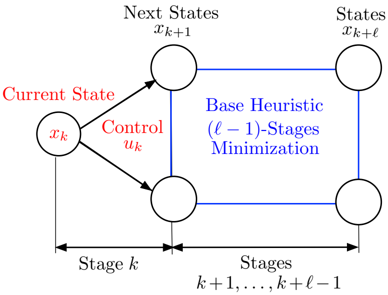
- A popular (semi-heuristic) strategy: Sample next the control u that minimizes the sum $Q_{u,n} + R_{u,n}$ where $R_{u,n}$ is an exploration index.
- $R_{u,n}$ is based on a confidence interval formula and depends on the sample count S_u of control u (which comes from analysis of multiarmed bandit problems).
- The UCB rule (upper confidence bound) sets $R_{u,n} = -c\sqrt{\log n / S_u}$, where c is a positive constant, selected empirically (values $c \approx \sqrt{2}$ are suggested, assuming that $Q_{u,n}$ is normalized to take values in the range $[-1, 0]$).
- MCTS with UCB rule has been extended (heuristically) to multistep lookahead ... but AlphaZero has used a different (semi-heuristic) rule.

Classical Control Problems - Infinite State and Control Spaces





Approximation in Value Space/Rollout for Infinite-Spaces Problems



Suppose the control space is infinite (so the number of Q-factors is infinite)

- One possibility is discretization of $U_k(x_k)$; but the number of Q-factors is excessive.
- Another possibility is to use optimization heuristics that look $(\ell - 1)$ steps ahead.
- Seemlessly combine the kth stage minimization and the optimization heuristic into a single ℓ-stage deterministic optimization (under favorable circumstances).
- Can solve it by nonlinear programming/optimal control methods (e.g., quadratic programming, gradient-based). Constraints can be readily accommodated.
- Possibility of a terminal cost approximation.
- This is the idea underlying model predictive control (MPC).

Approximation in Value Space for Multistage Linear/Integer Programming

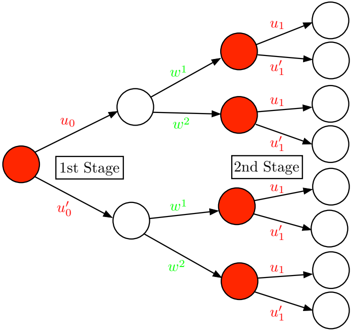
Generic resource allocation over time:

- System: $x_{k+1} = A_k x_k + B_k u_k$, ($x_k$ and u_k are vectors, A_k, B_k are given matrices)
- Objective: Minimize a linear cost

$$c_N{}' x_N + \sum_{k=0}^{N-1} (c_k{}' x_k + d_k{}' u_k)$$

  over N stages (c_k, d_k are given vectors, prime denotes transpose).
- Constraints: Linear on x_k and u_k (possibly some additional integer constraints).

- For large N and/or integer constraints this is a hard problem.
- One possibility: Truncate the horizon and use terminal cost approximation.
- Readily handles on-line replanning.
- Generalizes to integer constraints, making use of integer programming software.
- Using a different/simpler type of base heuristic and discretized DP is an alternative, but does not exploit the linear programming structure of the problem.



Example

u_0: Decide on production resources

w: Demand for product (values w^1, w^2 w/ Probs p^1, p^2)

u_1: Satisfy the demand as best as possible

- In the first stage we choose a vector $u_0 \in U_0$ with cost $g_0(u_0)$.
- Then an uncertain event will occur, represented by a random variable w, which takes one of the values $w^1, \ldots, w^m$ with probabilities $p^1, \ldots, p^m$.
- Once w occurs, we will know its value w^i, and then we choose a vector $\mu_1(u_0, w^i) \in U_1(u_0, w^i)$ at a cost $g_1\big(\mu_1(u_0, w^i), w^i\big)$.
- The objective is to minimize the expected cost $g_0(u_0) + \sum_{i=1}^{m} p^i g_1\big(\mu_1(u_0, w^i), w^i\big)$
- Can be viewed as a nonlinear programming problem, whose optimization variables are u_0, $\mu_1(u_0, w^i)$, $i = 1, \ldots, m$ (five vectors in the figure).

Rollout for Multistage Stochastic Programming

In multistage stochastic programming, the decision u_k at the kth stage is a function of the history $(u_0, w_0, u_1, w_1, \ldots, u_{k-1}, w_{k-1})$, the state of the kth stage.

Similar formulation to the two-stage case ... but exact solution by DP or NLP gets rapidly out of hand as the number of stages increases.

RL view of multistage stochastic programming:

- It is a special case of finite horizon stochastic optimal control.
- Approximation in value space applies.
- Rollout with or without truncation applies.
- Base heuristic could be based on two-stage stochastic programming with terminal cost approximation.
- Certainty equivalence approximations can be very useful: (Only w_0 is stochastic and subsequent disturbances are fixed at nominal values).
- Limited simulation approximations after the first step are possible.

About the Next Lecture

- Sequential estimation, Bayesian Optimization
- Adaptive control and rollout with a POMDP approach.
- Application to the Wordle puzzle.

Please review Sections 2.10 and 2.11 of the “Course in RL" textbook.

Recommended videolecture (Lecture 9 of the 2023 version of the course) at https://www.youtube.com/watch?v=AE9-81WtAHI.

PLEASE FOCUS ON YOUR TERM PAPER. A ONE-PAGE SUMMARY IS DUE AT THE END OF SPRING BREAK