Policy Gradient Methods in Reinforcement Learning

Peter Bartlett

U. C. Berkeley Statistics/Computer Science

April, 2003.

Policy Gradient Methods

1. Motivation: Performance Degradation with Value Function Methods
2. POMDPs and the Induced Markov Chain
3. Monte-Carlo Gradient Estimates with Recurrent States
4. Monte-Carlo Gradient Estimates without Recurrent States
5. Variance Reduction Techniques

Performance Degradation with Value Function Methods

Recall **exact** policy iteration:

- Given current policy μ_k, compute its value function J^{μ_k}.
- Generate μ_{k+1} by $\mu_{k+1} = \arg\max_\mu P^\mu J^{\mu_k}$.

Important property: μ_{k+1} is guaranteed to be better than μ_k in the sense that

$$J^{\mu_{k+1}} \succeq J^{\mu_k}.$$

Performance Degradation with Value Function Methods

Recall **approximate** policy iteration:

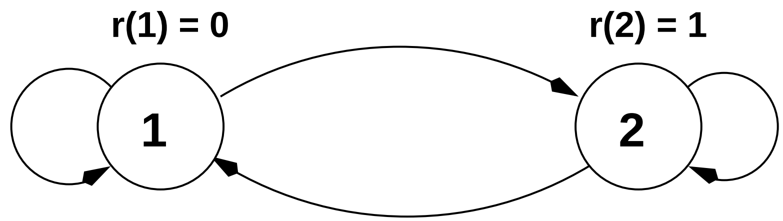
- Given current policy μ_k, compute **approximate** value function $\tilde{J}^{\mu_k}$.
- Generate μ_{k+1} by $\mu_{k+1} = \arg\max_\mu P^\mu \tilde{J}^{\mu_k}$.

The bad news: If $\tilde{J}^{\mu_k}$ approximates J^{μ_k}, the best we can say is:

$$J^{\mu_{k+1}} \geq J^{\mu_k} - \frac{2\alpha}{1-\alpha} \|\tilde{J}^{\mu_k} - J^{\mu_k}\|_\infty e,$$

where $\alpha \in [0, 1)$ is the discount factor.
In other words, unless the approximation is exact ($\|\tilde{J} - J\|_\infty = 0$), μ_{k+1} can be **worse** than μ_k.

$TD(1)$: Performance degradation



- Two controls u_1, u_2:

$$P(u_1) = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix}, \quad P(u_2) = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix},$$

- Optimal policy μ^* always chooses u_1: $\pi^* = [1/3, 2/3]$.
- Set $\tilde{J}(i, w) = w\phi(i)$ where $\phi(1) = 2, \phi(2) = 1$.

$TD(1)$: Performance degradation

- Solving Bellman's equations:

$$J^*(1) = \frac{2\alpha}{3(1-\alpha)} \qquad J^*(2) = 1 + J^*(1)$$

$$w < 0 \quad \Rightarrow \quad \tilde{J}(1,w) < \tilde{J}(2,w) \quad \Rightarrow \quad \text{optimal greedy policy}$$

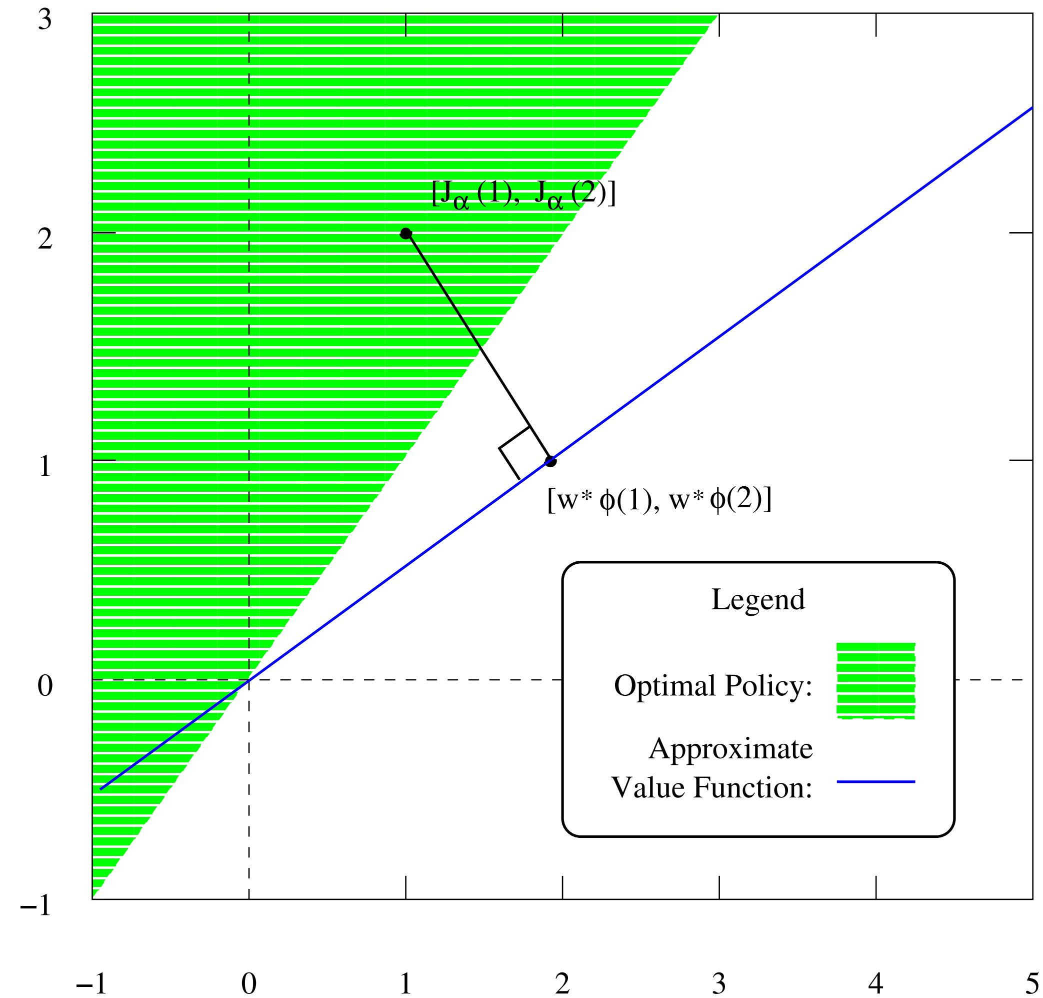
$$w > 0 \quad \Rightarrow \quad \tilde{J}(1,w) > \tilde{J}(2,w) \quad \Rightarrow \quad \text{suboptimal greedy policy}$$

- For optimal policy, solution found by $\mathrm{TD}(1)$ is minimizer of

$$\left\| \tilde{J}(\cdot, w) - J^* \right\|_{\pi^*}^2 \qquad \Rightarrow w = \frac{3+\alpha}{9(1-\alpha)} > 0$$

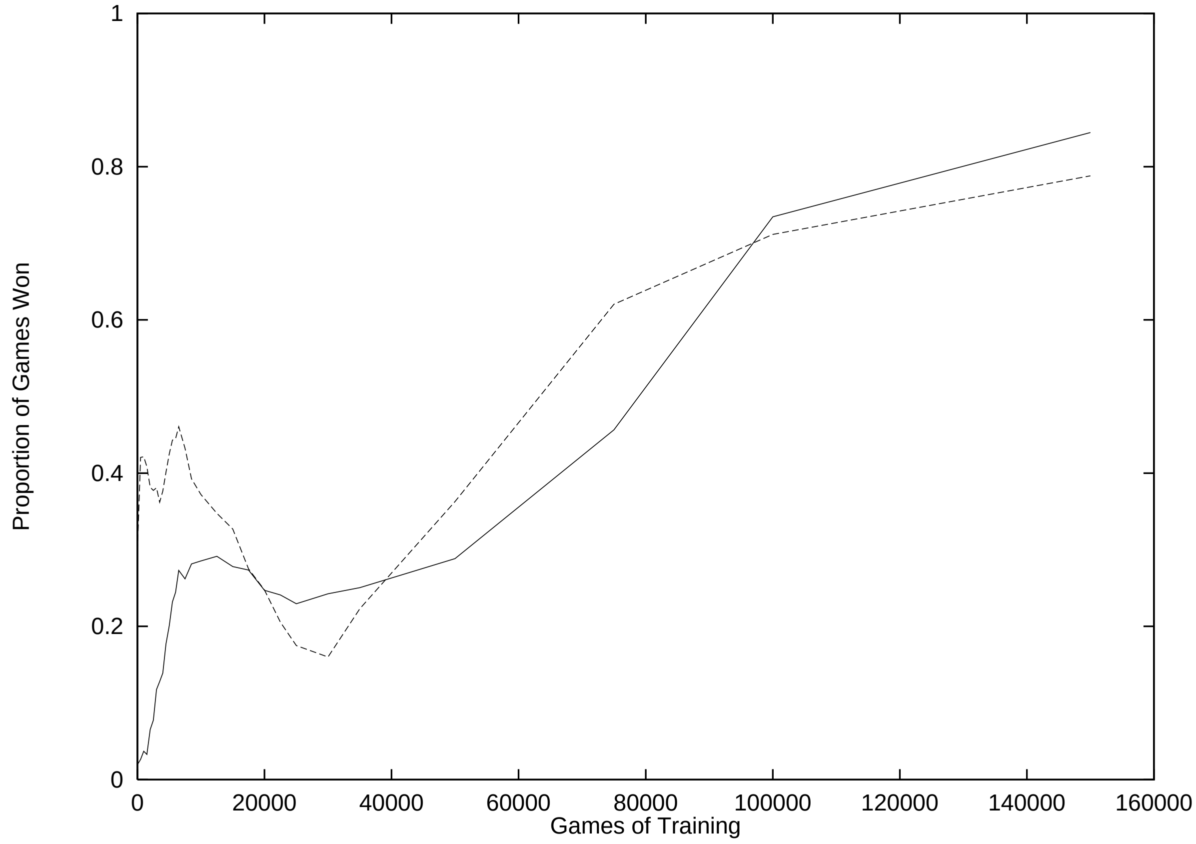
- Implies new greedy policy based on w will be **suboptimal**, even though previous policy was optimal.
- Can show that $w > 0$ for all subsequent policy iterations, so policy remains suboptimal.

Geometric Interpretation



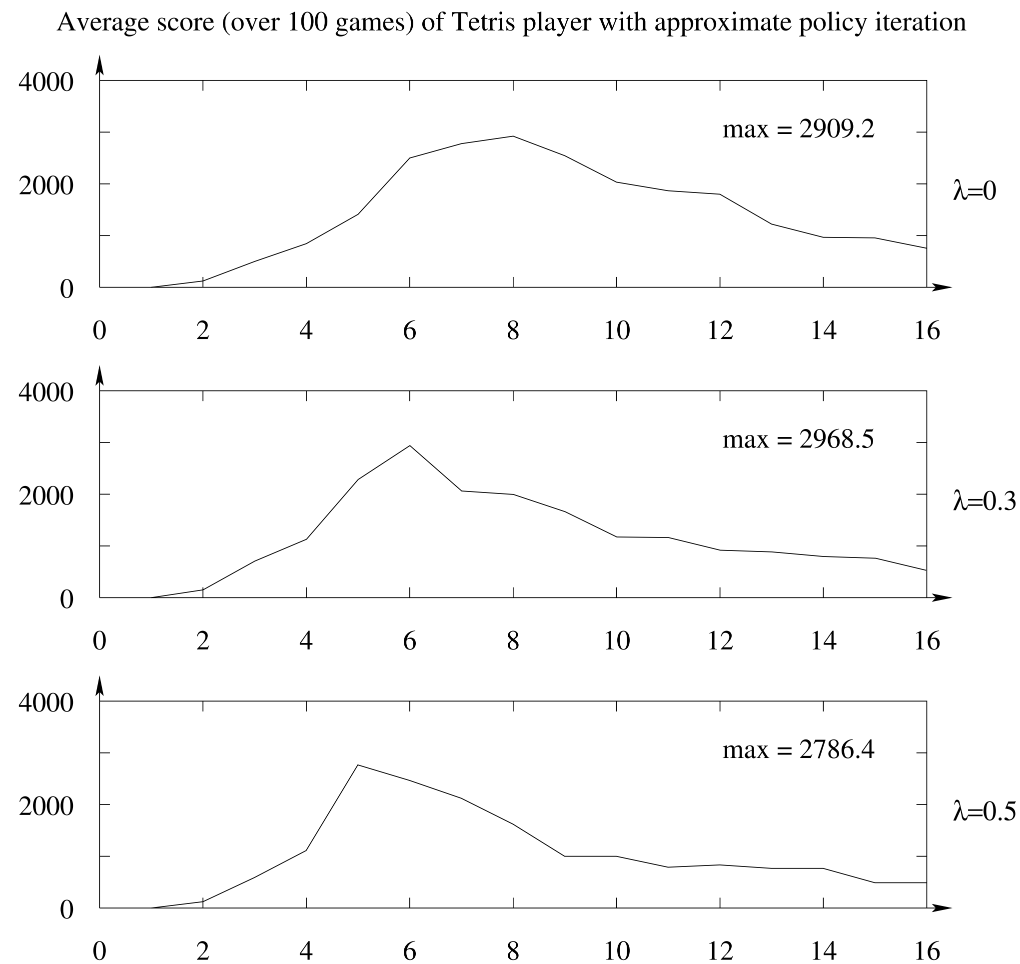
TD-Gammon: Performance Degradation

(Weaver, 2002)



TD(λ) **and Tetris: Performance Degradation**

(Bertsekas and Tsitsiklis, 1996)



Policy Gradient Methods

1. Motivation: Performance Degradation with Value Function Methods
2. POMDPs and the Induced Markov Chain
3. Monte-Carlo Gradient Estimates with Recurrent States
4. Monte-Carlo Gradient Estimates without Recurrent States
5. Variance Reduction Techniques

Policy Gradients

- The **policy** is paramount, **value functions** are secondary.
- View the policy as **parameterized**: $\mu(\theta, i)$.
- The policy could be greedy, based on some parameterized approximate value function $\tilde{J}(\theta, i)$, but it need not be.
- A desirable property of an algorithm for adjusting the parameters θ: it should improve the performance of the policy μ.

Policy Gradients

- Denote policy performance with parameters θ by

$$\eta(\theta) \quad \text{(to be defined shortly)}$$

- Want parameter adjustments to be within 90° of $\nabla\eta(\theta)$.
- So problem reduces to computation of $\nabla\eta(\theta)$.

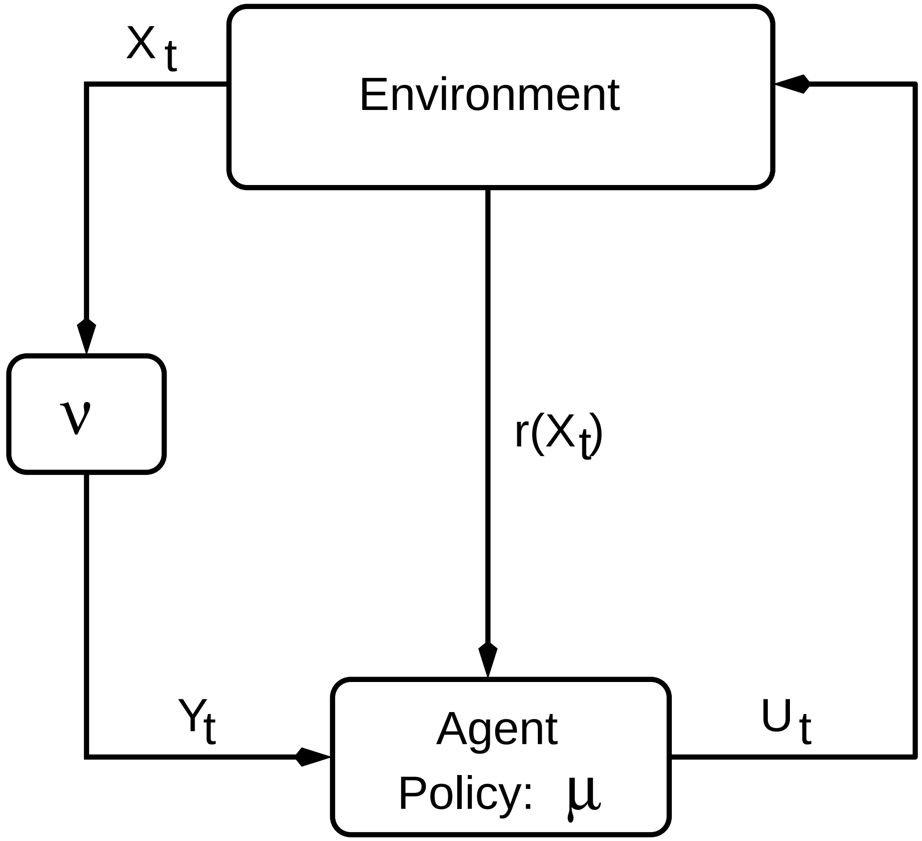
- Policy gradient techniques extend to **partially observable** Markov decision processes.

Partially Observable Markov Decision Processes

States:	$\mathcal{S} = \{1, 2, \ldots, n\}$	$X_t \in \mathcal{S}$
Observations:	$\mathcal{Y} = \{1, 2, \ldots, M\}$	$Y_t \in \mathcal{Y}$
Actions or Controls:	$\mathcal{U} = \{1, 2, \ldots, N\}$	$U_t \in \mathcal{U}$

Observation Process ν:	$\Pr(Y_t = y \mid X_t = i) = \nu_y(i)$
Stochastic Policy μ:	$\Pr(U_t = u \mid Y_t = y) = \mu_u(\theta, y)$
Rewards:	$r : \mathcal{S} \to \mathbb{R}$
Adjustable parameters:	$\theta \in \mathbb{R}^K$
Transition Probabilities:	$\Pr(X_{t+1} = j \mid X_t = i, U_t = u) = p_{ij}(u)$

POMDP



Sequence: $X_0, Y_0, U_0,\ \ X_1, Y_1, U_1, \ldots, X_t, Y_t, U_t, \ldots$

The Induced Markov Chain

- Transition Probabilities:

$$\begin{aligned} p_{ij}(\theta) &= \Pr\left(X_{t+1} = j | X_t = i\right) \\ &= \sum_{y \in \mathcal{Y}} \nu_y(i) \sum_{u \in \mathcal{U}} \mu_u(\theta, y) p_{ij}(u) \end{aligned}$$

- Transition Matrix:

$$P(\theta) = [p_{ij}(\theta)]$$

(Note change of notation: $P^\mu \to P(\theta)$.)

Finite Markov Chain $M = (\mathcal{S}, P)$: Definitions

Evolution of distribution: If $q = (q_1, \ldots, q_n)'$ is the distribution over states at time t, then the distribution at time $t+1$ is $q'P$.

A stationary distribution, π, satisfies $\pi' P = \pi'$ (left eigenvector).

j **is accessible from** i $(i \to j)$ if some t has $[P^t]_{ij} > 0$.

A communicating class is a set of mutually accessible states.

- A **recurrent class** does not access other states.
- A **transient class** is not recurrent.

The period of a state is the greatest common divisor of $\{t : [P^t]_{ii} > 0\}$.

An irreducible Markov chain has a single communicating class.

An ergodic Markov chain is irreducible and aperiodic (period=1).

Finite Markov Chain $M = (\mathcal{S}, P)$: Results

Recurrent classes dominate: $\Pr(X_t \in$ a recurrent class$) \to 1$.

Resonance: All states in a communicating class have the same period.

Uniqueness and positivity of stationary distribution: For an ergodic M, we can order the eigenvalues of P as

$$1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|.$$

The left eigenvector corresponding to $\lambda_1 = 1$ is the unique stationary distribution, and it is strictly positive. ($\Leftarrow$ Perron-Frobenius Theorem.)

Mixing: For ergodic M, $\lim_{t\to\infty} P^t = e\pi'$, where $e = [1 \ldots 1]'$.

π **captures asymptotics:** For ergodic M,

$$\lim_{T\to\infty} \frac{1}{T} \left|\{1 \leq t \leq T : X_t = i\}\right| \stackrel{\text{a.s.}}{=} \pi_i > 0.$$

Ergodic Markov Chains and Stationary Distributions

- If the finite Markov chain $(\mathcal{S}, P(\theta))$ is *ergodic* (ie: no transient states, aperiodic), then it has a unique *stationary distribution*, π:
  - $\pi' P(\theta) = \pi'$.
  - π_i is asymptotic proportion of time spent in state i.
  - Markov chain mixes (distribution of X_t approaches π).
  - $\pi_i > 0$ for all i, so all states are visited infinitely often.

> Convenient Assumption: For all values of the parameters θ, the induced Markov chain $(\mathcal{S}, P(\theta))$ is ergodic.

Exploiting ergodicity: Estimating quantities from simulation

Simulations approximate the stationary distribution :

Suppose we have a function $f : \mathcal{S} \to \mathbb{R}$

(eg: the error $(r(i) - \hat{r}(i))$ of an estimate $\hat{r}$ of the reward function).

Then the empirical average of f^2 over a sample path,

$$\|f\|_T^2 = \frac{1}{T} \sum_{t=1}^{T} f(X_t)^2,$$

approximates

$$\|f\|_\pi^2 = \mathbf{E}_\pi f^2 = \sum_{i \in \mathcal{S}} \pi(i) f(i)^2,$$

in the sense that

$$\lim_{T \to \infty} \|f\|_T^2 \overset{\text{a.s.}}{=} \|f\|_\pi^2.$$

Measuring Performance

- Infinite Horizon, Average Reward:

$$\eta(\theta) = \lim_{N\to\infty} \mathbf{E}\left[\frac{1}{N}\sum_{t=0}^{N-1} r(X_t)\right]$$
$$= \sum_{i\in\mathcal{S}} \pi_i(\theta) r(i)$$

- Problem: optimize $\eta(\theta)$ via gradient ascent.

- Hardest subproblem: compute $\nabla\eta(\theta)$.

Average versus Discounted Reward

Infinite Horizon, Expected Discounted Reward

- The expected discounted value (with initial state $X_0 \sim \pi$) is

$$\begin{aligned} \eta_\alpha(\theta) &= \mathbf{E}_{i\sim\pi} J_\alpha^{\mu(\theta)}(i) \\ &= \mathbf{E}_{i\sim\pi} \lim_{T\to\infty} \mathbf{E}\left[\sum_{t=0}^{T-1} \alpha^t r(X_t) | X_0 = i\right], \end{aligned}$$

  where $0 \leq \alpha < 1$. But this satisfies

$$\eta_\alpha(\theta) = \frac{\eta(\theta)}{1-\alpha}$$

  so optimizing $\eta_\alpha(\theta)$ is equivalent to optimizing $\eta(\theta)$.

From now on, I'll drop the dependence on θ where it's clear.

Policy Gradient Methods

1. Motivation: Performance Degradation with Value Function Methods
2. POMDPs and the Induced Markov Chain
3. Monte-Carlo Gradient Estimates with Recurrent States
4. Monte-Carlo Gradient Estimates without Recurrent States
5. Variance Reduction Techniques

Likelihood Ratio and Monte-Carlo

- **Likelihood Ratio**: general technique for estimating performance gradients via simulation (Monte-Carlo).
- Also known as the **score function** technique.
- Dates back at least to the sixties:

  (Alexandrov, Sysoyev, and Shemeneva, 1968; Rubinstein, 1969)

  For Markov chains: (Glynn, 1986; Reiman and Weiss, 1986; Williams, 1992)

Likelihood Ratio and Monte-Carlo

- Let Z be a random variable in some finite set $\mathcal{Z}$, with a parameterized distribution: $\Pr(Z = z) = q(\theta, z)$.
- For some function $f : \mathcal{Z} \to \mathbb{R}$, the gradient with respect to θ is

$$
\begin{aligned}
\nabla \mathbf{E}_{Z \sim q} f(Z) &= \sum_{z \in \mathcal{Z}} f(z) \nabla q(\theta, z) \\
&= \sum_{z \in \mathcal{Z}} f(z) \frac{\nabla q(\theta, z)}{q(\theta, z)} q(\theta, z) \\
&= \mathbf{E}_{Z \sim q} \left[f(Z) \frac{\nabla q(\theta, Z)}{q(\theta, Z)} \right] \\
&\overset{\text{a.s.}}{=} \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \left[f(Z_t) \frac{\nabla q(\theta, Z_t)}{q(\theta, Z_t)} \right],
\end{aligned}
$$

where $Z_1, \ldots, Z_T$ are i.i.d. with $Z_t \sim q$.

Likelihood Ratio and Monte-Carlo

If we can:

1. generate $Z_1, \ldots, Z_T$ according to $q(\theta, \cdot)$,
2. calculate $f(z)\dfrac{\nabla q(\theta, z)}{q(\theta, z)}$,
3. then

$$\hat{\Delta}_T = \frac{1}{T}\sum_{t=1}^{T} f(Z_t)\frac{\nabla q(\theta, Z_t)}{q(\theta, Z_t)}$$

   is an **unbiased** estimate of $\nabla \mathbf{E}_{Z\sim q} f(Z)$.

Likelihood Ratio for Parameterized Markov Chains

- Pick a **recurrent state** i^* (any state).
- Samples Z are sequences of states between visits to i^*.
- So $Z = (X_0, X_1, \ldots, X_T)$, where $X_0 = X_T = i^*$. (T is **random**)
- Markov condition implies Z are i.i.d. samples from $\mathcal{S} \times \mathcal{S} \times \mathcal{S} \ldots$
- Probability of sequence z is $q(\theta, z) = \prod_{t=0}^{T-1} p_{x_t x_{t+1}}(\theta)$.
- Define: $f(z) := \frac{1}{T} \sum_{t=0}^{T-1} r(x_t)$ (= average reward over z).
- Since $\eta(\theta) = \mathbf{E}_{Z \sim q} f(Z)$,

$$\nabla \eta(\theta) = \mathbf{E}_{Z \sim q} \left[f(Z) \frac{\nabla q(\theta, Z)}{q(\theta, Z)} \right].$$

Likelihood Ratio for Parameterized Markov Chains

- Observe:

$$
\begin{aligned}
\frac{\nabla q(\theta, z)}{q(\theta, z)} &= \nabla \log q(\theta, z) = \nabla \log \prod_{t=0}^{T-1} p_{x_t x_{t+1}}(\theta) \\
&= \nabla \sum_{t=0}^{T-1} \log p_{x_t x_{t+1}}(\theta) = \sum_{t=0}^{T-1} \frac{\nabla p_{x_t x_{t+1}}(\theta)}{p_{x_t x_{t+1}}(\theta)}
\end{aligned}
$$

- Thus,

$$
f(z) \frac{\nabla q(\theta, z)}{q(\theta, z)} = \frac{1}{T} \sum_{s=0}^{T-1} r(x_s) \sum_{t=0}^{T-1} \frac{\nabla p_{x_t x_{t+1}}(\theta)}{p_{x_t x_{t+1}}(\theta)}
$$

and $$\nabla \eta = \mathbf{E} \left[\frac{1}{T} \sum_{t=0}^{T-1} \frac{\nabla p_{X_t X_{t+1}}(\theta)}{p_{X_t X_{t+1}}(\theta)} \sum_{s=0}^{T-1} r(X_s) \right].$$

Average over N loops $Z_1, \ldots, Z_N$ to get an accurate estimate of $\nabla \eta(\theta)$.

Likelihood Ratio for POMDPs

- In this case $z = (x_0, y_0, u_0,\ x_1, y_1, u_1, \ldots, x_{T-1}, y_{T-1}, u_{T-1}, x_T)$, with $x_0 = x_T = i^*$ as before.

- Don't generally have access to p, ν, however

$$\frac{\nabla q(\theta, z)}{q(\theta, z)} = \nabla \log q(\theta, z) = \sum_{t=0}^{T-1} \nabla \log \left(\nu_{y_t}(x_t)\mu_{u_t}(\theta, y_t)p_{x_t x_{t+1}}(u_t)\right)$$

$$= \sum_{t=0}^{T-1} \frac{\nabla \mu_{u_t}(\theta, y_t)}{\mu_{u_t}(\theta, y_t)}.$$

- Thus,

$$\nabla \eta(\theta) = \mathbf{E}\left[\frac{1}{T}\sum_{t=0}^{T-1} \frac{\nabla \mu_{U_t}(\theta, Y_t)}{\mu_{U_t}(\theta, Y_t)} \sum_{s=0}^{T-1} r(X_s)\right].$$

- U_t and Y_t are available, even if X_t is not.

Estimating $\nabla\eta(\theta)$ for POMDPs: Example

Suppose $U_t \sim N(\theta, \sigma^2)$. Then

$$\frac{\nabla\mu_u}{\mu_u} = \frac{u-\theta}{\sigma^2}.$$

Suppose $r(X_t) \in \{0,1\}$ (so $\sum_t r(X_t)/T$ is proportion of successes).

Then $\nabla\eta(\theta)$ is the average of

$$\sum_t \frac{u_t - \theta}{\sigma^2} \times (\text{proportion of successes}) \,.$$

i.e.: sign of gradient depends on the sign of the correlation between $\mu_t - \theta$ and the proportion of successes.

Estimating $\nabla\eta(\theta)$ for POMDPs: REINFORCE

Recursive algorithm to compute $\frac{1}{T}\sum_{t=0}^{T-1}\frac{\nabla\mu_{u_t}(\theta,y_t)}{\mu_{u_t}(\theta,y_t)}\sum_{s=0}^{T-1}r(x_s)$:

1. Set $z_0 = \Delta_0 = 0$, $s_0 = 0$ ($z_0, \Delta_0 \in \mathbb{R}^K$, $s_0 \in \mathbb{R}$).
2. After Y_t, $U_t \sim \mu(\theta, Y_t)$, and $r(X_t)$:

$$s_{t+1} := s_t + r(X_t)$$

$$z_{t+1} := z_t + \frac{\nabla\mu_{U_t}(\theta, Y_t)}{\mu_{U_t}(\theta, Y_t)}$$

3. If $X_{t+1} = i^*$ stop.

Estimate for this loop is $s_{t+1}z_{t+1}/(t+1)$. Compute average over several loops.

Gradient Ascent

Given $\hat{\nabla}\eta(\theta)$, how do we adjust θ? Two approaches:

- **Gradient Ascent**: search for a maximum in the direction $\hat{\nabla}\eta(\theta)$. (Need to cope with noise.)

- **Stochastic Gradient Ascent**: After each loop Z_n:

$$\theta_{n+1} = \theta_n + \gamma_n f(Z_n) \frac{\nabla q(\theta, Z_n)}{q(\theta, Z_n)},$$

  for suitable step-sizes γ_n.

- Typically if $\sum_n \gamma_n = \infty$, $\sum_n \gamma_n^2 < \infty$, can guarantee convergence to local maximum of $\eta(\theta)$ (with probability 1).

Monte-Carlo Policy-Gradient Estimates with Recurrent States

Two problems with using recurrent states:

1. In POMDPs cannot rely on identifying i^*.
2. Since updates are only made at i^*, algorithm **variance** grows with recurrence time (time to return to i^*).

- Would like to estimate $\nabla\eta(\theta)$ without reference to a recurrent state.

Policy Gradient Methods

1. Motivation: Performance Degradation with Value Function Methods
2. POMDPs and the Induced Markov Chain
3. Monte-Carlo Gradient Estimates with Recurrent States
4. Monte-Carlo Gradient Estimates without Recurrent States
5. Variance Reduction Techniques

A biased gradient estimate

Recall: for $\beta \in [0,1)$, discounted value of state i is

$$J_\beta(i) = \mathbf{E}\left[r(X_0) + \beta r(X_1) + \beta^2 r(X_2) + \cdots \middle| X_0 = i\right].$$

Hence,

$$\begin{aligned}
\nabla\eta(\theta) &= \nabla\pi'(\theta) r \\
&= \nabla\pi'(\theta)(J_\beta(\theta) - \beta P(\theta) J_\beta(\theta)) && \text{(Bellman)} \\
&= \nabla\pi' J_\beta - \beta(\nabla\pi' - \pi'\nabla P) J_\beta && \text{(since } \pi' = \pi' P\text{)} \\
&= (1-\beta)\nabla\pi' J_\beta + \beta\pi'\nabla P J_\beta \\
&= (1-\beta)\nabla\pi' J_\beta + \underbrace{\beta\nabla_\beta\eta(\theta)}_{\text{biased gradient estimate}}
\end{aligned}$$

A biased gradient estimate

Theorem: For all $\beta \in [0, 1)$,

$$\nabla\eta = \beta\pi'\nabla P J_\beta + (1-\beta)\nabla\pi' J_\beta.$$

$$= \beta \underbrace{\nabla_\beta \eta}_{\text{biased gradient estimate}} + \underbrace{(1-\beta)\nabla\pi' J_\beta}_{\to 0 \text{ as } \beta\to 1}.$$

Notice:

$$\lim_{\beta\to 1}(1-\beta)\nabla\pi' J_\beta = \nabla\pi' \lim_{\beta\to 1}(1-\beta)\sum_{t=0}^{\infty}(\beta P)^t r$$
$$= \nabla\pi' e\pi' r$$
$$= \eta\nabla\pi' e = \eta\nabla(1) = 0.$$

Quantifying the bias

The bias of the gradient estimate $\nabla_\beta \eta$ is

$$\nabla \pi'(1-\beta)\sum_{t=0}^{\infty}(\beta P)^t r,$$

and $(1-\beta)\sum_{t=0}^{\infty}(\beta P)^t$ is a convex combination of $P, P^2, P^3, \ldots$

Thus, the bias is determined by how rapidly P^t approaches $e\pi'$.

Recall:

- If the finite Markov chain $(\mathcal{S}, P(\theta))$ is *ergodic* (ie: no transient states, aperiodic), then it has a unique stationary distribution, π, and the Markov chain mixes, that is, the distribution of X_t approaches π.

How rapidly?

Mixing Time of Markov Chains

- ℓ_1-distance: If $q, \bar{q}$ are distributions on the states, $\|q - \bar{q}\|_1 := \sum_{i=1}^n |q(i) - \bar{q}(i)|$

- $\underline{d}(t)$-distance: Let $p^t(i)$ be the distribution over states at time t, starting from state i, that is, $p^t(i)$ is the ith row of P^t.

$$\underline{d}(t) := \max_{i,j} \|p^t(i) - p^t(j)\|_1$$

- Unique stationary distribution $\Rightarrow \underline{d}(t) \to 0$.

- **Mixing time**:

$$\tau^* := \min \left\{ t \geq 1 \colon \underline{d}(t) \leq e^{-1} \right\}$$

Can show that $\underline{d}(t)$ decreases monotonically, so

$$\max_{i,j} \left\| p^t(i) - p^t(j) \right\|_1 \leq e^{-\lfloor t/\tau^* \rfloor}.$$

Bias and Mixing Time

$$\max_{i,j} \left\| p^t(i) - p^t(j) \right\|_1 \le e^{-\lfloor t/\tau^* \rfloor},$$

and so all rows $\tilde{p}^t(i)$ of $P^t - e\pi'$ have

$$\left\| \tilde{p}^t(i) \right\|_1 \le e^{-\lfloor t/\tau^* \rfloor}.$$

Theorem: For all $\beta \in [0,1)$, $\theta \in \mathbb{R}^k$,

$$\|\nabla \eta(\theta) - \nabla_\beta \eta(\theta)\| \le \text{constant} \times \tau^*(\theta)(1-\beta).$$

$$\frac{1}{1-\beta} \gg \tau^*(\theta) \qquad \Rightarrow \qquad \nabla_\beta \eta(\theta) \approx \nabla \eta(\theta).$$

Estimating $\nabla_\beta \eta(\theta)$: GPOMDP

Given: parameterized policies, $\mu_u(\theta, y)$, $\beta \in [0, 1)$:

1. Set $z_0 = \Delta_0 = 0 \in \mathbb{R}^K$.
2. for each observation Y_t, control U_t, reward $r(X_{t+1})$ do
   - Set $z_{t+1} = \beta z_t + \dfrac{\nabla \mu_{U_t}(\theta, Y_t)}{\mu_{U_t}(\theta, Y_t)}$ (eligibility trace)
   - Set $\Delta_{t+1} = \Delta_t + \frac{1}{t+1}\left[r(X_{t+1}) z_{t+1} - \Delta_t\right]$

Convergence of GPOMDP

GPOMDP computes
$$\Delta_T = \frac{1}{T}\sum_{t=0}^{T-1}\sum_{s=0}^{t}\beta^s \frac{\nabla\mu_{U_{t-s}}}{\mu_{U_{t-s}}} r(X_{t+1})$$
$$= \frac{1}{T}\sum_{t=0}^{T-1}\frac{\nabla\mu_{U_t}}{\mu_{U_t}}\sum_{s=0}^{T-t-1}\beta^s r(X_{t+s+1}).$$

Recall: $\nabla_\beta \eta(\theta) = \beta\pi'\nabla P J_\beta = \beta\pi'$ "$\nabla P/P$" PJ_β.

Theorem: For all $\beta \in [0,1)$, $\theta \in \mathbb{R}^K$,
$$\lim_{t\to\infty}\beta\Delta_t \stackrel{\text{a.s.}}{=} \nabla_\beta\eta(\theta).$$

Explanation of GPOMDP

Algorithm computes:

$$\Delta_T = \frac{1}{T}\sum_{t=0}^{T-1} \frac{\nabla \mu_{U_t}}{\mu_{U_t}} \underbrace{\left(r(X_{t+1}) + \beta r(X_{t+2}) + \cdots + \beta^{T-t-1} r(X_T)\right)}_{\text{Estimate of value of next state, } J_\beta(X_{t+1})}$$

- $\nabla \mu_{U_t}$ is the direction to increase the probability of the action U_t.
- It is weighted by an estimate of the expected discounted subsequent rewards, and
- Divided by μ_{U_t}: ensures "popular" actions don't dominate .

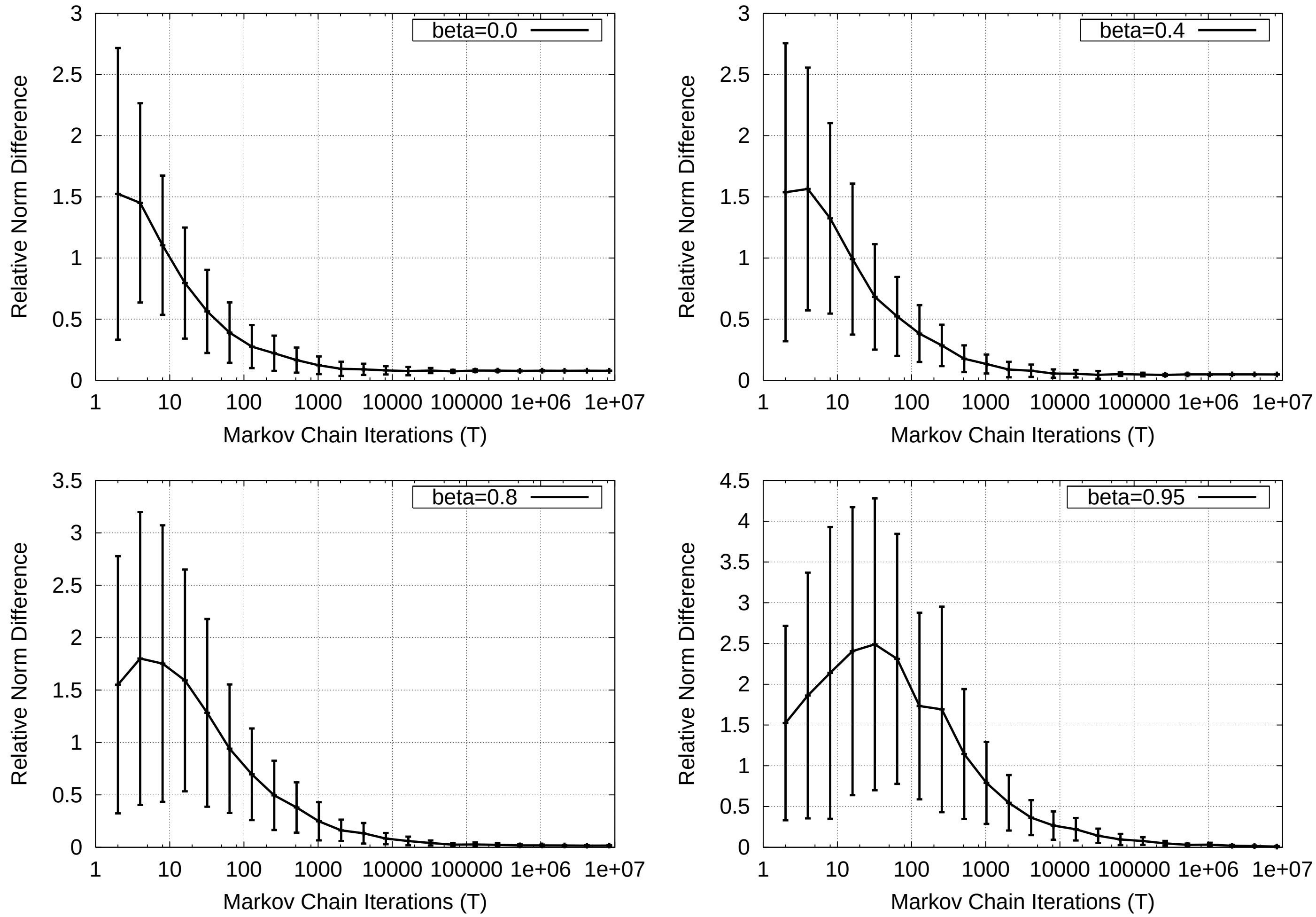
GPOMDP: Bias/Variance trade-off

$$\boxed{\Delta_t(\theta) \xrightarrow{t\to\infty} \nabla_\beta \eta(\theta) \xrightarrow{\beta\to 1} \nabla \eta(\theta)}.$$

Bias/Variance Tradeoff: $\beta \approx 1$ gives:

- Accurate gradient approximation ($\nabla_\beta \eta$ close to $\nabla \eta$), but
- Large variance in estimates Δ_t of $\nabla_\beta \eta$ for small t.
- Bias proportional to $\tau^*(1-\beta)$, variance proportional to $\tau^*/(1-\beta)^2$.

Bias/Variance trade-off: $\|\Delta_T - \nabla\eta\| / \|\nabla\eta\|$ versus T



Gradient Ascent

How to use the estimate $\Delta_T(\theta)$ of $\nabla_\beta \eta(\theta)$?

1. Stochastic gradient ascent:

- Could set $\theta_{t+1} := \theta_t + \gamma_t \Delta_T$ for small step sizes γ_t.
- Set $z_0 = 0 \in \mathbb{R}^K$. Loop forever:

$$
\begin{aligned}
z_{t+1} &= z_t + \beta_t \frac{\nabla \mu_{U_t}(\theta, Y_t)}{\mu_{U_t}(\theta, Y_t)} \\
\theta_{t+1} &= \theta_t + \gamma_{t+1} r(X_{t+1}) z_{t+1}
\end{aligned}
$$

- These stochastic approximation algorithms can be **very** slow.

Gradient Ascent

How to use the estimate $\Delta_T(\theta)$ of $\nabla_\beta \eta(\theta)$?

2. Line search:

- Search for a local maximum in direction Δ_T.
- e.g., find $\theta_1, \theta_2, \theta_3$ in direction $\Delta_T(\theta)$ such that $\eta(\theta_1) < \eta(\theta_2) > \eta(\theta_3)$.
- Problem: $\eta(\theta)$ is avaiable only via noisy simulation.
- Typically, line-search terminates before finding the maximum.

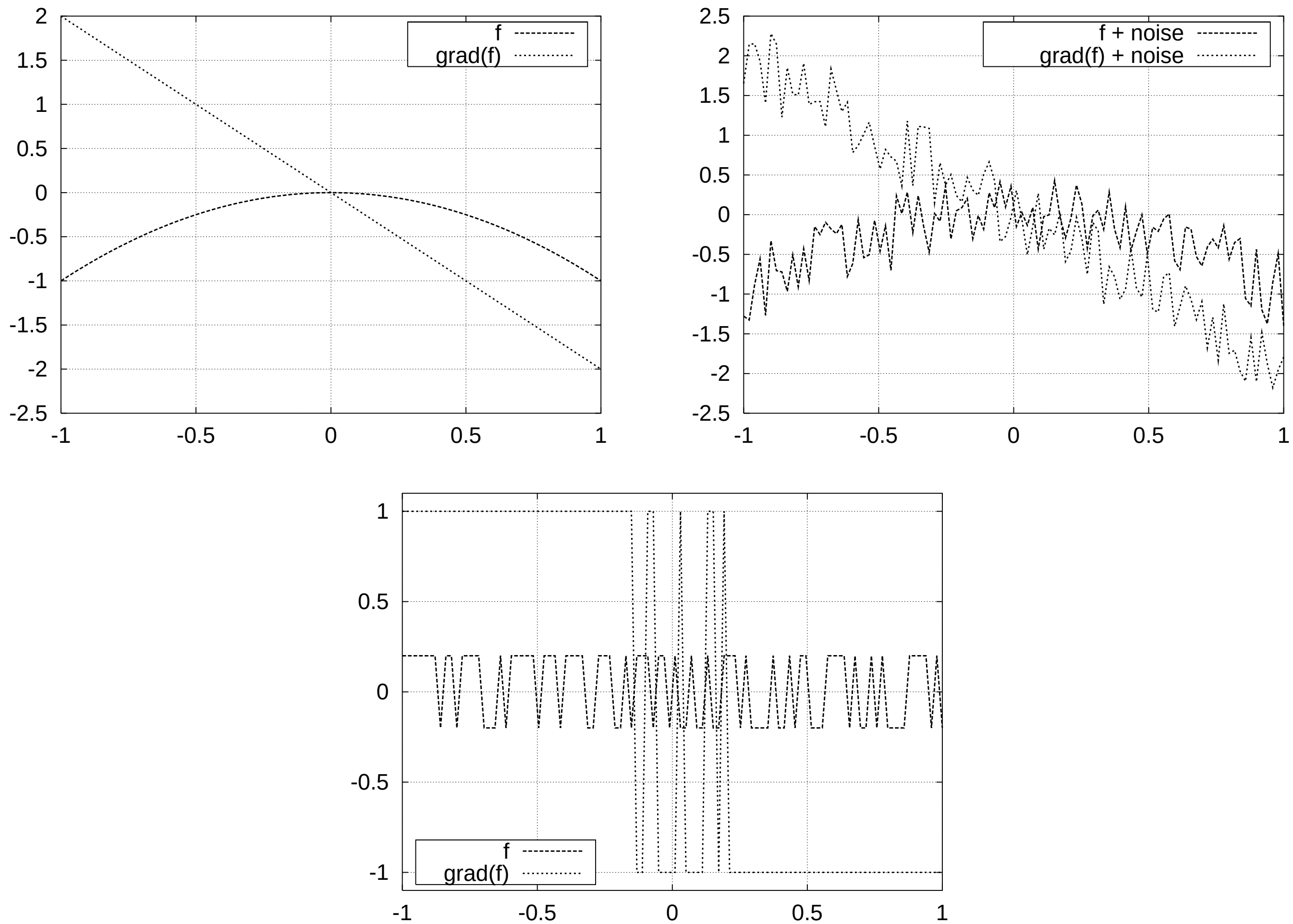
Line-search in the presence of noise

- Solution: Use gradients, not values, to bracket the maximum.
- Find θ_1, θ_2 in direction $\Delta_T(\theta)$ such that

$$\Delta_T(\theta_1) \cdot \Delta_T(\theta) > 0, \quad \Delta_T(\theta_2) \cdot \Delta_T(\theta) < 0$$

- Very robust to noise, if noise variances in gradient estimates and value estimates are similar. (Reasonable assumption when both are based on the same Markov process.)
- Enables the use of conjugate-gradient methods.

Line-search in the presence of noise



Monte-Carlo Gradient Estimates without Recurrent States

Summary

- GPOMDP gives a biased gradient estimate, without recurrent states.
- Parameter $\beta \in [0, 1)$ versus mixing time: bias/variance trade-off.
- Gradient-based line-search.

Policy Gradient Methods

1. Motivation: Performance Degradation with Value Function Methods
2. POMDPs and the Induced Markov Chain
3. Monte-Carlo Gradient Estimates with Recurrent States
4. Monte-Carlo Gradient Estimates without Recurrent States
5. Variance Reduction Techniques

Variance Reduction in Monte-Carlo Gradient Estimates

GPOMDP computes:

$$\Delta_T = \frac{1}{T} \sum_{t=0}^{T-1} \frac{\nabla \mu_{U_t}}{\mu_{U_t}} \left(r(X_{t+1}) + \beta r(X_{t+2}) + \cdots + \beta^{T-t-1} r(X_T) \right)$$

to estimate $\mathbf{E}_\pi \dfrac{\nabla \mu_{U_0}}{\mu_{U_0}} J_\beta(X_1)$.

This is a Monte-Carlo approximation of an integral.

Variance Reduction in Monte-Carlo Gradient Estimates

- Estimate $\int f \, dP$ using $\frac{1}{T} \sum_{t=1}^{T} f(X_t)$.
- **Additive control variate**: $\int f \, dP = \int (f - \phi) \, dP + \int \phi \, dP$.
- If $\int \phi \, dP$ is known, we replace f by $f - \phi$ in the estimate.
- If f and ϕ are **strongly correlated**, the variance is reduced:

$$\mathbf{Var}(f - \phi) = \mathbf{Var}(f) - 2\mathrm{Cov}(f, \phi) + \mathbf{Var}(\phi).$$

Control Variates for GPOMDP

$$\Delta_T = \frac{1}{T} \sum_{t=0}^{T-1} \frac{\nabla \mu_{U_t}(Y_t)}{\mu_{U_t}(Y_t)} \underbrace{\left(r(X_{t+1}) + \beta r(X_{t+2}) + \cdots + \beta^{2T-t-1} r(X_{2T})\right)}_{J_{t+1}}$$

- Baseline: $\dfrac{\nabla \mu}{\mu} \left(J_{t+1} - A(Y_t)\right)$.

  e.g. replace $r(X_t)$ by $r(X_t) - \mathbf{E}_\pi r$.
  (Then $A = \mathbf{E}_\pi J_\beta(X_{t+1})$, where $J_\beta(j) = \mathbf{E}(J_t | X_t = j)$.)

- Actor-critic: $\dfrac{\nabla \mu}{\mu} \tilde{J}(Y_{t+1})$.

Baseline Control Variates for GPOMDP

- **Any** $A : \mathcal{Y} \to \mathbb{R}$ **is unbiased**. (So we need only consider variance.)
- $$\mathbf{Var}\left(\frac{1}{T}\sum_{t=0}^{T-1}\frac{\nabla\mu}{\mu}\left(J_{t+1} - A(Y_t)\right)\right) \leq$$
$$\frac{\Omega}{T}\mathbf{Var}_\pi\left(\frac{\nabla\mu}{\mu}\left(J_\beta(X_{t+1}) - A(Y_t)\right)\right)$$
$$+ \frac{\Omega}{T}\mathbf{E}_\pi\left(\frac{\nabla\mu}{\mu}\left(J_{t+1} - J_\beta(X_{t+1})\right)\right)^2 + \left(\frac{\Omega}{T} + 1\right)\frac{C}{(1-\beta)^2}\beta^T.$$

Baseline Control Variates for GPOMDP

Optimal baseline: $$A(y) = \frac{\mathbf{E}\left[\left(\frac{\nabla\mu}{\mu}\right)^2 J_\beta \middle| y\right]}{\mathbf{E}\left[\left(\frac{\nabla\mu}{\mu}\right)^2 \middle| y\right]}$$

Optimal constant baseline: $$A = \frac{\mathbf{E}\left[\left(\frac{\nabla\mu}{\mu}\right)^2 J_\beta\right]}{\mathbf{E}\left(\frac{\nabla\mu}{\mu}\right)^2}$$

Excess variance with $\mathbf{E}J_\beta$: $$\frac{\left(\mathbf{E}\left(\frac{\nabla\mu}{\mu}\right)^2 \mathbf{E}J_\beta - \mathbf{E}\left[\left(\frac{\nabla\mu}{\mu}\right)^2 J_\beta\right]\right)^2}{\mathbf{E}\left(\frac{\nabla\mu}{\mu}\right)^2}$$

Another Variance Reduction Technique: Recycling a Random Seed

- Estimate $\int (f - g)\, dP$ using $\dfrac{1}{T}\sum_{t=1}^{T}(f(X_t) - g(X_t))$.
- Or should we use two sequences, $X_1, \ldots, X_T$ and $X'_1, \ldots, X'_T$?

  That is, $\dfrac{1}{T}\sum_{t=1}^{T}(f(X_t) - g(X'_t))$?
- If f and g are **strongly correlated**, the variance is reduced by using the same sequence:

$$\mathbf{Var}(f - g) = \mathbf{Var}(f) - 2\mathrm{Cov}(f, g) + \mathbf{Var}(g).$$

Recycling a Random Seed for a POMDP: PEGASUS

We can rewrite the update equations for a POMDP as deterministic equations involving a random sequence:

$$\begin{aligned} \Pr(Y_t = y_t | X_t = x_t) &= \nu_{y_t}(x_t) & \qquad Y_t &= f_\nu(X_t, B_t) \\ \Pr(U_t = u_t | Y_t = y_t) &= \mu_{u_t}(y_t) & U_t &= f_\mu(Y_t, B_t) \\ \Pr(X_{t+1} = x_{t+1} | X_t, U_t) &= p_{x_t x_{t+1}}(u_t) & X_{t+1} &= f_p(U_t, X_t, B_t), \end{aligned}$$

where the r.v.s $(B_0, B_1, B_2, \ldots)$ are uniform $[0, 1]$.

If we redefine the state as $(X_t, (B_t, B_{t+1}, \ldots))$, this gives a deterministic version of our POMDP.

Recycling a Random Seed for a POMDP: PEGASUS

Using this notation, it is clear how to recycle the random sequence, for example, to compare estimates of the difference of value under two policies μ and μ':

$$\mathbf{E}J_\beta^\mu(X_0) - \mathbf{E}J_\beta^{\mu'}(X_0) = \int_{b_0,b_1,\ldots} \left(\sum_{t=1}^{\infty} \beta^t r(x_t) - \sum_{t=1}^{\infty} \beta^t r(x'_t) \right),$$

where x_t and x_t are the states generated by the deterministic POMDPS with the same random sequence $b_0, b_1, \ldots$ and different policies μ and μ'.

Recycling a Random Seed for a POMDP: PEGASUS

- Fixed random sequence (i.e., fixed seed) implies value/average reward are deterministic functions of policy.
- Thus, we can directly optimize over policies:

$$\text{approximate } \arg\max_{\mu} \mathbf{E} J_{\beta}^{\mu}(X_0) \text{ by } \arg\max_{\mu} \frac{1}{n} \sum_{s=1}^{n} \sum_{t=1}^{\infty} \beta^t r\left(x_{\mu,t}^{s}\right),$$

  where $x_{\mu,t}^{s}$ is the sth state sequence generated by deterministic POMDP with a fixed random sequence and given policy.
- We need sample averages uniformly (over μ) close to expectations: 'simple' policy classes (small combinatorial dimension), 'smooth' POMDPs (Lipschitz state transitions, rewards).

Policy Gradient Methods

1. Motivation: Performance Degradation with Value Function Methods
2. POMDPs and the Induced Markov Chain
3. Monte-Carlo Gradient Estimates with Recurrent States
4. Monte-Carlo Gradient Estimates without Recurrent States
5. Variance Reduction Techniques