Recap: Policy Iteration and MDP

- The Policy Iteration begins with a (stationary) base policy $\mu^{(t)}$ and operates in two steps.

- <u>**Policy Evaluation step:**</u> We compute $J_{\mu^{(t)}}(1), \ldots, J_{\mu^{(t)}}(n)$ which solves the system of equations:

$$J_{\mu^{(t)}}(i) = \sum_{j=1}^{n} p_{ij}(\mu^{(t)}(i))\big(g(i, \mu^{(t)}(i), j) + \alpha J_{\mu^{(t)}}(j)\big)$$

  - This step, solves a "version" of the Bellman's Equation where we stick to base policy $\mu^{(t)}$.

  - This is a **linear** system on the variables $J_{\mu^{(t)}}(1), \ldots, J_{\mu^{(t)}}(n)$.

Recap Policy Iteration and MDP

- **<u>Policy Improvement step:</u>** We compute a new policy $\mu^{(t+1)}$ as:

$$\mu^{(t+1)}(i) \in \arg\min_{u \in U(i)} \left\{ \sum_{j=1}^{n} p_{ij}(u)\Big(g(i,u,j) + \alpha J_{\mu^{(t)}}(j)\Big) \right\}, \forall i \in \{1, \ldots, n\}$$

- Notice that this is similar to a 1-step lookahead minimization.

- So the Policy Improvement step, is essentially the Rollout Algorithm, where $\mu^{(t)}$ plays the role of the base policy and $\mu^{(t+1)}$ plays the role of the rollout policy.

- The PI Algorithm alternates between these two steps sequentially, until:

$$J_{\mu^{(t+1)}}(i) = J_{\mu^{(t)}}(i), \; \forall i \in \{1, \ldots, n\}$$

REINFORCE: Policy Gradient Algorithm

- At last, the algorithm known as the REINFORCE algorithm (or simple the Policy Gradient) is given as follows:

Algorithm 1 REINFORCE Algorithm (Policy Gradient)

Input: Initial DNN parameters $\theta^{(0)}$ and randomized policy $\tilde{\mu}(\theta^{(0)})$.

1: **for** $t = 0, ..., T$ **do** (obtaining new samples)

2: Collect S sample trajectories $z^s = (i_0^s, u_0^s, ..., i_M^s)$ using the policy $\tilde{\mu}(\theta^{(t)})$

3: Compute the policy gradient:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx \frac{1}{S}\sum_{s=1}^{S}\Bigg(\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\big)\Bigg)\Bigg(\sum_{k=0}^{M-1}\alpha^k g(i_k^s,u_k^s)+\alpha^M\hat{J}_M(i_m^s)\Bigg)$$

4: Perform the gradient step:

$$\theta^{(t+1)} = \theta^{(t)} - \gamma^{(t)}\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big)$$

5: **end for**

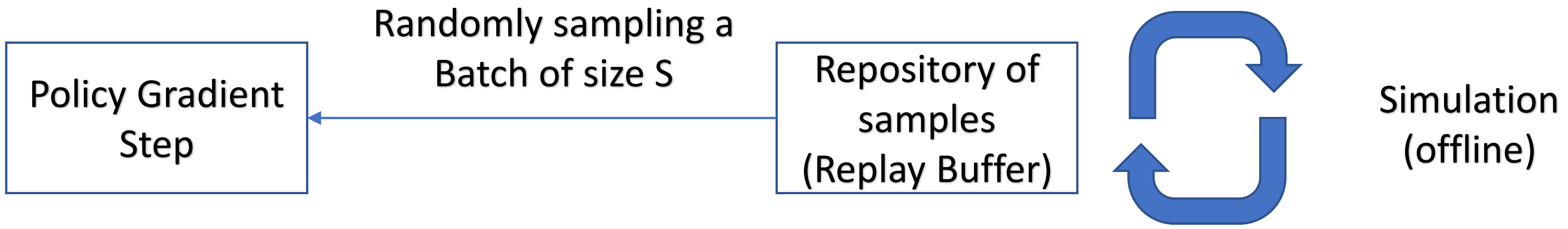
Output: The last DNN configuration $\theta^{(T)}$. A suboptimal policy $\tilde{\mu}(\theta^{(T)})$

Issues of Policy Gradient

- In addition the Policy Gradient (and the Critic-only) Algorithms are what is known as *on-policy algorithms:*
  - After every gradient-step we need to collect more samples with the updated policy.

- This fact can be very costly in practical problems, since between training steps we need to perform a lot of sampling.

- Ideally, we would like to do something like we did in the DQN, using a *Replay Buffer:*

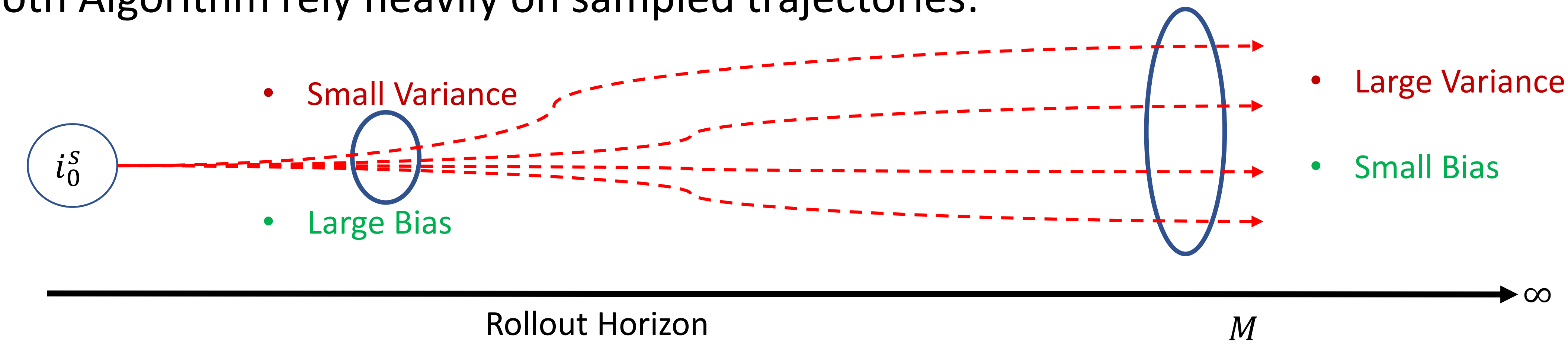


- We will study how to do so next time.

Issues of Policy Gradient

- We end the lecture some question regarding the Policy Gradient and the Critic-only Algorithm presented so far.

- Both Algorithm rely heavily on sampled trajectories:



- We need to address the Bias-Variance Trade-off.

- We will study how to address this issue and how to combine both algorithm into the Actor-Critic Algorithm (which is also a framework).

Reducing the Variance of Policy Gradient

- Let's start by addressing the variance issue of Policy gradient. We will present two modifications to the policy gradient:
  - Use causality
  - Use baselines

- Let's write once again the policy gradient, in the expectation form:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) = \mathbb{E}_{p(z|\theta)}\Bigg[\Bigg(\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)\Bigg)\Bigg(\sum_{k=0}^{M-1}\alpha^k g(i_k,u_k)+\alpha^M\hat{J}_M(i_m)\Bigg)\Bigg]$$

- Let's split the cost term in two:

$$C_k^M = \sum_{j=k}^{M}\alpha^j g(i_j,u_j)+\alpha^M\hat{J}_M(i_m) \qquad\qquad C_0^{k-1} = \sum_{j=0}^{k-1}\alpha^j g(i_j,u_j)$$

Reducing the Variance of Policy Gradient

- Then we can write the policy gradient as:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) = \mathbb{E}_{p(z|\theta)}\Bigg[\Bigg(\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)\Bigg)\Bigg(C_0^{k-1}+C_k^M\Bigg)\Bigg]$$

- Distributing the sum and the expectation we get:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) = \mathbb{E}_{p(z|\theta)}\Bigg[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)C_0^{k-1}\Bigg] + \mathbb{E}_{p(z|\theta)}\Bigg[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)C_k^M\Bigg]$$

- Now we will show that :

$$\mathbb{E}_{p(z|\theta)}\Bigg[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)C_0^{k-1}\Bigg] = 0$$

Reducing the Variance of Policy Gradient

- To that end notice that C_0^{k-1} only depends on $(i_0, u_0, \ldots, i_{k-1}, u_{k-1})$.

- Let's putting the expectation inside:

$$\mathbb{E}_{p(z|\theta)}\Big[\sum_{k=0}^{M-1} \nabla_\theta\big(\ln(p(u_k|i_k, \theta^{(t)}))\big)C_0^{k-1}\Big] = \Bigg(\sum_{k=0}^{M-1} \mathbb{E}_{p(z|\theta)}\Big[\nabla_\theta\big(\ln(p(u_k|i_k, \theta^{(t)}))\big)C_0^{k-1}\Big]\Bigg)$$

- Now we apply the Markov property. Let's focus on term with $k = 1$:

$$\mathbb{E}_{p(z|\theta)}\Big[\nabla_\theta\big(\ln(p(u_1|i_1, \theta^{(t)}))\big)C_0^0\Big] = \sum_{u_1}\sum_{i_1}\sum_{u_0}\sum_{i_0} \nabla_\theta\big(\ln(p(u_1|i_1, \theta^{(t)}))\big)p(u_1|i_1, \theta^{(t)})p_{i_0,i_1}(u_0)p(u_0|i_0, \theta^{(t)})p(i_0)C_0^0 =$$

$$\sum_{u_1}\sum_{i_1} \nabla_\theta\big(\ln(p(u_1|i_1, \theta^{(t)}))\big)p(u_1|i_1, \theta^{(t)}) \sum_{u_0}\sum_{i_0} p_{i_0,i_1}(u_0)p(u_0|i_0, \theta^{(t)})p(i_0)C_0^0 =$$

Reducing the Variance of Policy Gradient

- Now using the log-trick $\nabla \ln p = \frac{\nabla p}{p}$:

$$\sum_{u_1}\sum_{i_1}\nabla_\theta\big(\ln(p(u_1|i_1,\theta^{(t)}))\big)p(u_1|i_1,\theta^{(t)})\sum_{u_0}\sum_{i_0}p_{i_0,i_1}(u_0)p(u_0|i_0,\theta^{(t)})p(i_0)C_0^0 =$$

$$\sum_{u_1}\sum_{i_1}\nabla_\theta\big(p(u_1|i_1,\theta^{(t)})\big)\sum_{u_0}\sum_{i_0}p_{i_0,i_1}(u_0)p(u_0|i_0,\theta^{(t)})p(i_0)C_0^0 =$$

$$\sum_{u_1}\sum_{i_1}\nabla_\theta\big(p(u_1|i_1,\theta^{(t)})\big)\sum_{u_0}\sum_{i_0}p(i_1,i_0,u_0|\theta^{(t)})C_0^0$$

- Now by conditioning on i_1 and packing the two sums w.r.t. (i_0, u_0):

$$\sum_{u_1}\sum_{i_1}\nabla_\theta\big(p(u_1|i_1,\theta^{(t)})\big)p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\left[C_0^0|i_1,\theta^{(t)}\right]$$

- Now we have to bring the gradient outside the sum, by “reverting” the product rule.

Reducing the Variance of Policy Gradient

$$\sum_{u_1}\sum_{i_1}\nabla_\theta\big(p(u_1|i_1,\theta^{(t)})\big)p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big] =$$

$$\nabla_\theta\big(\sum_{u_1}\sum_{i_1}p(u_1|i_1,\theta^{(t)})p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\big) - \sum_{u_1}\sum_{i_1}p(u_1|i_1,\theta^{(t)})\nabla_\theta\big(p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\big) =$$

$$\nabla_\theta\big(\sum_{i_1}p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\big) - \sum_{i_1}\nabla_\theta\big(p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\big) =$$

$$\sum_{i_1}\nabla_\theta\big(p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\big) - \sum_{i_1}\nabla_\theta\big(p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\big) = 0$$

- Now we can repeat this for every $0 \leq k \leq M-1$.

Reducing the Variance of Policy Gradient

- Hence the policy gradient can be written as:

$$\nabla_\theta\left(\mathbb{E}_{p(z|\theta^{(t)})}\left[F(z)\right]\right) = \mathbb{E}_{p(z|\theta)}\left[\sum_{k=0}^{M-1} \nabla_\theta\left(\ln(p(u_k|i_k, \theta^{(t)}))\right) C_k^M\right] \quad C_k^M = \sum_{j=k}^{M} \alpha^j g(i_j, u_j) + \alpha^M \hat{J}_M(i_m)$$

- Now, by using Sample Averaging Approximation (SAA):

$$\nabla_\theta\left(\mathbb{E}_{p(z|\theta^{(t)})}\left[F(z)\right]\right) \approx \frac{1}{S}\sum_{s=1}^{S}\left(\sum_{k=0}^{M-1} \nabla_\theta\left(\ln(p(u_k^s|i_k^s, \theta^{(t)}))\right) \underbrace{\left(\sum_{j=k}^{M-1} \alpha^j g(i_j^s, u_j^s) + \alpha^M \hat{J}_M(i_m^s)\right)}_{\text{smaller variance}}\right)$$

- Let’s compare to what it was before:

$$\nabla_\theta\left(\mathbb{E}_{p(z|\theta^{(t)})}\left[F(z)\right]\right) \approx \frac{1}{S}\sum_{s=1}^{S}\left(\sum_{k=0}^{M-1} \nabla_\theta\left(\ln(p(u_k^s|i_k^s, \theta^{(t)}))\right)\left(\sum_{j=0}^{M-1} \alpha^j g(i_j^s, u_j^s) + \alpha^M \hat{J}_M(i_m^s)\right)\right)$$

Reducing the Variance of Policy Gradient

- We can write in a compact way:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}[F(z)]\big) = \mathbb{E}_{p(z|\theta^{(t)})}\left[\sum_{k=0}^{M-1} \nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)\right]$$

This is the cost-to-go from i_k with terminal cost approximation at the truncation M

- Now we can also add a baseline vector b to further reduce the variance:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}[F(z)]\big) = \mathbb{E}_{p(z|\theta^{(t)})}\left[\sum_{k=0}^{M-1} \nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\Big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k) - b\Big)\right]$$

- Notice that:

$$\mathbb{E}_{p(z|\theta^{(t)})}\left[\sum_{k=0}^{M-1} \nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)\right] = \sum_{z\in Z}\sum_{k=0}^{M-1} \nabla_\theta\big(p(u_k|i_k,\theta^{(t)})\big) = \nabla_\theta\Big(\sum_{z\in Z} p(z|\theta^{(t)})\Big) = 0$$

- This is direct by applying the log-trick again and taking the gradient out of the sum.

Reducing the Variance of Policy Gradient

- We want to pick a baseline thar reduces the variance as much as possible when we sample.

- The policy gradient is an expectation. So we can write variance as:

$$Var = \mathbb{E}_{p(z|\theta^{(t)})}\left[\left(\sum_{k=0}^{M-1} g(i_k, u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k) - b\big)\right)^2\right] - \mathbb{E}_{p(z|\theta^{(t)})}\left[\sum_{k=0}^{M-1} g(i_k, u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k) - b\big)\right]^2$$

$$g_k(i_k, u_k|\theta^{(t)}) = \nabla_\theta\big(\ln(p(u_k|i_k, \theta^{(t)}))\big)$$

- We want the variance to be as small as possible, so we take the derivative w.r.t. b and set it zero.

Reducing the Variance of Policy Gradient

$$Var = \mathbb{E}_{p(z|\theta^{(t)})}\left[\left(\sum_{k=0}^{M-1} g(i_k, u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k) - b\big)\right)^2\right] - \underbrace{\mathbb{E}_{p(z|\theta^{(t)})}\left[\sum_{k=0}^{M-1} g(i_k, u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k) - b\big)\right]^2}_{\mathbb{E}_{p(z|\theta^{(t)})}\left[\sum_{k=0}^{M-1} g(i_k, u_k|\theta^{(t)})\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)\right]}$$

(Baseline does not change expectation)

$$\frac{\partial Var}{\partial b} = -2\mathbb{E}_{p(z|\theta^{(t)})}\left[\left(\sum_{k=0}^{M-1} g(i_k, u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)\big)\right)\left(\sum_{k=0}^{M-1} g(i_k, u_k|\theta^{(t)})\right)\right] + 2b\mathbb{E}_{p(z|\theta^{(t)})}\left[\left(\sum_{k=0}^{M-1} g(i_k, u_k|\theta^{(t)})\right)^2\right] = 0$$

$$b^* = \frac{\mathbb{E}_{p(z|\theta^{(t)})}\left[\left(\sum_{k=0}^{M-1} g(i_k, u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)\big)\right)\left(\sum_{k=0}^{M-1} g(i_k, u_k|\theta^{(t)})\right)\right]}{\mathbb{E}_{p(z|\theta^{(t)})}\left[\left(\sum_{k=0}^{M-1} g(i_k, u_k|\theta^{(t)})\right)^2\right]}$$

Reducing the Variance of Policy Gradient

- Using SAA:

a vector!

$$b^* \approx \frac{\sum_{s=1}^{S}\left(\sum_{k=0}^{M-1} g(i_k^s, u_k^s|\theta^{(t)})(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k^s))\right)\left(\sum_{k=0}^{M-1} g(i_k^s, u_k^s|\theta^{(t)})\right)}{\sum_{s=1}^{S}\left(\sum_{k=0}^{M-1} g(i_k^s, u_k^s|\theta^{(t)})\right)^2}$$

a vector!

- This is the “best” possible baseline for reducing the variance.

- But, in practice, people use a simpler baseline:

$$\bar{b} = \frac{1}{S}\sum_{s=1}^{S}\left(\sum_{j=0}^{M-1} \alpha^j g(i_j^s, u_j^s) + \alpha^M \hat{J}_M(i_m^s)\right) \Rightarrow \boxed{\text{Average Cost}}$$

Introducing bias

- We touch now a very subtle point, which lies in purposefully introducing bias in order to improve the behavior of policy gradient.

- Let's write the policy gradient with baseline:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}[F(z)]\big) \approx \frac{1}{S}\sum_{s=1}^{S}\Bigg(\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)})\big)\Big(\sum_{j=k}^{M-1}\alpha^j g(i_j^s,u_j^s)+\alpha^M\hat{J}_M(i_m^s)-b\Big)\Bigg)$$

- By manipulating the $\alpha's$ terms we can write an equivalent form of the policy gradient:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}[F(z)]\big) \approx \frac{1}{S}\sum_{s=1}^{S}\Bigg(\sum_{k=0}^{M-1}\alpha^k\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)})\big)\Big(\sum_{j=k}^{M-1}\alpha^{j-k} g(i_j^s,u_j^s)+\alpha^{M-k}\hat{J}_M(i_m^s)-b\Big)\Bigg)$$

Introducing bias

- Note that if M is large, and α small, then many terms on this gradient will vanish.

- One approach done in practice to handle that is to introduce bias by letting:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx \frac{1}{S}\sum_{s=1}^{S}\Bigg(\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\big(\sum_{j=k}^{M-1}\alpha^{j-k}g(i_j^s,u_j^s)+\alpha^{M-k}\hat{J}_M(i_m^s)-b\big)\Bigg)$$

- This biased version can be seen as the gradient of an average-cost infinite-horizon DP with a dampening to reduce variance in the cost-to-go values.

- We have not covered average-cost infinite-horizon DP’s in the course. The analysis of biased policy gradient is a bit beyond our scope.
    - The main idea is given in “Bias in natural actor-critic algorithms. ICML 2014”, which we refer for further reading.

Combining the Critic and the Actor

- Now we are able to go towards the Actor-Critic Algorithm.

- Let's state again the altered policy gradient with baseline and using SAA:

$$\nabla_\theta \big(\mathbb{E}_{p(z|\theta^{(t)})} \big[F(z) \big] \big) \approx \frac{1}{S} \sum_{s=1}^{S} \sum_{k=0}^{M-1} \nabla_\theta \big(\ln(p(u_k^s | i_k^s, \theta^{(t)})) \Big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k^s) - b \Big) \Big]$$

- It turns out we can also define a state-dependent baseline $b(i_k)$:

$$\nabla_\theta \big(\mathbb{E}_{p(z|\theta^{(t)})} \big[F(z) \big] \big) \approx \frac{1}{S} \sum_{s=1}^{S} \sum_{k=0}^{M-1} \nabla_\theta \big(\ln(p(u_k^s | i_k^s, \theta^{(t)})) \Big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k^s) - b(i_k^s) \Big) \Big]$$

- And everything still holds fine (unbiased gradient and variance reduction)

Combining the Critic and the Actor

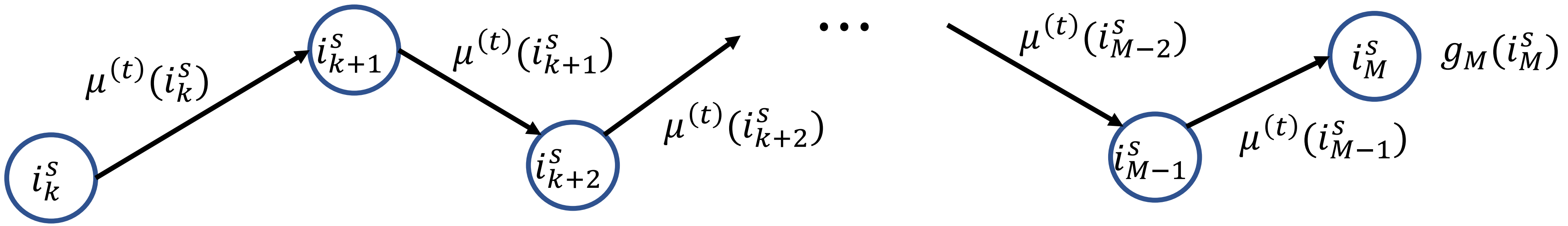
- Let's focus on the cost-to-go value:

$$\nabla_\theta\left(\mathbb{E}_{p(z|\theta^{(t)})}\left[F(z)\right]\right) \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=0}^{M-1}\nabla_\theta\Big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\Big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k^s) - b(i_k^s)\Big)\Big]$$

- This cost-to-go is obtained by the summation (with bias):

$$J_{\tilde{\mu}(\theta^{(t)})}(i_k^s) = \sum_{j=k}^{M-1}\alpha^{j-k}g(i_j^s,u_j^s) + \alpha^{M-k}\hat{J}_M(i_m^s)$$

A single trajectory "starting" from i_k^s and running for $M-k$ stages



Combining the Critic and the Actor

- Ideally we would like to use many trajectories to compute the cost-to-go:

- Then for each sampled pair (i_k^s, u_k^s) we can define the *true* Q-factor:

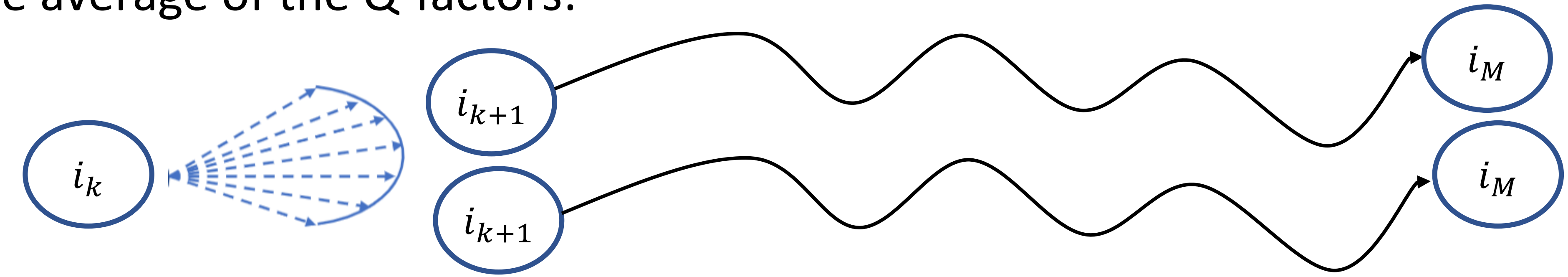
$$Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s, u_k^s) = g(i_k^s, u_k^s) + \mathbb{E}_{p(z|\theta^{(t)})}\left[\sum_{j=k+1}^{\infty} \alpha^{j-k} g(i_j, u_j) \mid i_k^s, u_k^s\right]$$

- And we would, then, use those in the policy gradient:

$$\nabla_\theta\left(\mathbb{E}_{p(z|\theta^{(t)})}\left[F(z)\right]\right) \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=0}^{M-1} \nabla_\theta\left(\ln(p(u_k^s|i_k^s, \theta^{(t)}))\left(Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s, u_k^s) - b(i_k^s)\right)\right]$$

Combining the Critic and the Actor

- However this is only the ideal case. We have to resort to sampling to obtain the a sample average of the Q-factors:



- For example, for a single sample we would return to the previous case:

$$\nabla_\theta\left(\mathbb{E}_{p(z|\theta^{(t)})}\left[F(z)\right]\right) \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=0}^{M-1}\nabla_\theta\left(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\left(\hat{Q}_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s)-b(i_k^s)\right)\right]$$

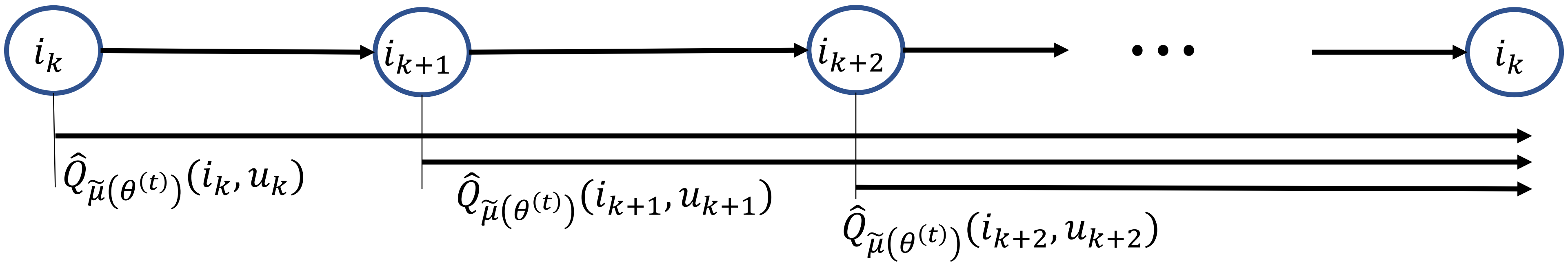
- Where:

$$\hat{Q}_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s)=\sum_{j=k}^{M-1}\alpha^{j-k}g(i_j^s,u_j^s)+\alpha^{M-k}\hat{J}_M(i_m^s)$$

A single sample of the Q-factor associated with pair (i_k^s, u_k^s)

Combining the Critic and the Actor

- But note that for each stage k, the horizon for every Q-factor is different!



- So we have to be careful:
  - The problem is infinite-horizon and we are using terminal cost approximations
  - for each stage k, there is a Q-factor that "starts" at stage k and goes until stage M

- Given the policy $\tilde{\mu}(\theta^{(t)})$, we know that the Q-factors are given by the Bellman's Equation:

$$Q_{\tilde{\mu}(\theta^{(t)})}(i,u) = \sum_{j=1}^{n} p_{ij}(\tilde{\mu}(\theta^{(t)})(i))\big(g(i,u,j) + \alpha Q_{\tilde{\mu}(\theta^{(t)})}(j, \tilde{\mu}(\theta^{(t)})(j))\big)$$

Combining the Critic and the Actor

- Now consider first this baseline (which only depends on the stage):

$$b(i_k^j) = \frac{1}{S}\sum_{s=1}^{S} Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s, u_k^s), \; \forall j \in \{1, ..., S\}$$

Averages all Q-values that start at stage k

- We can use a state-dependent baseline, by fixing the state:

$$b(i_k^s) = \sum_{u \in U(i)} p(u|i_k^s, \theta^{(t)}) Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s, u) = J_{\tilde{\mu}(\theta^{(t)})}(i_k^s)$$

The **true** value function(cost-to-go) from i_k^s

- If we use that, then the policy gradient becomes:

$$\nabla_\theta\left(\mathbb{E}_{p(z|\theta^{(t)})}\left[F(z)\right]\right) \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=0}^{M-1} \nabla_\theta\Big(\ln(p(u_k^s|i_k^s, \theta^{(t)}))\Big(\overbrace{Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s, u_k^s) - J_{\tilde{\mu}(\theta^{(t)})}(i_k^s)}^{\text{The }\textbf{true}\text{ quantities!}}\Big)\Big]$$

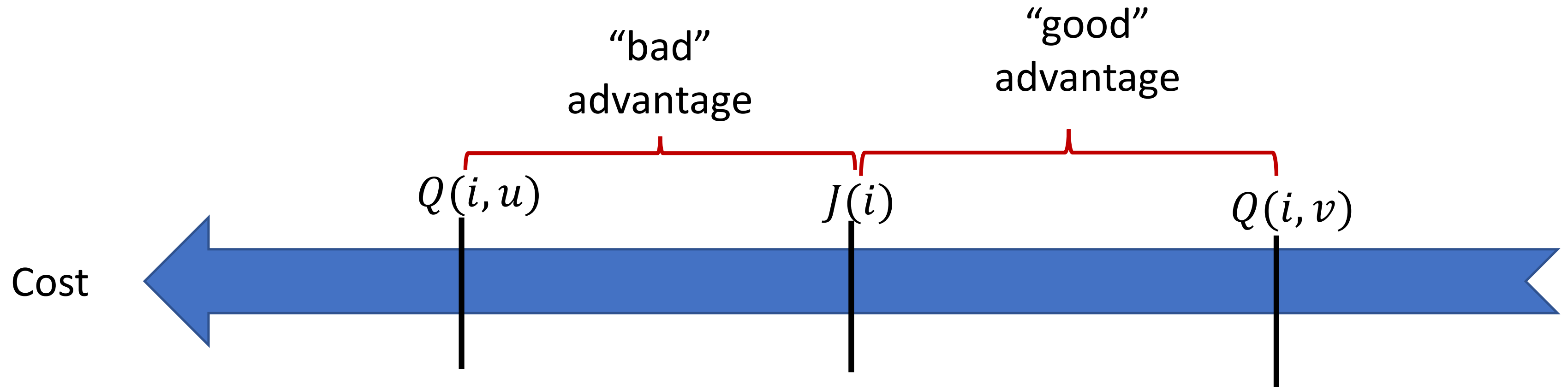
Combining the Critic and the Actor

- We can write the policy gradient in terms of the **Advantage:**

$$\nabla_\theta \left(\mathbb{E}_{p(z|\theta^{(t)})} [F(z)] \right) \approx \frac{1}{S} \sum_{s=1}^{S} \sum_{k=0}^{M-1} \nabla_\theta \left(\ln(p(u_k^s | i_k^s, \theta^{(t)})) \left(A_{\tilde{\mu}(\theta^{(t)})}(i_k^s, u_k^s) \right) \right]$$

$$A_{\tilde{\mu}(\theta^{(t)})}(i_k^s, u_k^s) = Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s, u_k^s) - J_{\tilde{\mu}(\theta^{(t)})}(i_k^s)$$

- The Advantage is a <u>**relative**</u> quantity!



- We can, in fact, re-write all algorithms covered so far in terms of the Advantage.

Combining the Critic and the Actor

- And more! We can, in fact, re-write the Advantage as follows:

$$A_{\tilde{\mu}(\theta^{(t)})}(i_k^s, u_k^s) = \underbrace{g(i_k^s, u_k^s) + \alpha \mathbb{E}_{p(i_{k+1}|\theta^{(t)})}\left[J_{\tilde{\mu}(\theta^{(t)})}(i_{k+1})\right]}_{Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s, u_k^s)} - J_{\tilde{\mu}(\theta^{(t)})}(i_k^s)$$

$$Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s, u_k^s) = g(i_k^s, u_k^s) + \mathbb{E}_{p(z|\theta^{(t)})}\left[\sum_{j=k+1}^{\infty} \alpha^{j-k} g(i_j, u_j) \mid i_k^s, u_k^s\right]$$

- The Critic comes in to **<u>approximate</u>** the Advantage.

Combining the Critic and the Actor

- Let's recall the critic problem, where now we use another DNN, say ϕ:

$$\phi^{(t)} = \arg\min_{\phi} \left\{ \sum_{l=1}^{L} \left(\tilde{J}(i_0^l, \phi) - \beta^l \right)^2 \right\}$$

$$\beta^l = \sum_{k=0}^{M-1} \alpha^k g(i_k^s, \mu^{(t)}(i_k^l), i_{k+1}^s) + \alpha^M \tilde{J}(i_M^s, \phi^{(t-1)})$$

- We have L samples: $(i_k^s, \hat{Q}(i_k^s, u_k^s))$
- So L is larger than S.
- Each sample has different horizon

- Solving the regression problem to (local) optimality in practice may be costly.

Combining the Critic and the Actor

- Then we can, instead perform a single gradient step:

$$\phi^{(t)} = \phi^{t-1} - \gamma^{(t)} \sum_{l=1}^{L} \nabla_\phi \tilde{J}(i_0^l, \phi)\big(\tilde{J}(i_0^l, \phi) - \beta^l\big)$$

- In practice, we can perform more than one step and recompute the labels after a few steps and then repeat, in a **optimistic version** of the critic-step with $\phi_0^{(t-1)} = \phi^{(t-1)}$:

$$\phi_{m+1}^{(t-1)} = \phi_m^{(t-1)} - \gamma_p^{(t-1)} \sum_{l=1}^{L} \nabla_\phi \tilde{J}(i_0^l, \phi)\big(\tilde{J}(i_0^l, \phi) - \beta_p^l\big), \quad \forall m \in \{0, ..., P\}$$

$$\beta_p^l = \sum_{k=0}^{M-1} \alpha^k g(i_k^s, \mu^{(t)}(i_k^l), i_{k+1}^s) + \alpha^M \tilde{J}(i_M^s, \phi_p^{(t-1)}), \quad \forall p \in \{0, ..., P\}$$

$$\phi^{(t)} \leftarrow \phi_P^{(t-1)}$$

Combining the Critic and the Actor

- So we are essentially **reusing** trajectories to obtain the labels $\beta^{l'}s$.
  - This is actually a problem as it limits exploration. We will push this back for now.

- In addition, the labels may have high variance.

- An alternative, is we can used a short-horizon version (low variance, high bias):

$$\beta^l = \sum_{k=0}^{M-1} \alpha^k g(i_k^s, \mu^{(t)}(i_k^l), i_{k+1}^s) + \alpha^M \tilde{J}(i_M^s, \phi^{(t-1)})$$

- Low bias
- High variance

Monte-Carlo Estimates

$$\beta^l = g(i_k^s, \mu^{(t)}(i_k^l), i_{k+1}^s) + \alpha \tilde{J}(i_{k+1}^s, \phi^{(t-1)})$$

- High bias
- Low variance

Bootstrap Estimates

Combining the Critic and the Actor

- Then after solving the regression problem and obtaining $\phi^{(t)}$ we can write the policy gradient:

$$\nabla_\theta\left(\mathbb{E}_{p(z|\theta^{(t)})}\left[F(z)\right]\right) \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=0}^{M-1}\nabla_\theta\left(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\left(\tilde{A}(i_k^s,u_k^s)\right)\right]$$

- Where the advantage can be estimated, via two methods as well:

$$\tilde{A}(i_k^s,u_k^s) = \sum_{j=k}^{M-1}\alpha^{j-k}g(i_j^s,u_j^s) + \alpha^{M-k}J(i_M,\phi^{(t)}) - J(i_k^s,\phi^{(t)})$$

Monte-Carlo Estimates

$$\tilde{A}(i_k^s,u_k^s) = g(i_k^s,u_k^s) + \alpha J(i_{k+1},\phi^{(t)}) - J(i_k^s,\phi^{(t)})$$

Bootstrap Estimates

- The **Actor-Critic Algorithm** then iterates:
  - Critic-Step: Improves the critic $\phi^{(t+1)} \leftarrow \phi^{(t)}$, via regression
  - Actor-Step: Improves the actor $\theta^{(t+1)} \leftarrow \theta^{(t)}$, via policy gradient

Actor-Critic Algorithm

Algorithm 1 Actor-Critic Algorithm

Input: Initial DNN parameters $\theta^{(0)}$, randomized policy $\tilde{\mu}(\theta^{(0)})$.
Input: Initial DNN parameters $\phi^{(0)}$, cost-go-go approximate function $\tilde{J}(\cdot, \phi^{(0)})$.

1: **for** $t = 0, ..., T$ **do** (obtaining new samples)
2: Collect S sample trajectories $z^s = (i_0^s, u_0^s, ..., i_M^s)$ using the policy $\tilde{\mu}(\theta^{(t)})$
3: Perform the **critic step**:

$$\phi_{m+1}^{(t)} = \phi_m^{(t)} - \gamma_p^{(t)} \sum_{l=1}^{L} \nabla_\phi \tilde{J}(i_0^l, \phi)\big(\tilde{J}(i_0^l, \phi) - \beta_p^l\big), \quad \forall m \in \{1, ..., P\}$$

$$\phi^{(t+1)} \leftarrow \phi_P^{(t)}$$

4: Evaluate the advantage $\tilde{A}(i_k^s, u_k^s)$ for every sample pair (i_k^s, u_k^s).
5: Compute the policy gradient:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\left[F(z)\right]\big) \approx \frac{1}{S} \sum_{s=1}^{S} \sum_{k=0}^{M-1} \nabla_\theta\big(\ln(p(u_k^s|i_k^s, \theta^{(t)}))\Big(\tilde{A}(i_k^s, u_k^s)\Big)\Big]$$

6: Perform the **actor-step** (gradient-step):

$$\theta^{(t+1)} = \theta^{(t)} - \gamma^{(t)} \nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)}}\left[F(z)\right]\big)$$

7: **end for**

Output: The last DNN configurations $\theta^{(T)}$ and $\phi^{(T)}$. A suboptimal policy $\tilde{\mu}(\theta^{(T)})$. An approximate cost-to-go function $\tilde{J}(\cdot, \phi^{(T)})$