IEOR 265 - Lecture 12
Actor-Critic Algorithm

1 Improving Policy Gradient

On the last lecture we presented a Critic-only algorithm, based on Regression, and an Actor-only algorithm, based on the policy gradient. In this lecture, we move towards the **Actor-Critic Algorithm** which uses approximations in both the critic step and the actor step. To that end, we start by addressing the issues of policy gradient. We state the REINFORCE algorithm here again:

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 $\tilde{\mu}(\theta^{(T)})$

And we note the policy gradient, in the same way as the critic-only algorithm, is an *on-policy* algorithm: it requires samples at every iteration and the samples are always generated by a policy, which cannot be reused in latter iterations. In addition the policy gradient also suffers from the bias-variance tradeoff: if the rollout horizon M is too long, there will be a lot of variance in the simulated costs. On the other hand, if the rollout horizon M is too short, there will be a lot of bias since the terminal cost approximation will be used "too soon". And we illustrate it once more in the following figure:

On this section we will focus primarily in addressing the bias-variance trade-off issue of policy gradient. We will do so by leveraging: (1) causality; (2) baselines. We start by writing again the policy gradient, in the expectation

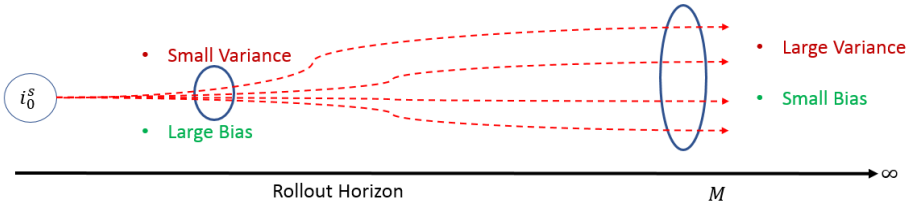

Figure 1: Schematic representation of the bias-variance trade-off faced by the critic-only algorithm: longer trajectories have less bias as costs are computed accumulated for longer, but have longer variance; while shorter horizon have low variance but hight bias.

form:

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

And we split the cost-term in two:

$$C_k^M = \sum_{j=k}^{M}\alpha^j g(i_j,u_j)+\alpha^M\hat{J}_M(i_m) \qquad C_0^{k-1} = \sum_{j=0}^{k-1}\alpha^j g(i_j,u_j) \quad (2)$$

Then we can write the policy gradient as:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) = \mathbb{E}_{p(z|\theta)}\Bigg[\Bigg(\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)\Bigg)\Big(C_0^{k-1}+C_k^M\Big)\Bigg] \quad (3)$$

Distributing the sum and the expectation we get:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) = \mathbb{E}_{p(z|\theta)}\Bigg[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)C_0^{k-1}\Bigg]+ \mathbb{E}_{p(z|\theta)}\Bigg[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)C_k^M\Bigg] \quad (4)$$

Now we will show that:

$$\mathbb{E}_{p(z|\theta)}\Bigg[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)C_0^{k-1}\Bigg] = 0 \quad (5)$$

Now we narrow our analysis to the term above. Notice that C_0^{k-1} only depends on $(i_0,u_0,...,i_{k-1},u_{k-1})$. We can exchange the expectation and the

summation (due to linearity):

$$\mathbb{E}_{p(z|\theta)}\Big[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)C_0^{k-1}\Big] = \Big(\sum_{k=0}^{M-1}\mathbb{E}_{p(z|\theta)}\Big[\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)C_0^{k-1}\Big]\Big) \tag{6}$$

Now let's focus on a single term of the summation. In particular the term with $k=1$. By applying the Markov Property we can write:

$$\mathbb{E}_{p(z|\theta)}\Big[\nabla_\theta\big(\ln(p(u_1|i_1,\theta^{(t)}))\big)C_0^{0}\Big] = \sum_{u_1}\sum_{i_1}\sum_{u_0}\sum_{i_0}\nabla_\theta\big(\ln(p(u_1|i_1,\theta^{(t)}))\big)p(u_1|i_1,\theta^{(t)})p_{i_0,i_1}(u_0)p(u_0|i_0,\theta^{(t)})p(i_0)C_0^0 \tag{7}$$

Now re-arranging the terms that that does not depend on (i_0,u_0):

$$\sum_{u_1}\sum_{i_1}\nabla_\theta\big(\ln(p(u_1|i_1,\theta^{(t)}))\big)p(u_1|i_1,\theta^{(t)})\sum_{u_0}\sum_{i_0}p_{i_0,i_1}(u_0)p(u_0|i_0,\theta^{(t)})p(i_0)C_0^0 \tag{8}$$

Now we write the following chain of equalities:

$$\begin{aligned}&\sum_{u_1}\sum_{i_1}\nabla_\theta\big(\ln(p(u_1|i_1,\theta^{(t)}))\big)p(u_1|i_1,\theta^{(t)})\sum_{u_0}\sum_{i_0}p_{i_0,i_1}(u_0)p(u_0|i_0,\theta^{(t)})p(i_0)C_0^0 =\\ &\sum_{u_1}\sum_{i_1}\nabla_\theta\big(p(u_1|i_1,\theta^{(t)})\big)\sum_{u_0}\sum_{i_0}p_{i_0,i_1}(u_0)p(u_0|i_0,\theta^{(t)})p(i_0)C_0^0 =\\ &\sum_{u_1}\sum_{i_1}\nabla_\theta\big(p(u_1|i_1,\theta^{(t)})\big)\sum_{u_0}\sum_{i_0}p(i_1,i_0,u_0|\theta^{(t)})C_0^0\end{aligned} \tag{9}$$

where in the first equality we use the "log-trick" $\nabla\ln(p)=\frac{\nabla p}{p}$. Now conditioning the terms inside the last two sums on i_1 we have that:

$$\sum_{u_1}\sum_{i_1}\nabla_\theta\big(p(u_1|i_1,\theta^{(t)})\big)p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big] \tag{10}$$

where we "packed" the expectation:

$$\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big] = \sum_{u_0}\sum_{i_0}p(i_0,u_0|i_1,\theta^{(t)})C_0^0 \tag{11}$$

Now we bring the gradient outside the sum, by applying the product rule in reverse. This lead us to:

$$\begin{aligned}&\sum_{u_1}\sum_{i_1}\nabla_\theta\big(p(u_1|i_1,\theta^{(t)})\big)p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big] =\\ &\nabla_\theta\Big(\sum_{u_1}\sum_{i_1}p(u_1|i_1,\theta^{(t)})p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\Big)-\\ &\sum_{u_1}\sum_{i_1}p(u_1|i_1,\theta^{(t)})\nabla_\theta\big(p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\big)\end{aligned} \tag{12}$$

Now we note that we can marginalize out u_1 (i.e.: we remove u_1 by summing over all u_1):

$$\nabla_\theta\big(\sum_{i_1} p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\big) - \sum_{i_1}\nabla_\theta\big(p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\big) =$$
$$\sum_{i_1}\nabla_\theta\big(p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\big) - \sum_{i_1}\nabla_\theta\big(p(i_1|\theta^{(t)})\mathbb{E}_{(i_0,u_0)}\big[C_0^0|i_1,\theta^{(t)}\big]\big) = 0 \tag{13}$$

Now we can repeat the same argument for every $0 \leq k \leq M-1$, which yields:

$$\mathbb{E}_{p(z|\theta)}\Big[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)C_0^{k-1}\Big] = 0 \tag{14}$$

Hence the policy gradient can be written as:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) = \mathbb{E}_{p(z|\theta)}\Big[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)C_k^M\Big] \tag{15}$$

where:

$$C_k^M = \sum_{j=k}^{M}\alpha^j g(i_j,u_j) + \alpha^M\hat{J}_M(i_m) \tag{16}$$

Now, by using Sample Average Approximation (SAA):

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx$$
$$\frac{1}{S}\sum_{s=1}^{S}\Big(\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\big(\sum_{j=k}^{M-1}\alpha^j g(i_j^s,u_j^s) + \alpha^M\hat{J}_M(i_m^s)\big)\Big) \tag{17}$$

Let's compare that to what it was before:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx$$
$$\frac{1}{S}\sum_{s=1}^{S}\Big(\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\big(\sum_{j=0}^{M-1}\alpha^j g(i_j^s,u_j^s) + \alpha^M\hat{J}_M(i_m^s)\big)\Big) \tag{18}$$

Both estimates are unbiased: in expectation they are equal to the true gradient of the REINFORCE optimization problem. However, the variance of the new estimate is smaller, since we are summing less terms (the costs are usually non-negative). This change, by itself as simple as it is, has a major impact in reducing the variance of the policy gradient.

In addition, in the new estimate, the accumulated costs C_k^M obey causality: they are accumulated from stage k to M. In other words, they are the cost-to-go from i_k with terminal cost approximation at the truncated stage M. Then we can equivalently write:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) = \mathbb{E}_{p(z|\theta^{(t)})}\Big[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)\Big] \tag{19}$$

where we now explicitly write $C_k^M = \bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)$, to emphasize that this is the "cost-to-go" starting from i_k associated with the policy $\tilde{\mu}(\theta^{(t)})$. Note that, however this is not the true cost-to-go $J_{\tilde{\mu}(\theta^{(t)})}(i_k)$ since we still discount every stage cost absolutely w.r.t. to stage zero.

1.1 Adding a baseline

We further improve the policy gradient by including baselines. The goal is to reduce the variance by introducing a baseline b as follows:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) = \mathbb{E}_{p(z|\theta^{(t)})}\Bigg[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\Big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)-b\Big)\Bigg] \tag{20}$$

And we notice that (we leave the full derivation as an exercise) that:

$$\mathbb{E}_{p(z|\theta^{(t)})}\Bigg[\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\Bigg] = \sum_{z\in Z}\sum_{k=0}^{M-1}\nabla_\theta\big(p(u_k|i_k,\theta^{(t)})\big) = \nabla_\theta\big(\sum_{z\in Z}p(z|\theta^{(t)})\big) = 0 \tag{21}$$

so by adding a baseline b, we are not introducing any bias into the policy gradient. Then the goal is to select b in order to minimize the variance. To that end, we write the variance as:

$$Var = \mathbb{E}_{p(z|\theta^{(t)})}\Bigg[\Big(\sum_{k=0}^{M-1}g(i_k,u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)-b\big)\Big)^2\Bigg] - \mathbb{E}_{p(z|\theta^{(t)})}\Bigg[\sum_{k=0}^{M-1}g(i_k,u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)-b\big)\Bigg]^2 \tag{22}$$

where we define $g_k(i_k,u_k|\theta^{(t)}) = \nabla_\theta\big(\ln(p(u_k|i_k,\theta^{(t)}))\big)$. The variance is, in fact, a convex function of the baseline b, then we can take the derivative w.r.t b and set it to zero. We note that the second term does not really depend on b, since:

$$\mathbb{E}_{p(z|\theta^{(t)})}\Bigg[\sum_{k=0}^{M-1}g(i_k,u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)-b\big)\Bigg]^2 = \mathbb{E}_{p(z|\theta^{(t)})}\Bigg[\sum_{k=0}^{M-1}g(i_k,u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)\big)\Bigg]^2 \tag{23}$$

since by adding b we do not introduce any bias, that is, we do not change the expectation. Hence we only need to differentiate the first term of the variance, which gives:

$$\frac{\partial Var}{\partial b} = -2\mathbb{E}_{p(z|\theta^{(t)})}\Bigg[\Big(\sum_{k=0}^{M-1}g(i_k,u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)\big)\Big)\Big(\sum_{k=0}^{M-1}g(i_k,u_k|\theta^{(t)})\Big)\Bigg] + 2b\mathbb{E}_{p(z|\theta^{(t)})}\Bigg[\Big(\sum_{k=0}^{M-1}g(i_k,u_k|\theta^{(t)})\Big)^2\Bigg] = 0 \tag{24}$$

Then we write the optimal baseline:

$$b^* = \frac{\mathbb{E}_{p(z|\theta^{(t)})}\Bigg[\Big(\sum_{k=0}^{M-1}g(i_k,u_k|\theta^{(t)})\big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k)\big)\Big)\Big(\sum_{k=0}^{M-1}g(i_k,u_k|\theta^{(t)})\Big)\Bigg]}{\mathbb{E}_{p(z|\theta^{(t)})}\Bigg[\Big(\sum_{k=0}^{M-1}g(i_k,u_k|\theta^{(t)})\Big)^2\Bigg]} \tag{25}$$

Now, using SAA, we write:

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

this is the "best" possible baseline for reducing. However, in practice, a simpler baseline is often used:

$$\bar{b} = \frac{1}{S}\sum_{s=1}^{S}\Big(\sum_{j=0}^{M-1}\alpha^j g(i_j^s,u_j^s) + \alpha^M \hat{J}_M(i_m^s)\Big) \tag{26}$$

This baseline has a nice intuition: it is the average cost across all sampled trajectories and it is easier to compute than the optimal baseline.

1.2 Introducing bias

Now we perform out last modification to the policy gradient. We will purposefully introduce bias in the estimate. The reason for that is very subtle. Let's write the policy gradient with baseline:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx$$
$$\frac{1}{S}\sum_{s=1}^{S}\Big(\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\big(\sum_{j=k}^{M-1}\alpha^j g(i_j^s,u_j^s) + \alpha^M\hat{J}_M(i_m^s) - b\big)\Big) \tag{27}$$

By manipulating the discount factors α's we can write an equivalent expression:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx$$
$$\frac{1}{S}\sum_{s=1}^{S}\Big(\sum_{k=0}^{M-1}\alpha^k\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\big(\sum_{j=k}^{M-1}\alpha^{j-k} g(i_j^s,u_j^s) + \alpha^{M-k}\hat{J}_M(i_m^s) - b\big)\Big) \tag{28}$$

Note that if M is large, and α small, then many terms of the above gradient will vanish (they will be very small). This is not ideal, since we use SAA to approximate the gradient and if many terms are close to zero, the gradient step will not make much progress. In order to solve this problem, the following change has been proposed:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx$$
$$\frac{1}{S}\sum_{s=1}^{S}\Big(\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\big(\sum_{j=k}^{M-1}\alpha^{j-k} g(i_j^s,u_j^s) + \alpha^{M-k}\hat{J}_M(i_m^s) - b\big)\Big) \tag{29}$$

and we note that the discount factor is only present in the cost term (where the costs are now discounted relatively with stage k, instead of absolutely with stage 0). This introduces bias on the estimate of the policy gradient. It is not trivial to see why this would be a good idea. In principle, introducing bias is bad,

because the Stochastic Gradient Descent converge properties rely on the gradient estimates to be unbiased. However, it turns out that this change does not bring any problems. In fact, it can be shown that this "biased" gradient is actually an unbiased gradient of another DP problem (one that is infinite-horizon average-cost problems). We have not covered such problems, only focusing on infinite-horizon discounted problem, so we will not delve into this matter further. We refer to the paper [**?**] for extra reading in introducing bias in the policy gradient. For our purposes it suffices to use the version of Eq(29) as the estimate of the policy gradient

2 Actor-Critic Algorithm

2.1 Policy Gradient as a function of the Advantage

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

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\Big(\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k^s)-b\Big)\Big] \tag{30}$$

It turns out we can also use a state-dependent baseline $b(i_k)$ and everything still hold:

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

We let such verification as an exercise (the proof follows very closely to the argument done in the previous state for a constant baseline b). The key idea to introduce the Critic is by focusing on the cost-to-go values $\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k^s)$ for every sampled trajectory. We state here again how these values are obtained, now in the biased version, with relative discounting:

$$\bar{J}_{\tilde{\mu}(\theta^{(t)})}(i_k^s) = J_{\tilde{\mu}(\theta^{(t)})}(i_k^s) = \sum_{j=k}^{M-1}\alpha^{j-k}g(i_j^s,u_j^s)+\alpha^{M-k}\hat{J}_M(i_m^s) \tag{31}$$

Note that this cost-to-go-values are now computed **as-if** stage k is actually stage 0: we perform relative discounting; accumulate costs for M time periods; use a terminal cost approximation at stage M. Hence this cost-to-go is in fact the true cost-to-go starting from i_k^s (hence the first equality).

Ideally we would like to compute $J_{\tilde{\mu}(\theta^{(t)})}(i_k^s)$ exactly, instead of relying on simulation. Let's suppose for a moment that this is possible. Then we can define the *true* Q-factor:

$$Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s) = g(i_k^s,u_k^s)+\mathbb{E}_{p(z|\theta^{(t)})}\Big[\sum_{j=k+1}^{\infty}\alpha^{j-k}g(i_j,u_j)\mid i_k^s,u_k^s\Big] \tag{32}$$

Which can then be used directly in the policy gradient:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\Big(Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s)-b(i_k^s)\Big)\Big] \tag{33}$$

However this is only the ideal case. We have to use SAA to compute such Q-factors. For a single sample trajectory for *every* pair (i_k^s, u_k^s) we return to Eq(31). We can re-write Eq(31) using the sampled Q-factors as:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx \\ \frac{1}{S}\sum_{s=1}^{S}\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\Big(\hat{Q}_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s) - b(i_k^s)\Big)\Big] \tag{34}$$

where:

$$\hat{Q}_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s) = \sum_{j=k}^{M-1}\alpha^{j-k}g(i_j^s,u_j^s) + \alpha^{M-k}\hat{J}_M(i_m^s) \tag{35}$$

Now we note that for each stage k, the horizon for every Q-factor is different! This fact is better illustrated by a figure:

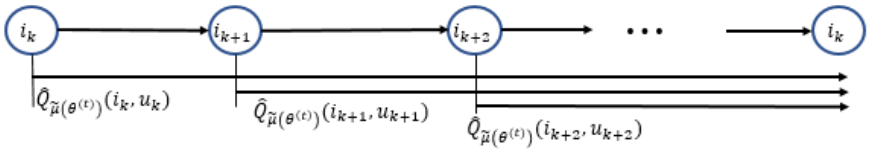

Figure 2: Schematic representation of the Q-factor computation: every Q-factor is computed starting from stage k until stage M

So we have to be careful: (1) the DP problem is infinite-horizon and we are using terminal cost approximations; (2) for each stage k, there is a Q-factor that "starts" at stage k and goes until stage M. Nevertheless, the goal is to estimate the Q-factors for every pair (i,u) given the policy $\tilde{\mu}(\theta^{(t)})$. And if correctly estimated, the Q-factors must solve the Bellman's Equation:

$$Q_{\tilde{\mu}(\theta^{(t)})}(i,u) = \sum_{j=1}^{n}p_{ij}(\tilde{\mu}(\theta^{(t)})(i))\big(g(i,u,j) + \alpha Q_{\tilde{\mu}(\theta^{(t)})}(j,\tilde{\mu}(\theta^{(t)})(j))\big) \tag{36}$$

Now consider the following state-dependent baseline:

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

This baseline is the *true* cost-to-go function starting from i_k^s. Then the policy gradient would become:

the quantity in parenthesis in called the **Advantage** of the pair (i_k^s, u_k^s). Again this is an ideal scenario, and we cannot compute the true cost-to-go and the true Q-factors. Nevertheless, we can re-write the above equation in terms of the advantage:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\Big(A_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s)\Big)\Big] \tag{37}$$

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

Figure 3: Policy gradient using the true cost-to-go function and true Q-factors

$$A_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s) = Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s) - J_{\tilde{\mu}(\theta^{(t)})}(i_k^s) \tag{38}$$

Note that the Advantage is a **relative** quantity: and we can, in fact, re-write

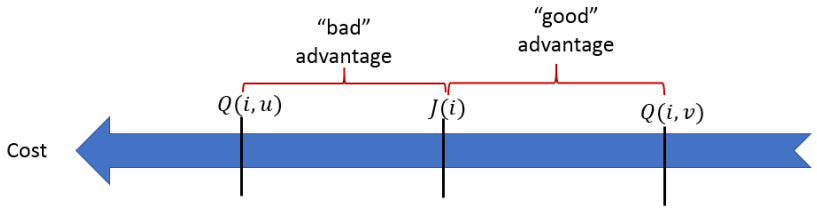

Figure 4: Schematic representation of the Advantage: it represents the relative merit between one control against another, given the current state

all algorithms we covered so far in terms of the Advantage. And more: We can write the Advantage of the pair (i_k^s, u_k^s) as follows:

$$A_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s) = g(i_k^s,u_k^s) + \alpha\mathbb{E}_{p(i_{k+1}|\theta^{(t)})}\Big[J_{\tilde{\mu}(\theta^{(t)})}(i_{k+1})\Big] - J_{\tilde{\mu}(\theta^{(t)})}(i_k^s) \tag{39}$$

where

$$Q_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s) = g(i_k^s,u_k^s) + \alpha\mathbb{E}_{p(i_{k+1}|\theta^{(t)})}\Big[J_{\tilde{\mu}(\theta^{(t)})}(i_{k+1})\Big] \tag{40}$$

The Critic comes in play precisely to approximate the Advantage.

2.2 Combining the Critic and the Actor

Now we are at last ready to bring the Critic into picture. Our goal is to approximate the Advantage:

$$A_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s) = g(i_k^s,u_k^s) + \alpha\mathbb{E}_{p(i_{k+1}|\theta^{(t)})}\Big[J_{\tilde{\mu}(\theta^{(t)})}(i_{k+1})\Big] - J_{\tilde{\mu}(\theta^{(t)})}(i_k^s) \tag{41}$$

to use it in the policy gradient:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}[F(z)]\big) \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\Big(A_{\tilde{\mu}(\theta^{(t)})}(i_k^s,u_k^s)\Big)\Big] \tag{42}$$

To that end, we will use another approximation architecture, say a DNN with parameter ϕ, which is obtain as the solution of a Regression problem:

$$\phi^{(t)} = \arg\min_{\phi}\left\{\sum_{l=1}^{L}\big(\tilde{J}(i_0^l,\phi) - \beta^l\big)^2\right\} \tag{43}$$

where the "labels" are computed as the cost-to-go starting from every state i_k^s, for all s and k, in the fashion outline in [figure]. So, in general, L (the total number of samples used in the critic-step) may be potentially much larger than S (the number of simulated trajectories):

$$\beta^l = \sum_{k=0}^{M-1} \alpha^k g(i_k^s, \mu^{(t)}(i_k^s), i_{k+1}^s) + \alpha^M \tilde{J}(i_M^s, \phi^{(t-1)}) \tag{44}$$

where the above label is computed starting from state i_0^s as an example. We note that the terminal cost approximation $\tilde{J}(\cdot)$ used here is the approximate cost-to-go using the previous iteration configuration $\phi^{(t-1)}$.

So, essentially, we are **reusing** simulated trajectories to obtain the labels β^l's. This is actually a problem as it limits exploration. We will push this back for now to focus on the optimization of Eq(43).

Note that solving Eq(43) to (local) optimality can be very costly. Then we can, instead perform a single gradient step:

$$\phi^{(t)} = \phi^{t-1} - \gamma^{(t)} \sum_{l=1}^{L} \nabla_\phi \tilde{J}(i_0^l, \phi)\big(\tilde{J}(i_0^l, \phi) - \beta^l\big) \tag{45}$$

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

$$\phi_{m+1}^{(t-1)} = \phi_m^{(t-1)} - \gamma_p^{(t-1)} \sum_{l=1}^{L} \nabla_\phi \tilde{J}(i_0^l, \phi)\big(\tilde{J}(i_0^l, \phi) - \beta_p^l\big), \quad \forall m \in \{0, ..., P\} \tag{46}$$

$$\beta_p^l = \sum_{k=0}^{M-1} \alpha^k g(i_k^s, \mu^{(t)}(i_k^s), i_{k+1}^s) + \alpha^M \tilde{J}(i_M^s, \phi_p^{(t-1)}), \quad \forall p \in \{0, ..., P\} \tag{47}$$

$$\phi^{(t)} \leftarrow \phi_P^{(t-1)} \tag{48}$$

These labels's may have high variance (recall the critic algorithm is also affected by the bias-variance trade-off). Handling the bias-variance trade-off is harder: we are already computing labels using reusing sampled trajectories. In practice, two variations are often used:

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

- Low bias
- High variance

Monte-Carlo Estimates

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

- High bias
- Low variance

Bootstrap Estimates

Two ways of computing the labels for the critic are as follows: (1) the Monte-Carlo estimates uses the rollout trajectories for a number of stages and uses terminal cost approximation; (2) the Bootstrap estimates immediately uses the terminal cost approximation after one stage.If often boils down to experimentation with the application at hand, in order to decide which variation to use in order to compute the labels.

After solving the regression problem, either until (local) optimality, or by using the optimistic approach, we can write the policy gradient as:

$$\nabla_\theta\big(\mathbb{E}_{p(z|\theta^{(t)})}\big[F(z)\big]\big) \approx \frac{1}{S}\sum_{s=1}^{S}\sum_{k=0}^{M-1}\nabla_\theta\big(\ln(p(u_k^s|i_k^s,\theta^{(t)}))\Big(\tilde{A}(i_k^s,u_k^s)\Big)\Big] \tag{49}$$

where the advantage approximation can be estimated via two variations as well:

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

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

Two two variants share the same interpretation: (1) the Monte-Carlo estimates uses the rollout trajectories for a number of stages and uses the previous configuration $\phi^{(t-1)}$ as the terminal cost approximation and as the baseline estimate; (2) the Bootstrap estimates immediately uses the previous configuration $\phi^{(t-1)}$ as the terminal cost approximation after one stage and as the baseline estimate. The bootstrap estimate is particularly nice because it estimates the advantage as a **temporal difference**: by taking the control u_k^s at stage i_k^s we compute the difference of it's cost plus the (discounted) future cost and the present cost-to-go. This interpretation of temporal differences (TD) can be carried over to many other approximation techniques (what is called as the "TD(λ) methods", which we will cover further ahead). Then, the Actor-Critic Algorithm then iterates:

1. Improves the critic $\phi^{(t+1)} \leftarrow \phi^{(t)}$ via regression
2. Improves the actor $\theta^{(t+1)} \leftarrow \theta^{(t)}$

2.3 The Algorithm

We conclude our lecture by stating the Actor-Critic Algorithm:

Algorithm 2 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:

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

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

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

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 $\tilde{J}(\cdot, \phi^{(T)})$

This Algorithm can be further improved by series of changes and variations: (1) using parallelization, with GPU's; (2) using asynchronous updates; (3) off-policy sampling; (4) improvement by enhancing exploration; and more. But overall, all these variations share the foundation presented here, which is an Algorithm that uses approximation architectures to approximate both the policy evaluation and policy improvement steps of the Policy Iteration Algorithm.