## Steps to understanding Policy-gradient methods

- Policy approximation  $\pi(a|s, \theta)$
- The average-reward (reward rate) objective  $\bar{r}(\boldsymbol{\theta})$

 $\Delta\boldsymbol{\theta}_t \approx \alpha$ 

 $\partial \bar{r}({\boldsymbol{\theta}})$ 

 $\partial\bm\theta$ 

- Stochastic gradient ascent/descent
- The policy-gradient theorem and its proof
- Approximating the gradient
- Eligibility functions for a few cases
- A final algorithm

# Policy Approximation

- Policy = a function from state to action
	- How does the agent select actions?
	- In such a way that it can be affected by learning?
	- In such a way as to assure exploration?
- Approximation: there are too many states and/or actions to represent all policies
	- To handle large/continuous action spaces

### What is learned and stored?

- 1. *Action-value methods*: learn the value of each action; pick the max (usually)
- 2. *Policy-gradient methods*: learn the parameters **u** of a stochastic *policy*, update by  $\nabla$ <sub>u</sub>Performance
	- including *actor-critic methods*, which learn *both* value and policy parameters
- *3. Dynamic Policy Programming*
- 4. *Drift-diffusion models* (Psychology)

### Actor-critic architecture



### Action-value methods *a s*0*,r* x(*s*<sup>0</sup> *, r|s, a*) athor i

• The *value of an action in a state given a policy* is the expected future reward starting from the state taking that first action, then following the policy thereafter the expected *l* future rev  $\overline{\phantom{a}}$ ard starting from

$$
q_{\pi}(s, a) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t \middle| S_0 = s, A_0 = a\right]
$$

• Policy: pick the max most of the time but sometimes pick at random ( $\varepsilon$ -greedy)  $A_t = \argmax Q$ *a*  $\hat{O}$  $t^{}(S_t, a)$  $A_t = \arg \max_a Q_t(S_t, a)$ 

### Why approximate policies rather than values?

- In many problems, the policy is simpler to approximate than the value function
- In many problems, the optimal policy is stochastic
	- e.g., bluffing, POMDPs
- To enable smoother change in policies
- To avoid a search on every step (the max)
- To better relate to biology

### Gradient-bandit algorithm  $\blacksquare$ use those estimates to select actions. This is not the good approach, but it is not only one possible. In this section we consider learning a numerical *preference Ht*(*a*) for each action *a*. The larger the preference, the more often that action is taken, but Cradiant bandit algorithm diadiene dandie aigunanni where here we have chosen *<sup>X</sup><sup>t</sup>* <sup>=</sup> *<sup>R</sup>*¯*<sup>t</sup>* and substituted *<sup>R</sup><sup>t</sup>* for *<sup>q</sup>*⇤(*At*), which is permitted **Gradient-bandit algorithm** 1*a*=*<sup>b</sup>* ⇡*t*(*a*) adient-bandit algorithm  $\overline{L}$  $\overline{f}$ P*<sup>k</sup> .* = ⇡*t*(*a*)*,* (2.9)

- Store action preferences  $H_t(a)$ <br>**as the relation of the preference**  $\Omega(a)$ rather than action-value estimates  $\; Q_t(a) \;$ tore action preferent<br>ather than action  $n-1$ P*<sup>k</sup>*  $n$  cess  $\pi_t(a)$ <br>alue estimates  $Q_t(a)$  $t(a)$  $\mathbf{v}$  action preferences  $\mathbf{u}_t(a)$  for the probability of  $\mathbf{v}_t(a)$  for the probability of the probability of  $\mathbf{v}_t(a)$  $\mathbf{r}$  at time action value coefficies  $\mathcal{L}(a)$
- Instead of  $\varepsilon$ -greedy, pick actions by an exponential soft-max:  $n \cdot 1$  $\alpha$ distribution (i.e., Gibbs or Boltzmann distribution) as  $\alpha$  $\alpha$  *e e f e f e h e f e f e f e f e f e f e f e f e f e f e f e f e f e f e f e f e f e f e f e f e f e f e f e* ead of  $\varepsilon$ ereed<br>Ereed dy, pick actio ⇡*t*(*At*)  $\overline{a}$  $ext{a}$  exponential sof  $\mathbf{r}$ */*⇡*t*(*At*) tead of  $\varepsilon$ -greedy, pick actions by an exponential soft-max:

$$
\Pr\{A_t = a\} \doteq \frac{e^{H_t(a)}}{\sum_{b=1}^k e^{H_t(b)}} \doteq \pi_t(a)
$$

- $\bullet$  Also store the sample average of rewards as  $\bar{R}_t$ taking action *a* at time *t*. Initially all preferences are the same (e.g., *H*1(*a*)=0*,* 8*a*)  $\bullet$  Also store the sample average of rewards as  $R_t$  $\frac{1}{\sqrt{2}}$ *h* the sample average of rewa  $T_{\text{eff}}$  is  $\frac{1}{2}$  and  $\frac{1}{2}$  and  $\frac{1}{2}$  and  $\frac{1}{2}$  and square-square-square-square-square-square-square-square-square-square-square-square-square-square-square-square-square-square-square-square-square-squar  $r$ ds as  $\bar{R}_t$
- Then update: so that all actions have all actions have an equal probability of being selected. In the selection of being se **e** Then update: *<sup>H</sup>t*+1(*a*) *.*

$$
H_{t+1}(a) = H_t(a) + \alpha (R_t - \bar{R}_t) \left( \mathbf{1}_{a=A_t} - \pi_t(a) \right)
$$
  
For 0, depending on whether

*R*<sup>t</sup> *R*<sup>*t*</sup> *R*<sub>*t*</sub> *R*<sub>*t*</sub> *R*<sub></sub>  $\frac{1}{\sqrt{2}}$   $\frac{1}{\sqrt{2}}$   $\frac{1}{\sqrt{2}}$   $\frac{1}{\sqrt{2}}$ = *H*<sup>t</sup>(*a*) *H*<sub>t</sub>(*a*) *H*<sub>t</sub>(*a*) *H*<sub>t</sub>(*a*) *H*<sub>t</sub>(  $P$ *redicate (subsc*  $f(r(t))$  is true appears in the denominator in the uncertainty of the uncertainty term is the uncertainty term in the term is decreased. The term is determined to the uncertainty of the uncertainty of the term is determined. The second ter other hand, each time an action other than *a* is selected *t* is increased; as it appears in the predicate (subscript) is true

### Gradient-bandit algorithms on the 10-armed testbed



Figure 2.6: Average performance of the gradient-bandit algorithm with and without a reward baseline on the 10-armed testbed when the  $q_*(a)$  are chosen to be near  $+4$  rather than near zero.

$$
\frac{\partial}{\partial x}\left[\frac{f(x)}{g(x)}\right] = \frac{\frac{\partial f(x)}{\partial x}g(x) - f(x)\frac{\partial g(x)}{\partial x}}{g(x)^2}
$$
\n
$$
\frac{\partial}{\partial H_t(a)} = \frac{\partial}{\partial H_t(a)}\pi_t(b)
$$
\n
$$
= \frac{\frac{\partial}{\partial H_t(a)}}{\frac{\partial H_t(a)}{\partial H_t(a)}} \left[\frac{e^{H_t(b)}}{\sum_{c=1}^k e^{H_t(c)}}\right]
$$
\n
$$
= \frac{\frac{\partial e^{H_t(b)}}{\partial H_t(a)} \sum_{c=1}^k e^{H_t(c)} - e^{H_t(b)} \frac{\partial \sum_{c=1}^k e^{H_t(c)}}{\partial H_t(a)}}{\left(\sum_{c=1}^k e^{H_t(c)}\right)^2}
$$
 (by the quotient rule)\n
$$
= \frac{1_{a=b}e^{H_t(a)} \sum_{c=1}^k e^{H_t(c)}}{\left(\sum_{c=1}^k e^{H_t(c)}\right)^2}
$$
 (because  $\frac{\partial e^x}{\partial x} = e^x$ )\n
$$
= \frac{1_{a=b}e^{H_t(b)}}{\sum_{c=1}^k e^{H_t(c)}} - \frac{e^{H_t(b)} e^{H_t(a)}}{\left(\sum_{c=1}^k e^{H_t(c)}\right)^2}
$$
\n
$$
= 1_{a=b} \pi_t(b) - \pi_t(b) \pi_t(a)
$$
\n
$$
= \pi_t(b) (1_{a=b} - \pi_t(a)).
$$
 Q.E.D.

## Steps to understanding Policy-gradient methods

- Policy approximation  $\pi(a|s, \theta)$
- The average-reward (reward rate) objective  $\bar{r}(\boldsymbol{\theta})$

 $\Delta\boldsymbol{\theta}_t \approx \alpha$ 

 $\partial \bar{r}({\boldsymbol{\theta}})$ 

 $\partial\bm\theta$ 

- Stochastic gradient ascent/descent
- The policy-gradient theorem and its proof
- Approximating the gradient
- Eligibility functions for a few cases
- A complete algorithm

### eg, linear-exponential policies (discrete actions) *<sup>R</sup> <sup>R</sup>*¯ + ˆ*v*(*S*<sup>0</sup> *,* w) *v* ˆ(*S,* w) *R*<br>Rod – C  $G$  and *s* w w + 2 aw ew + 2 aw CHOOSE AVRAMA Take action *A*, observe *S*<sup>0</sup> *, R <u><i>R <i>R*</del>  $\overline{AB}$  *x*</u> *<sup>R</sup>*¯ *<sup>R</sup>*¯ <sup>+</sup> ↵✓

• The "preference" for action *a* in state *s* is linear in  $\boldsymbol{\theta}$  and a state-action feature vector  $\boldsymbol{\phi}(s,a)$ e "preference" for a  $\theta$  and a state-ad <sup>e</sup>✓ e✓ <sup>+</sup> <sup>r</sup>⇡(*A|S,*✓)

*b*

• The probability of action *a* in state *s* is exponential in its preference e probability

$$
\pi(a|s, \boldsymbol{\theta}) \doteq \frac{\exp(\boldsymbol{\theta}^{\top}\boldsymbol{\phi}(s, a))}{\sum_{b} \exp(\boldsymbol{\theta}^{\top}\boldsymbol{\phi}(s, b))}
$$

• Corresponding eligibility function: × *∠<sub>b</sub>* ∠<sub>*a*</sub> *γ*(*a*<sub>γ</sub>),<br>esponding eligibility functi  $\nabla \pi(a|s, \boldsymbol{\theta})$  $\pi(a|s,\boldsymbol{\theta})$  $= \boldsymbol{\phi}(s, a) - \sum$  $\pi(b|s,\boldsymbol{\theta})\boldsymbol{\phi}(s,b)$ 

### Policy-gradient setup

parameterized  
\npolicies\n
$$
\pi(a|s, \theta) \doteq \Pr\{A_t = a \mid S_t = s\}
$$
\naverage-reward  
\nobjective\n
$$
r(\pi) \doteq \lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^n \mathbb{E}_{\pi}[R_t] = \sum_s d_{\pi}(s) \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)r
$$
\nsteady-state  
\ndistribution\n
$$
d_{\pi} \doteq \lim_{t \to \infty} \Pr\{S_t = s\}
$$
\ndifferential\n
$$
\hat{v}_{\pi}(s) \doteq \sum_{k=1}^{\infty} \mathbb{E}_{\pi}[R_{t+k} - r(\pi) \mid S_t = s]
$$
\ndifferential\n
$$
\tilde{q}_{\pi}(s, a) \doteq \sum_{k=1}^{\infty} \mathbb{E}_{\pi}[R_{t+k} - r(\pi) \mid S_t = s, A_t = a]
$$
\ndifferential\naction-value\n
$$
\text{stochastic} \Delta \theta_t \approx \alpha \frac{\partial r(\pi)}{\partial \theta} \doteq \alpha \nabla r(\pi)
$$

$$
\begin{aligned}\n\text{stochastic} \quad &\Delta \boldsymbol{\theta}_t \approx \alpha \frac{\partial r(\pi)}{\partial \boldsymbol{\theta}} \doteq \alpha \nabla r(\pi) \\
\text{policy-gradient} \quad &\nabla r(\pi) = \sum_s d_\pi(s) \sum_a \tilde{q}_\pi(s, a) \nabla \pi(a|s, \boldsymbol{\theta}) \\
&= \mathbb{E} \bigg[ \left( \tilde{q}_\pi(S_t, A_t) - v(S_t) \right) \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t)} \bigg| S_t \sim d_\pi, A_t \sim \pi(\cdot|S_t, \boldsymbol{\theta}) \bigg] \\
&= \mathbb{E} \bigg[ \left( \tilde{G}_t^\lambda - \hat{v}(S_t, \mathbf{w}) \right) \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t)} \bigg| S_t \sim d_\pi, A_{t:\infty} \sim \pi \bigg] \\
&\approx \left( \tilde{G}_t^\lambda - \hat{v}(S_t, \mathbf{w}) \right) \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t)} \qquad \text{(by sampling under } \pi)\n\end{aligned}
$$

$$
\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \left(\tilde{G}_t^{\lambda} - \hat{v}(S_t, \mathbf{w})\right) \frac{\nabla \pi(A_t | S_t, \boldsymbol{\theta})}{\pi(A_t | S_t)}
$$

e.g., in the one-step linear case:

$$
= \boldsymbol{\theta}_t + \alpha \Big( R_{t+1} - \bar{R}_t + \mathbf{w}_t^\top \boldsymbol{\phi}_{t+1} - \mathbf{w}_t^\top \boldsymbol{\phi}_t) \Big) \frac{\nabla \pi (A_t | S_t, \boldsymbol{\theta})}{\pi (A_t | S_t)}
$$

Deriving the policy-gradient theorem:  $\nabla r(\pi) = \sum_s d_{\pi}(s) \sum_a \tilde{q}_{\pi}(s, a) \nabla \pi(a|s, \theta)$ :

$$
\nabla \tilde{v}_{\pi}(s) = \nabla \sum_{a} \pi(a|s, \theta) \tilde{q}_{\pi}(s, a)
$$
  
\n
$$
= \sum_{a} \left[ \nabla \pi(a|s, \theta) \tilde{q}_{\pi}(s, a) + \pi(a|s, \theta) \nabla \tilde{q}_{\pi}(s, a) \right]
$$
  
\n
$$
= \sum_{a} \left[ \nabla \pi(a|s, \theta) \tilde{q}_{\pi}(s, a) + \pi(a|s, \theta) \nabla \sum_{s', r} p(s', r|s, a) \left[ r - r(\pi) + \tilde{v}_{\pi}(s') \right] \right]
$$
  
\n
$$
= \sum_{a} \left[ \nabla \pi(a|s, \theta) \tilde{q}_{\pi}(s, a) + \pi(a|s, \theta) \left[ -\nabla r(\pi) + \sum_{s', r} p(s'|s, a) \nabla \tilde{v}_{\pi}(s') \right] \right]
$$

$$
\therefore \nabla r(\pi) = \sum_{a} \Big[ \nabla \pi(a|s, \boldsymbol{\theta}) \tilde{q}_{\pi}(s, a) + \pi(a|s, \boldsymbol{\theta}) \sum_{s'} p(s'|s, a) \nabla \tilde{v}_{\pi}(s') \Big] - \nabla \tilde{v}_{\pi}(s)
$$

$$
\therefore \nabla r(\pi) = \sum_{a} \left[ \nabla \pi(a|s, \boldsymbol{\theta}) \tilde{q}_{\pi}(s, a) + \pi(a|s, \boldsymbol{\theta}) \sum_{s'} p(s'|s, a) \nabla \tilde{v}_{\pi}(s') \right] - \nabla \tilde{v}_{\pi}(s)
$$

$$
\begin{split}\n\therefore \sum_{s} d_{\pi}(s) \nabla r(\pi) &= \sum_{s} d_{\pi}(s) \sum_{a} \nabla \pi(a|s, \theta) \tilde{q}_{\pi}(s, a) \\
&\quad + \sum_{s} d_{\pi}(s) \sum_{a} \pi(a|s, \theta) \sum_{s'} p(s'|s, a) \nabla \tilde{v}_{\pi}(s') - \sum_{s} d_{\pi}(s) \nabla \tilde{v}_{\pi}(s) \\
&= \sum_{s} d_{\pi}(s) \sum_{a} \nabla \pi(a|s, \theta) \tilde{q}_{\pi}(s, a) \\
&\quad + \sum_{s'} \sum_{s} d_{\pi}(s) \sum_{a} \pi(a|s, \theta) p(s'|s, a) \nabla \tilde{v}_{\pi}(s') - \sum_{s} d_{\pi}(s) \nabla \tilde{v}_{\pi}(s)\n\end{split}
$$

$$
\nabla r(\pi) = \sum_{s} d_{\pi}(s) \sum_{a} \nabla \pi(a|s, \boldsymbol{\theta}) \tilde{q}_{\pi}(s, a)
$$

### Complete PG algorithm Complete ru algorithm

Initialize parameters of policy  $\boldsymbol{\theta} \in \mathbb{R}^n$ , and state-value function  $\mathbf{w} \in \mathbb{R}^m$ Initialize eligibility traces  $e^{\theta} \in \mathbb{R}^n$  and  $e^w \in \mathbb{R}^m$  to **0** Initialize  $R = 0$ 

On each step, in state *S*:

Choose *A* according to  $\pi(\cdot|S, \theta)$ Take action *A*, observe  $S'$ , *R*  $\delta \leftarrow R - \bar{R} + \hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w})$  $\bar{R} \leftarrow \bar{R} + \alpha^{\theta} \delta$  $\mathbf{e}^{\mathbf{w}} \leftarrow \lambda \mathbf{e}^{\mathbf{w}} + \nabla_{\mathbf{w}} \hat{v}(S, \mathbf{w})$  $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}} \delta \mathbf{e}^{\mathbf{w}}$  $\mathbf{e}^{\boldsymbol{\theta}} \leftarrow \lambda \mathbf{e}^{\boldsymbol{\theta}} + \frac{\nabla \pi(A|S,\boldsymbol{\theta})}{\pi(A|S,\boldsymbol{\theta})}$  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} \delta \mathbf{e}^{\boldsymbol{\theta}}$ 

update eligibility trace for critic form TD error from critic update average reward estimate update critic parameters update eligibility trace for actor update actor parameters

### **The generality of the** policy-gradient strategy ⌘r⇡(*At|St,* ✓) ⇡(*At|St,* ✓)  $\overline{a}$  $\int f f(x) dx$  $\overline{\phantom{a}}$ ⌘r⇡(*At|St,* ✓) ⇡(*At|St,* ✓)  $\overline{\phantom{a}}$  $\overline{\phantom{a}}$ *S<sup>t</sup>* ⇠ *d*⇡*, At*:<sup>1</sup> ⇠ ⇡ ⌘r⇡(*At|St,* ✓)

- Can be applied whenever we can compute the effect of parameter changes on the action probabilities, • Can be applied whenever we can compute the pr *<sup>t</sup> v* ˆ(*St,* w)  $\mathbf{\nabla} \nabla \pi(A_t|S_t, \boldsymbol{\theta})$
- E.g., has been applied to spiking neuron models e.g., in the one-step linear case:
	- There are many possibilities other than linearricre are many possibilities of exponential and linear-gaussian<br>  $\cdot$
	- e.g., mixture of random, argmax, and fixedwidth gaussian; learn the mixing weights, drift/ diffusion models

### eg, linear-gaussian policies (continuous actions)



### eg, linear-gaussian policies (continuous actions) ✓ ✓ + ↵✓ e✓ <sup>e</sup>✓ e✓ <sup>+</sup> <sup>r</sup>⇡(*A|S,*✓) ⇡(*A|S,*✓) exp(✓>(*s, a*)) ⇡(*a|s,* ✓) *.* (*s*) *.* = exp(✓<sup>&</sup>gt; (*s*) z, linear  $\mathbf{z}$ , linear

• The mean and std. dev. for the action taken in state *s* are linear and linear-exponential in d. dev. for the action taken i a mean and std. dev. for the mean and std. dev. for the sense of t = ear and linear-ex ⇡(*a|s,* ✓) *.*  $\overline{a}$ and std. lev for the action e actio

$$
\boldsymbol{\theta} \doteq (\boldsymbol{\theta}_{\mu}^{\top}; \boldsymbol{\theta}_{\sigma}^{\top})^{\top} \qquad \mu(s) \doteq \boldsymbol{\theta}_{\mu}^{\top} \boldsymbol{\phi}(s) \qquad \sigma(s) \doteq \exp(\boldsymbol{\theta}_{\sigma}^{\top} \boldsymbol{\phi}(s))
$$

• The probability density function for the action taken in state *s* is gaussian ⇡(*a|s,* ✓) *.*  $\begin{array}{c}\n\lambda & \Omega\n\end{array}$ tio (*s*) *<sup>µ</sup>*(*s*) *.* = ✓<sup>&</sup>gt; *<sup>µ</sup>* (*s*) **(en in state s is gate)** 

$$
\pi(a|s,\boldsymbol{\theta}) \doteq \frac{1}{\sigma(s)\sqrt{2\pi}} \exp\left(-\frac{(a-\mu(s))^2}{2\sigma(s)^2}\right)
$$

# Gaussian eligibility functions

$$
\frac{\nabla_{\boldsymbol{\theta}_{\mu}}\pi(a|s,\boldsymbol{\theta})}{\pi(a|s,\boldsymbol{\theta})} = \frac{1}{\sigma(s)^2}(a-\mu(s))\boldsymbol{\phi}_{\mu}(s)
$$

$$
\frac{\nabla_{\boldsymbol{\theta}_{\sigma}}\pi(a|s,\boldsymbol{\theta})}{\pi(a|s,\boldsymbol{\theta})} = \left(\frac{(a-\mu(s))^2}{\sigma(s)^2} - 1\right)\boldsymbol{\phi}_{\sigma}(s)
$$

### The generality of the policy-gradient strategy (2) = E *q*˜⇡(*St, At*) *v*(*St*)  $\mathbf{f} \times \mathbf{f}$ Ì I *S<sup>t</sup>* ⇠ *d*⇡*, A<sup>t</sup>* ⇠ ⇡(*·|St,* ✓) = E *G*˜  $\sim$   $\delta$  and  $\sim$ *<sup>t</sup> v* ˆ(*St,* w) ⇡(*At|St,* ✓) l, i, *S<sup>t</sup>* ⇠ *d*⇡*, At*:<sup>1</sup> ⇠ ⇡ ⇣ *G*˜ ⌘r⇡(*At|St,* ✓)

- Can be applied whenever we can compute the effect of parameter changes on the action probabilities, **Examplied** whenever we can compute the effect of ✓*t*+1 *.* = ✓*<sup>t</sup>* + ↵ ⇣ *G*˜ *r*  $\overline{\phantom{a}}$  $\bigtriangledown$ ,  $\nabla \pi(A_t|S_t, \boldsymbol{\theta})$
- Can we apply PG when outcomes are viewed as action? e.g., in the one-step linear case: in the one-step linear case: in the one-step linear case: in the one-step linear<br>The one-step linear case: in the one-step linear case: in the one-step linear case: in the one-step linear
	- e.g., the action of "turning on the light" or the action of "going to the bank" r the act are the bank"<br>and if "going to the bank"  $\alpha$  the action of going to the bank<br>*A*<sup>t</sup>
	- is this an alternate strategy for temporal abstraction?  $t$  this an alternate s
	- We would need to learn—not compute—the gradient of these states w.r.t. the policy parameter

### Have we eliminated action?

- If any state can be an action, then what is still special about actions?
- The parameters/weights are what we can really, directly control
- We have always, in effect, "sensed" our actions (even in the  $\varepsilon$ -greedy case)
- Perhaps actions are just sensory signals that we can usually control easily
- Perhaps there is no longer any need for a special concept of action in the RL framework