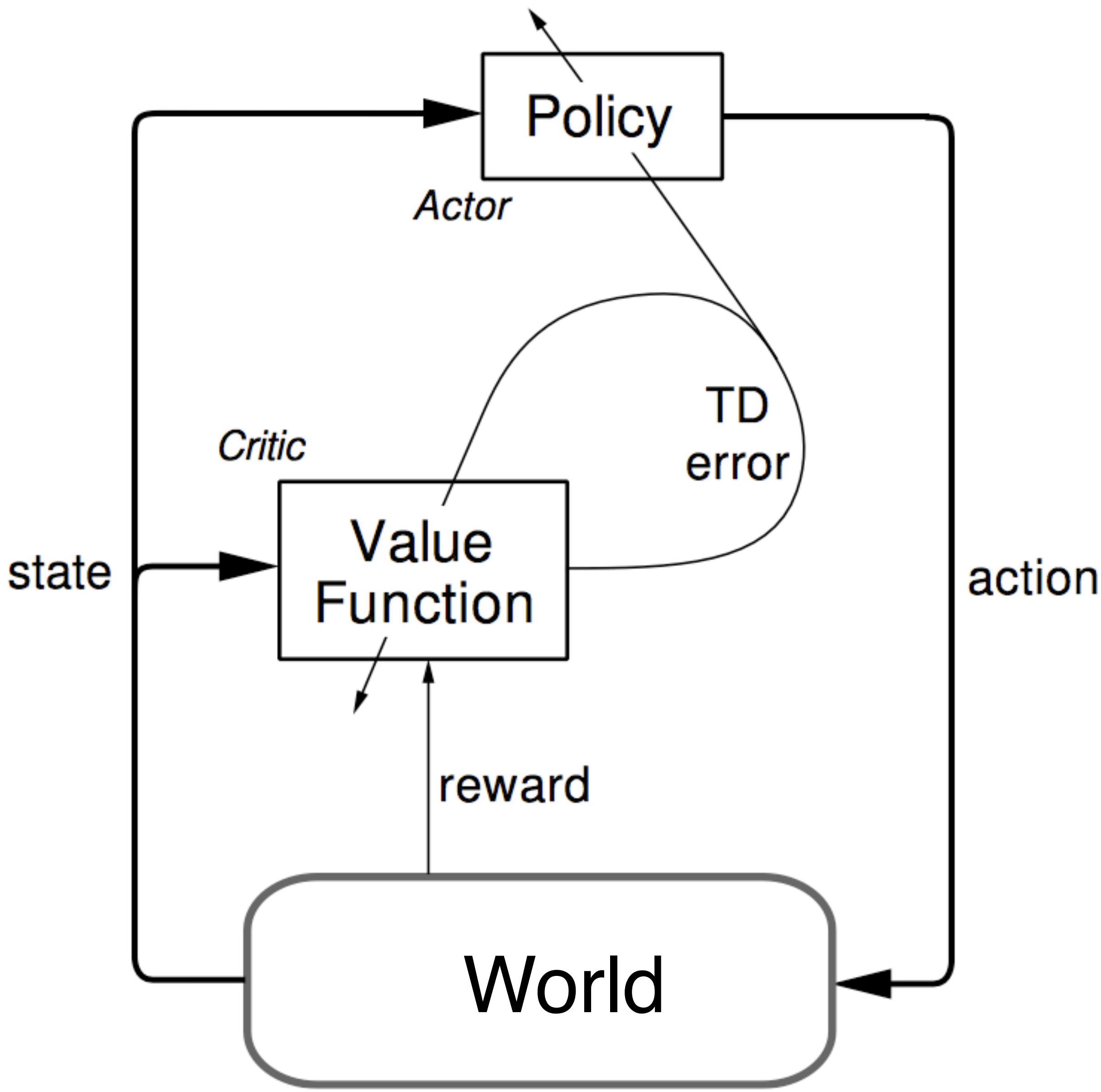
Policy Approximation

- 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 $\mathbf{u}$ of a stochastic *policy*, update by $\nabla_{\mathbf{u}}\text{Performance}$
   - 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

- 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

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

- Policy: pick the max most of the time

$$A_t = \arg\max_a \hat{Q}_t(S_t, a)$$

but sometimes pick at random (ε-greedy)

We should never discount when approximating policies!

γ is ok it there is a start state/distribution

Average reward setting

- All rewards are compared to the average reward

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

- where

$$\bar{r}(\pi) = \lim_{t\to\infty} \frac{1}{t}\mathbb{E}\left[R_1 + R_2 + \cdots + R_t \mid A_{0:t-1} \sim \pi\right]$$

- and we learn an approximation

$$\bar{r}_t \approx \bar{r}(\pi_t)$$

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

Policy-gradient methods

- The policy itself is learned and stored
  - the policy is parameterized by $\mathbf{u} \in \mathbb{R}^n$
  - we learn and store $\mathbf{u}$

$$\mathbb{P}r\left[A_t = a\right] = \pi_{\mathbf{u}_t}(a|S_t)$$

- $\mathbf{u}$ is updated by approximate gradient ascent

$$\mathbf{u}_{t+1} = \mathbf{u}_t + \alpha \widehat{\nabla_{\mathbf{u}} \bar{r}(\pi_{\mathbf{u}})}$$

eg, linear-exponential policies (discrete actions)

- The "preference" for action a in state s is linear in $\mathbf{u}$

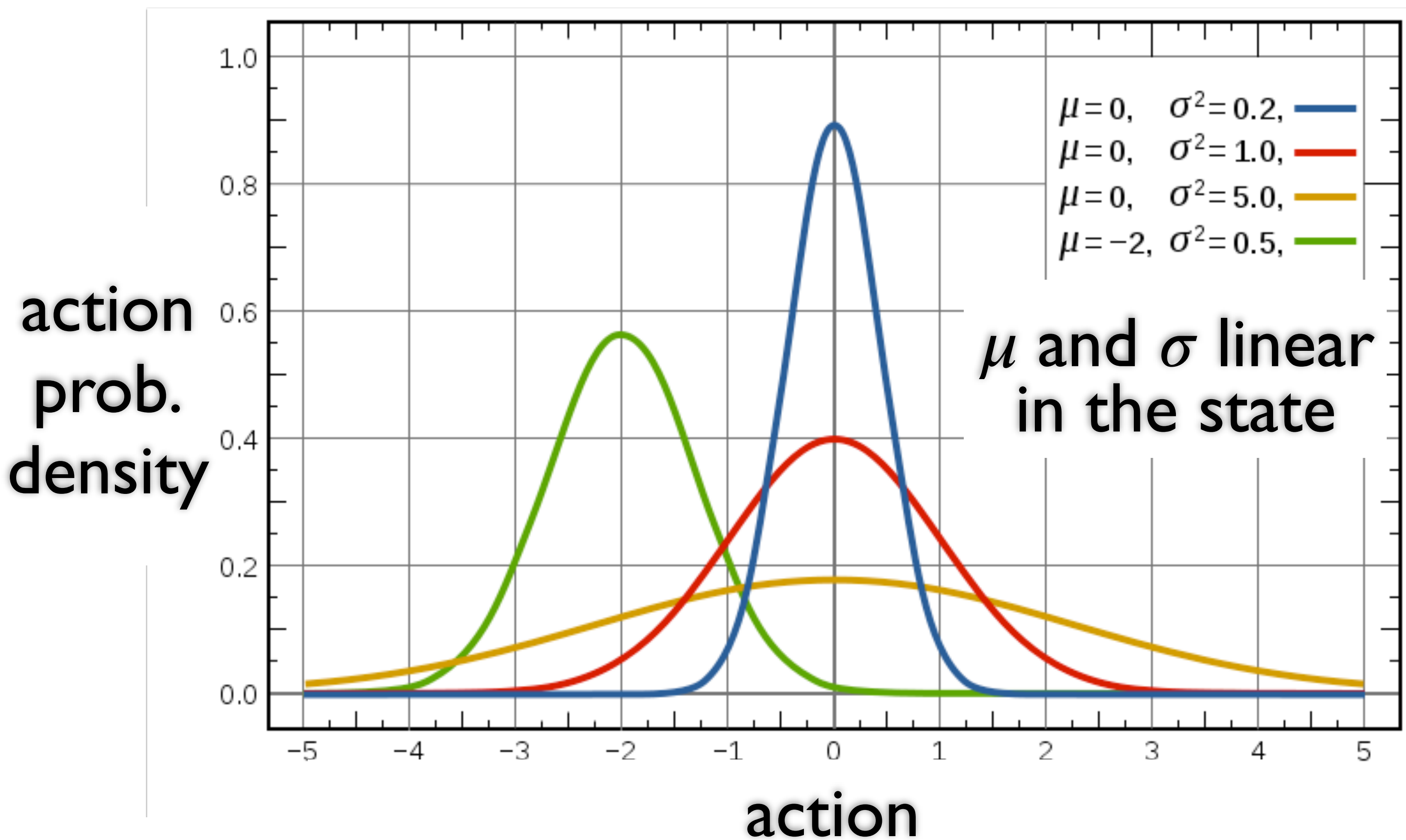
$$\mathbf{u}^\top \mathbf{x}_{sa} \equiv \sum_i u(i)\mathbf{x}_{sa}(i)$$

feature vector $\in \mathbb{R}^n$

- The probability of action a in state s is exponential in its preference

$$\pi_{\mathbf{u}}(a|s) = \frac{e^{\mathbf{u}^\top \mathbf{x}_{sa}}}{\sum_b e^{\mathbf{u}^\top \mathbf{x}_{sb}}}$$

eg, linear-gaussian policies (continuous actions)



eg, linear-gaussian policies (continuous actions)

- The mean and std. dev. for the action taken in state s are linear and linear-exponential in $\mathbf{u}_\mu$, $\mathbf{u}_\sigma$

$$\mu(s) = \mathbf{u}_\mu^\top \phi_s \qquad\qquad \sigma(s) = e^{\mathbf{u}_\sigma^\top \phi_s}$$

- The probability density function for the action taken in state s is gaussian

$$\pi_{\mathbf{u}}(a|s) = \frac{1}{\sigma(s)\sqrt{2\pi}} e^{-\frac{(a-\mu(s))^2}{2\sigma(s)^2}}$$

Policy Approximation

Average Reward per step objective

No discounting over the on-policy distribution & prop

The policy gradient theorem and its proof

Gradient Ascent (stochastic)

Approximating the gradient

Eligibilities fns for discrete & continuous actions

A final algorithm

$$v_\pi^\gamma(s) \doteq \sum_{t=0}^{\infty} E_\pi\left[\gamma^t R_{t+1} \mid S_0 = s\right]$$

$$= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi^\gamma(s')\right]$$

$$d_\pi(s) \doteq \lim_{t\to\infty} \Pr\left[S_t = s \mid A_{0:t} \sim \pi\right]$$

exists & assume does not depend on S_0

ergodic

$$d_\pi(s') = \sum_s d_\pi(s) \sum_a \pi(a|s)\, p(s'|s,a)$$

$$\bar{r}(\pi) \doteq \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} R_t$$

assume exists and does not depend on S_0

$$= \sum_s d_\pi(s) \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\, r$$

$$\tilde{v}_\pi(s) \doteq \sum_{t=0}^{\infty}\left(E_\pi\left[\gamma^t R_{t+1} \mid S_0 = s\right] - \bar{r}(\pi)\right)$$

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

$$\tilde{q}_\pi(s,a) \doteq \text{similar}$$

Is $J(\pi) \doteq \sum_s d_\pi(s)\, v_\pi^\gamma(s)$ a good objective?

Or is it the same as $\bar{r}(\pi)$?

$$
\begin{aligned}
J(\pi) &\doteq \sum_s d_\pi(s)\, v_\pi^\gamma(s) \\
&= \sum_s d_\pi(s) \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi^\gamma(s')\right] \\
&= \bar{r}(\pi) + \gamma \sum_s d_\pi(s) \sum_a \pi(a|s) \sum_{s'} p(s'|s,a)\, v_\pi^\gamma(s') \\
&= \bar{r}(\pi) + \gamma \sum_{s'} v_\pi^\gamma(s') \sum_s d_\pi(s) \sum_a \pi(a|s)\, p(s'|s,a) \\
&= \bar{r}(\pi) + \gamma \sum_{s'} v_\pi^\gamma(s')\, d_\pi(s') \\
&= \bar{r}(\pi) + \gamma J(\pi) \\
&= \bar{r}(\pi) + \gamma \bar{r}(\pi) + \gamma^2 \bar{r}(\pi) + \gamma^3 \bar{r}(\pi) + \ldots \\
&= \frac{\bar{r}(\pi)}{1-\gamma}
\end{aligned}
$$

i.e., a scaled $\bar{r}(\pi)$, thus no effect of γ on the ordering of any policies

Gradient Ascent

Let $u \in \mathbb{R}^n$ be the policy parameter

$$u_{t+1} = u_t + \alpha \nabla_u r(\pi_u)$$

Stochastic Gradient Ascent

$$u_{t+1} = u_t + \alpha \widehat{\nabla_u r(\pi_u)}$$

where $E_\pi\left[\widehat{\nabla_u r(\pi_u)}\right] = \nabla_u r(\Pi_u)$

Policy Gradient Theorem

$$\nabla_u r(\pi_u) = \sum_s d_\pi(s) \sum_a \tilde{q}_\pi(s,a) \nabla_u \Pi_u(s,a)$$

$$= E_\pi\left[\tilde{q}_\pi(S_t,A_t) \frac{\nabla_u \Pi_u(S_t,A_t)}{\Pi_u(S_t,A_t)} \right]$$

$$u_{t+1} = u_t + \alpha\left[\tilde{Q}(S_t,A_t) - b(S_t)\right] \frac{\nabla_u \Pi_u(S_t,A_t)}{\Pi_u(S_t,A_t)}$$

Proof of P.G.T.

$$\nabla_u \tilde{v}_\pi(s) \doteq \nabla_u \sum_a \pi_u(a|s)\, \tilde{q}_\pi(s,a)$$

$$= \sum_a \left[\nabla \pi(a|s)\, \tilde{q}_\pi(s,a) + \pi(a|s)\, \nabla \tilde{q}_\pi(s,a) \right]$$

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

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

Re-arranging terms

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

summing over d_π:

$$\sum_s d_\pi(s)\, \nabla \bar{r}(\pi_u) = \sum_s d_\pi(s) \sum_a \nabla \pi(a|s)\, \tilde{q}_\pi(s,a) + \overbrace{\sum_s d_\pi(s)\, \pi(a|s) \sum_{s'} p(s',r|s,a)}^{d_\pi(s')} \nabla \tilde{v}_\pi(s')$$

$$- \sum_s d_\pi(s)\, \nabla \tilde{v}_\pi(s)$$

$$\nabla \bar{r}(\pi_u) = \sum_s d_\pi(s) \sum_a \tilde{q}_\pi(s,a)\, \nabla \pi(a|s)$$

QED

Approximating the Gradient

PGT

$$\nabla_u \bar{r}(\pi_u) = \sum_s d_\pi(s) \sum_a \nabla_u \pi_u(a|s)\, q_\pi(s,a)$$

note $\sum_a \nabla_u \pi(a|s) = 0 \; \forall$

$$= \sum_s d_\pi(s) \sum_a \nabla_u \pi_u(a|s) \left[q_\pi(s,a) - b(s) \right] \qquad \forall b : \mathcal{S} \to \mathbb{R}$$

$$= \sum_s d_\pi(s) \sum_a \pi_u(a|s) \left[q_\pi(s,a) - b(s) \right] \underbrace{\frac{\nabla_u \pi_u(a|s)}{\pi_u(a|s)}}_{g(s,a)}$$

$$= E_\pi \left[\left(q_\pi(S_t, A_t) - b(S_t) \right) g(S_t, A_t) \right]$$

$$u_{t+1} = u_t + \cancel{\alpha(\hat{q}(S_t, A_t) - b(S_t)}$$

$$+ \alpha \left[\hat{q}(S_t, A_t) - b(S_t) \right] \frac{\nabla \pi_u(A_t|S_t)}{\pi_u(A_t|S_t)}$$

e.g.

$$= u_t + \alpha \left[R_{t+1} - \bar{R}_t + \hat{v}(S_{t+1}, w_t) - \hat{v}(S_t, w_t) \right] \frac{\nabla \pi(A_t|S_t)}{\pi(A_t|S_t)}$$

e^u_t

Elig traces:

$$e^u_t = e^u_{t-1} + \frac{\nabla_u \pi_u(A_t|S_t)}{\pi(A_t|S_t)}$$

Final Complete Algorithm

$$\left.\begin{array}{lll} u, e^u & \in \mathbb{R}^n & \text{actor} \\ v, e^v & \in \mathbb{R}^m & \text{critic} \\ \bar{R} & \in \mathbb{R} & \end{array}\right\} \text{init all to } \emptyset$$

On each step, in state S:

Choose A according to $\pi_u(\cdot|S)$

Take A; observe S', R

$$\delta \leftarrow R - \bar{R} + v^T x(S') - v^T x(S)$$

$$\bar{R} \leftarrow \bar{R} + \alpha_R \delta$$

$$e^v \leftarrow \lambda e^v + x(S)$$

$$v \leftarrow v + \alpha_v \delta e^v$$

$$e^u \leftarrow \lambda e^u + \frac{\nabla_u \pi_u(A|S)}{\pi_u(A|S)}$$

$$u \leftarrow u + \alpha_u \delta e^u$$

The algorithms mentioned above are independent of the structure of the policy distribution used in the policy. For discrete actions, the Gibbs distribution is often used. In this paper, for continuous actions, we define the policy such that actions are taken according to a normal distribution, as suggested by Williams [10], with a probability density function defined as $\mathcal{N}(s,a) = \frac{1}{\sqrt{2\pi\sigma^2(s)}}\exp\left(-\frac{(a-\mu(s))^2}{2\sigma^2(s)}\right)$ where $\mu(s)$ and $\sigma(s)$ are respectively the mean and standard deviation of the distribution $\pi(\cdot|s)$.

In our policy parameterization, the scalars $\mu(s) = \mathbf{u}_\mu{}^\mathsf{T}\mathbf{x}_\mu(s)$ and $\sigma(s) = \exp\left(\mathbf{u}_\sigma{}^\mathsf{T}\mathbf{x}_\sigma(s)\right)$ are defined as linear combinations, where the parameters of the policy are $\mathbf{u} = (\mathbf{u}_\mu{}^\mathsf{T}, \mathbf{u}_\sigma{}^\mathsf{T})^\mathsf{T}$, and the features vector in state s is $\mathbf{x}_u(s) = (\mathbf{x}_\mu(s)^\mathsf{T}, \mathbf{x}_\sigma(s)^\mathsf{T})^\mathsf{T}$.

The compatible features $\frac{\nabla_\mathbf{u}\pi(a|s)}{\pi(a|s)}$ depend on the structure of the probability density of the policy. Given that our policy density is a normal distribution, the compatible features for the mean and the standard deviation are [10]:

$$\frac{\nabla_{\mathbf{u}_\mu}\pi(a|s)}{\pi(a|s)} = \frac{1}{\sigma(s)^2}(a-\mu(s))\,\mathbf{x}_\mu(s) \tag{9}$$

$$\frac{\nabla_{\mathbf{u}_\sigma}\pi(a|s)}{\pi(a|s)} = \left(\frac{(a-\mu(s))^2}{\sigma(s)^2} - 1\right)\mathbf{x}_\sigma(s) \tag{10}$$

where $\frac{\nabla_\mathbf{u}\pi(a|s)}{\pi(a|s)} = \left(\frac{\nabla_{\mathbf{u}_\mu}\pi(a|s)}{\pi(a|s)}^\mathsf{T}, \frac{\nabla_{\mathbf{u}_\sigma}\pi(a|s)}{\pi(a|s)}^\mathsf{T}\right)^\mathsf{T}$.

The compatible feature in (9), used to update the parameters $\mathbf{u}_\mu$ of the policy, has a $\frac{1}{\sigma(s)^2}$ factor: the smaller the standard deviation is, the larger the norm of $\frac{\nabla_{\mathbf{u}_\mu}\pi(a_t|s_t)}{\pi(a_t|s_t)}$ is, and vice-versa. We observed that such an effect can cause instability, particularly because $\lim_{\sigma\to 0}\frac{(a-\mu(s))}{\sigma(s)^2} = \infty$.

The generality of the policy-gradient strategy

- Can be applied whenever we can compute the effect of parameter changes
  on the action probabilities, $\nabla_{\mathbf{u}} \pi(a|s)$
- E.g., has been applied to spiking neuron models
- There are many possibilities other than linear-exponential and linear-gaussian
  - e.g., mixture of random, argmax, and fixed-width gaussian; learn the mixing weights
  - drift/diffusion models?

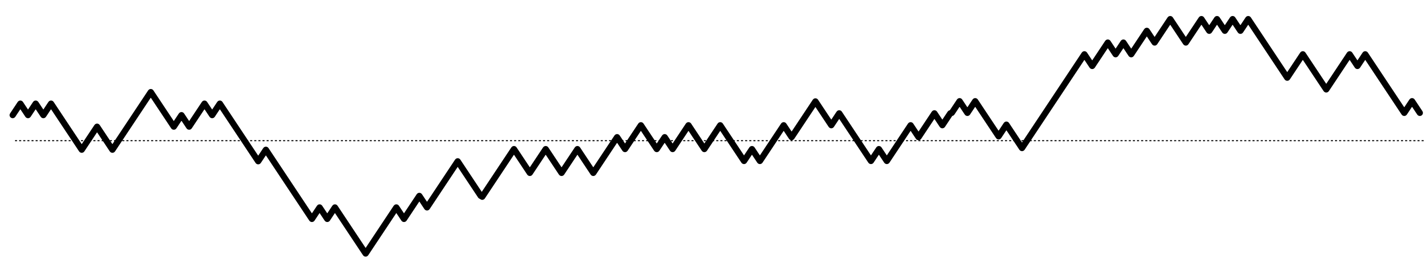
Can policy gradient methods solve the *twitching problem?*

(the problem of decisiveness in adaptive behavior)

- The problem:
  - we need stochastic policies to get exploration
  - but all of our policies have been i.i.d. (independent, identically distributed)
  - if the time step is small, the robot just twitches! 😞
  - really, no aspect of behavior should depend on the length of the time step

Can we design a parameterized policy whose stochasticity is independent of the time step?

- let a “noise” variable take a random walk, drifting but tending back to zero



- *add* it to the action, and adapt its parameters by the PG algorithm (or have several such noise variables)

The generality of the policy-gradient strategy (2)

- Can be applied whenever we can compute the effect of parameter changes
  on the action probabilities, $\nabla_{\mathbf{u}}\pi(a|s)$
- Can we apply PG when outcomes are viewed as action?
  - e.g., the action of "turning on the light"
    or the action of "going to the bank"
  - is this an alternate strategy for temporal abstraction?
- 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 ε-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