Optimizing a Finite-state controller

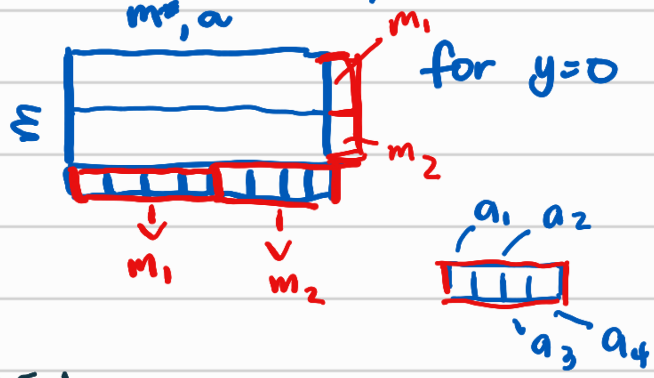
Goal: Optimize a policy π_θ to maximize average rewards

POLICY/STRATEGY : π_θ (parametrized by θ)

$\pi_\theta = \text{Prob}(a, m^* \mid m, y)$ in code: pi(O,M,M*A)

i.e. pi[0,:,:]

$\boxed{M=2}$



markov chain illustration

→ y=1

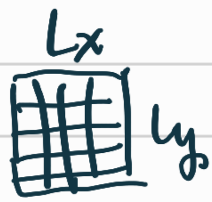
--→ y=0

(memory state transitions only)

*To couple w/ state actions, above the arrow, add the dominant action:

e.g. $\text{Prob}(a=\uparrow, m^*=m_1 \mid m=m_1, y=0) = 0.8$

ENVIRONMENT SET-UP



size: $L_y \times L_x$

target: (L_{x0}, L_{y0})

information / signal : in code: PObs (O, Ly×Lx)

$\text{Prob}(y \mid s) \approx$

compute from particle counts from part 1

Options: (1) time-averaged

(2) time-averaged, symmetrical

(3) snapshot at certain time : c/c_{max}

THRESHOLD!!

$P(y=1 \mid s) = 1$

if $\bar{c}(s) > th$

Some detection model we could use :

(1) conical distribution (w/ tanh functions)

(2) From paper of infotaxis

$$C(s|s_0) = \frac{c_{max}}{\|s-s_0\|_2 + 0.01} e^{-\frac{\|s-s_0\|_2}{\lambda} - \frac{yV_D}{2}}$$

where $\lambda = \sqrt{\frac{D\tau}{1+\frac{v^2\tau}{4D}}}$

Rewards function

In our set-up: reward = 0 if target found

reward = $-(1-\gamma)$ if target not found

Average reward given state s, m:

$$r(s,m) = -(1-\gamma) \sum_{a,s',y} p(s'|s,a) f(y|s) \pi_\theta(a, m'|m,y) \mathbb{1}(x' \neq x_s)$$

$\eta(s,m) \Rightarrow$ occupancy of state-memory pair

$$\eta = (1-\gamma T)^{-1} \rho \qquad T: S \times M \to S \times M$$

$$T(s', m' | s, m)$$

$V(s,m) \Rightarrow$ value of the policy

$$V^T = r^T (1-\gamma T)^{-1}$$

expected return: $G = r^T \eta = V^T \rho$ | For $\gamma \to 1$ $G \approx -(1-\gamma) \cdot$ av. time

Optimization : NPG (Natural Policy Gradient)

pseudocode :

```
init θ [0, M, M*A]
pi = softmax (θ)        -> Boltzman param
value_prev = 0
value_new = 1

until value_new - value prev > tol :
    calculate η, Q
    grad = f × η × Q
    θ = θ + (learning_rate) grad
```