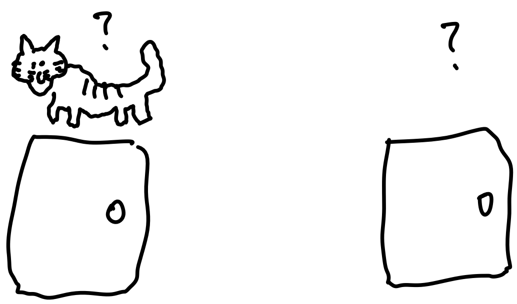
POMDPs

$\pi(s)$ $\quad$ $\pi(a|s)$

Type of uncertainty		Sequential Decision Problem	Algos
Outcome	Alleatory	MDP	Policy Iter Value Iter MCTS, etc.
Model Uncertainty	Epistemic Static	MDP with unknown model	Q-learning Policy Gradient ...
State	Epistemic Dynamic	POMDP	Point-Based VI POMCP, DESPOT



$S = \{TL, TR\}$

$A = \{OL, OR, \text{Listen}\}$

$R = +10$ open empty door, -100 if open tiger door

$\gamma = 0.99$

$O = \{TR, TL\}$ $\quad$ 95% accurate

$\pi_0(o) = L$ $\qquad$ $\pi_1(TR) = TR$

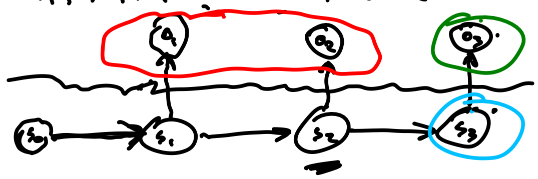
$\pi_1(TL) = TL$

$Q^\pi_1 = 0.85 \cdot 10 - 0.15 \cdot 100$

$= -6.5$

belief over where tiger based on $o_1, o_2, o_3 \ldots$

HMM   Hidden Markov Model



$T(s_1|s_0)$

$$P(s_1|o_1) = \frac{P(o_1|s_1)\,P(s_1)}{P(o_1)} = \frac{P(o_1|s_1)\sum_{s_0} P(s_1|s_0)\,P(s_0)}{P(o_1)}$$

$$\propto P(o_1|s_1)\sum_{s_0} P(s_1|s_0)\,P(s_0)$$

$b_0 \equiv P(s_0)$ $b_0(s) \equiv P(s_0 = s)$

$h_k = (o_1, o_2, o_3, \ldots o_k)$ ← "history" "information state" $(b_0, o_1, o_2 \ldots)$

$$P(s_k|h_k) = \frac{P(o_k|s_k, h_{k-1})\,P(s_k, h_{k-1})}{P(h_k)}$$

$o_3 \perp h_{k-1} \mid s_3$

$P(o_k|s_k, h_{k-1}) = P(o_k|s_k)$

$$\propto P(o_k|s_k)\sum_{s_{k-1}} P(s_k|s_{k-1}, h_{k-1})\,P(s_{k-1}|h_{k-1})\,P(h_{k-1})$$

$s_k \perp h_k \mid s_{k-1}$

$$\rightarrow \underbrace{P(s_k|h_k)}_{b_k} \propto P(o_k|s_k)\sum_{s_{k-1}} P(s_k|s_{k-1})\,\underbrace{P(s_{k-1}|h_{k-1})}_{b_{k-1}}$$

b_0
loop k
  observe o_k
  $b_{k+1} \leftarrow \text{UpdateBelief}(b_k, o_k)$

$$\boxed{b'(s') \propto O(o|s')\sum_{s} T(s'|s)\,b(s)}$$

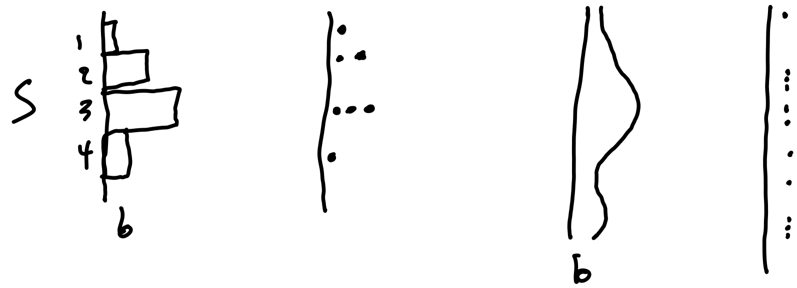
UpdateBelief (b, o)
  for $s' \in S$
    $b'(s') \leftarrow O(o|s')\sum_{s} T(s'|s)\,b(s)$] Approximate with MC simulations
  $b' \leftarrow b' / \sum_{s'} b(s')$ ← "normalization"
  return b'

$O(|S|^2)$

Particle Filtering

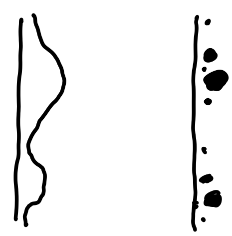
PF with Rejection Sampling "Unweighted"

Need <u>only</u> $s', o = G(s, v)$!



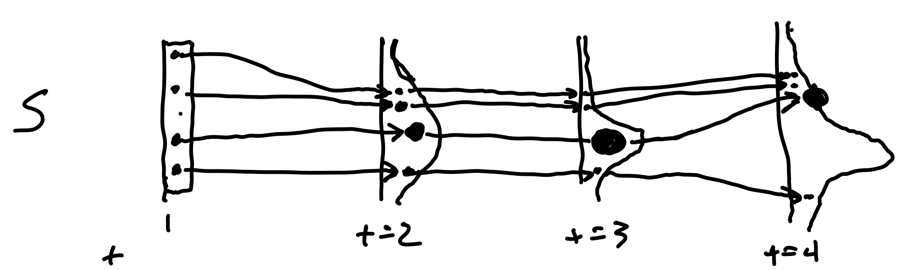

```
BeliefUpdate(b, o)
    b' ← ∅          ← multiset / vector
    while |b'| < n
        s = randomly sample from b
        s', õ = G(s, v)
        if õ = o        ← low probability
            insert s' into b'
    return b'
```

Weighted Particle Filter $b = \{(s_1, w_1), (s_2, w_2) \ldots\}$



UB(b, o) UB(b, a, o)

```
UB(b, o)
    b' ← ∅
    for i in 1:|b|
        s' ← G(s_i, v_i)
        w' ← O(o | s') w_i
        insert (s', w') into b'
    return b'
```



Particle Depletion

few particles that represent the true state

Weighted PF with Resampling

$b = (s_1, s_2, s_3 \dots)$

UB(b, o)

$\tilde{b} \leftarrow \emptyset$

for i in $1:n$

  $s' = G(s_i, v_i)$

  $w' = O(o \mid s')$

  insert (s', w') into $\tilde{b}$ ← weighted particles

$b' =$ sample n particles from $\tilde{b}$

return b'

WPF w/R can be implemented in $O(n)$?!?!

$b' = \emptyset$

for i in $1:n$

  insert rand($\tilde{b}$) into b'

← $O(n)$ if no preprocessing

```
r = rand()
i = 1
while u < r
    u += w_i
    i += 1
```

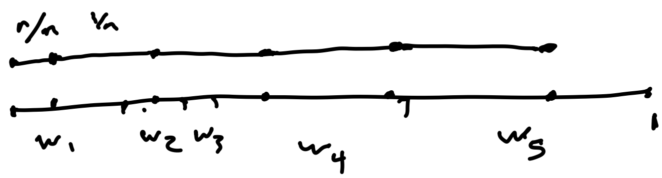
method 1: Build alias table

method 2: Low variance resampling technique from *Probabilistic Robotics* – Thrun

$r = \text{rand}()$

$n = 5$

$O(n)$



$\tilde{b}$

$b' = (s_1, s_2, s_4, s_4, s_5)$