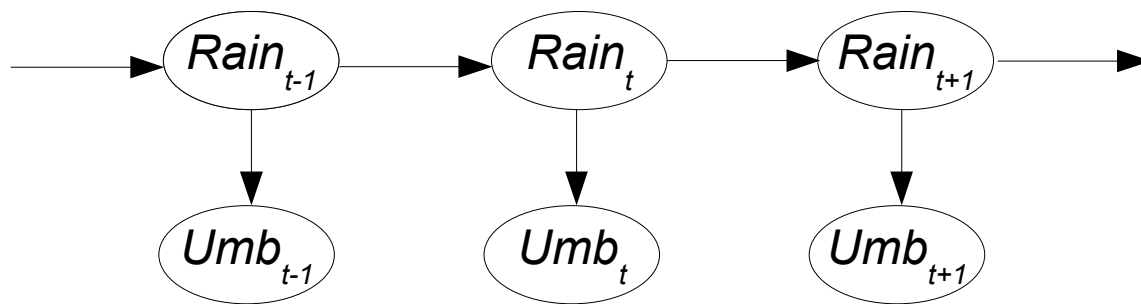


# Hidden Markov Models (HMMs)

- A process with exactly one discrete state variable
- Possible values are possible states of the world



# Simplified matrix algorithms

- Transition and sensor models are matrices of probabilities

$$\mathbf{T}_{ij} = P(X_t=j | X_{t-1}=i) \quad S \times S \text{ matrix}$$

E.g., for umbrella world  $\mathbf{T} = \begin{matrix} & 0.7 & 0.3 \\ & 0.3 & 0.7 \end{matrix}$

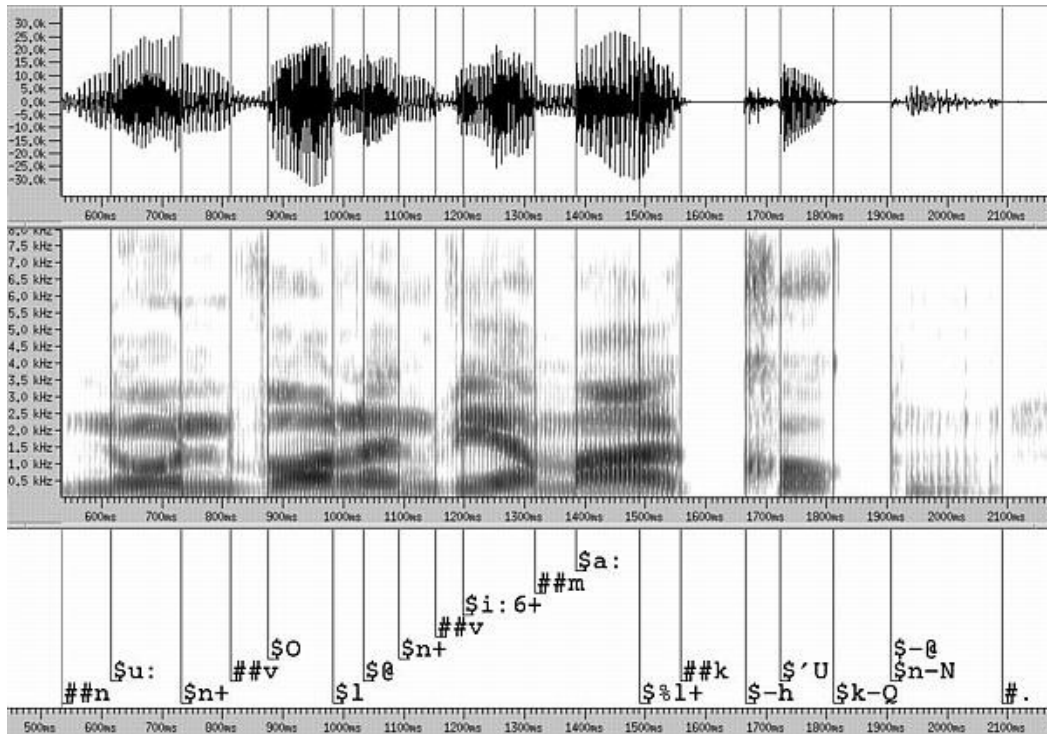
$\mathbf{O}_t$  - diagonal matrix of  $P(e_t | X_t=i)$

- Column vectors for forward and backward messages

$$\mathbf{f}_{1:t+1} = \alpha \mathbf{O}_{t+1} \mathbf{T}' \mathbf{f}_{1:t}$$

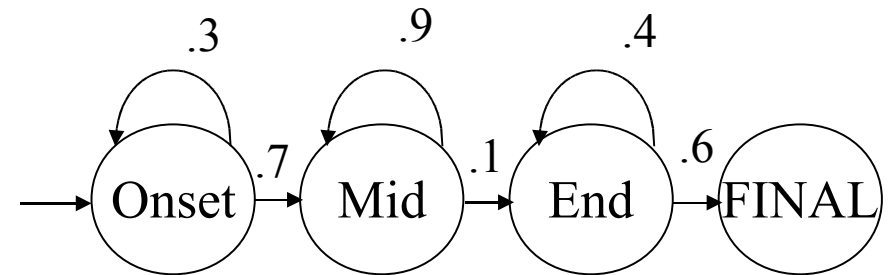
$$\mathbf{b}_{k+1:t} = \mathbf{T} \mathbf{O}_{k+1} \mathbf{b}_{k+2:t}$$

# HMM use: speech recognition



Picture from: Kohler (2000). *Investigating unscripted speech: Implications for phonetics and phonology*. In: *Festschrift for Björn Lindblom, Phonetica* 57.

Phone HMM for [m]:



Output probabilities:  
 $P(\text{features}|\text{phone}) =$

Onset	Mid	End
$C1 : .5$	$C4 : .2$	$C7 : .1$
$C2 : .2$	$C5 : .7$	$C8 : .5$
$C3 : .3$	$C6 : .1$	$C9 : .4$

# Inference in speech recognition

- Phone from sound wave
- Word(s) from phone sequence
- Sentence(s) from word sequence
  
- Natural language understanding:
  - Meaning from sentence
  - Intention from meaning

# Decision-making under uncertainty

Chapters 16.1-3, 16.5, 17.1-3

# Acting under uncertainty

- So far, the agent has been reasoning about the world, but not acting in it
- What action to chose under uncertainty?

# Utility theory

- Rational agent decides on a course of action that will maximize expected utility
- Decision theory = probability theory + utility theory
- Utility or value functions constructed based on rational preferences for outcomes
  - orderability
  - transitivity
  - continuity
  - substitutability
  - monotonicity

# Decision networks

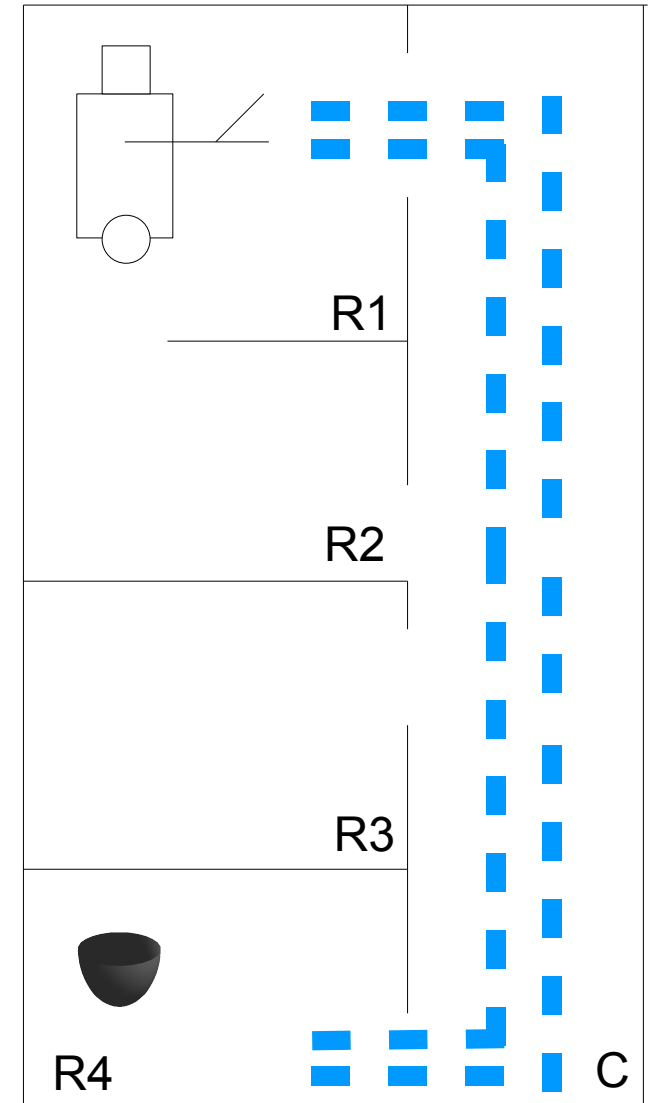
- Also called influence diagrams: Bayes nets augmented with special nodes for actions and utilities
- Chance nodes, decision nodes, utility nodes
- Take road A and you have 90% chance to find a brick of gold if it's been left there by a Wumpus. Take road B and you have 50% chance to find a mountain of gold, and a 50% chance of being run over by a bus.

# Evaluating decision networks

- Set the evidence variables for the current state
- For each possible value of the decision node:
  - Set decision node to that value
  - Calculate posterior probabilities for parent nodes of utility node, using some probabilistic inference algorithm
  - Calculate the resulting utility for the action
- Return the action with the highest utility

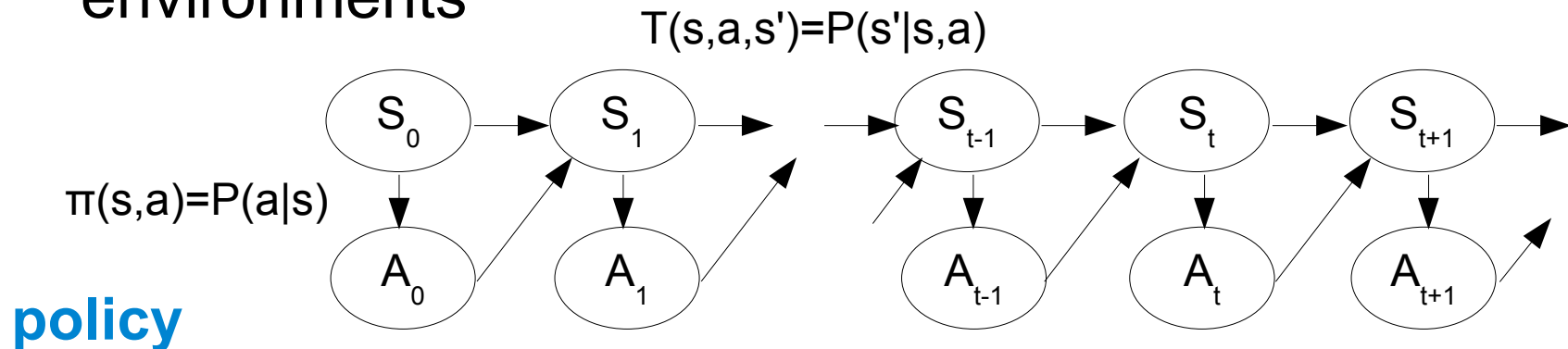
# Decision-making over time

- Choosing an action now (today) given that we'll have to choose again in a moment (tomorrow)
- Utility depends on a sequence of decisions
- Delayed utility signal



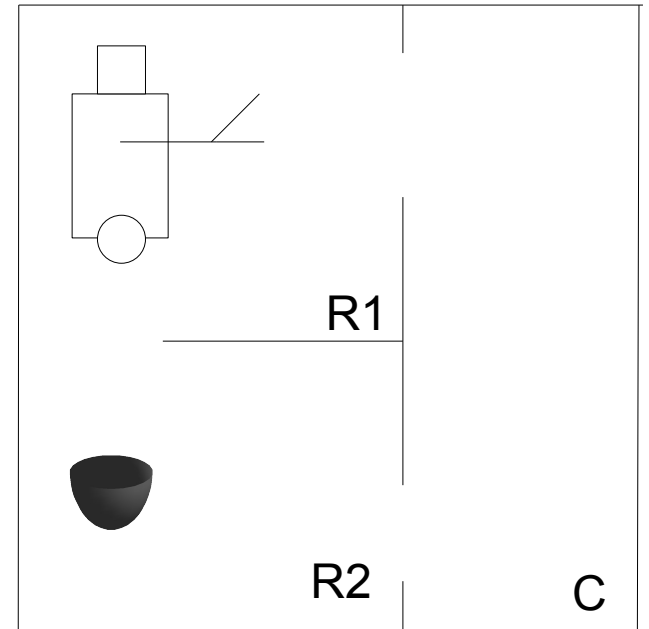
# Decision processes

- Fully observable, stochastic (non-deterministic) environments



- Markov Decision Processes (MDPs): decision processes that have the Markov property (first-order)
  - initial state  $S_0$
  - transition model  $T(s,a,s')=P(s'|s,a)$
  - reward function  $R(s)$

# Graphical transition model



- **a.k.a transition graph**

# Policies and utility

- Policies are solutions of the MDP that specify action distributions for every possible state
- Optimal policies  $\pi^*$  maximize expected utility
- In a MDP, utility is a function of an experience history
  - $h = \langle s_0, a_0, s_1, a_1, \dots, s_t, a_t, \dots \rangle$
- What's a terminal state? When does a history end?
  - finite-horizon MDPs: there are  $T$  steps

$$U_h([s_0, s_1, \dots, s_T]) = \sum_{t=0}^T R(s_t)$$

- infinite-horizon MDPs: there are  $\infty$  steps

$$U_h([s_0, s_1, \dots, s_T]) = \sum_{t=0}^{\infty} \gamma^t R(s_t), \quad 0 < \gamma < 1$$

# Policies and utility cont.

- Optimal policy  $\pi^* = \operatorname{argmax}_{\pi} \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi \right]$
- Example optimal policy

# Policies: deterministic or stochastic

- Deterministic policies always return an action with probability 1:  $\pi(s, a) = 1$  and  $\pi(s, a') = 0$  for all  $a' \neq a$
- Today: deterministic policies

# Finding $\pi^*$ : value iteration

- Idea: calculate the utility of each state, then use state utility (called state  $\pi$  **value**) to find the optimal action
- Utility of a state is the expected utility of state sequences that may follow it

- depends on the policy

$$U^\pi(s) = E \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \mid \pi, s_0 = s \right]$$

- real value of a state is  $U^* = U^{\pi^*}$
- how to choose action that maximizes expected utility?

$$\pi^*(s) = \operatorname{argmax}_a \sum_{s'} T(s, a, s') U^*(s')$$

# Relationship between state utilities

- The utility of a state is its immediate reward plus the expected discounted utility of the next state, assuming optimal action choice:

$$U^*(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s') U^*(s')$$

Bellman equation

$U^*$  is the unique solution

- $N$  states gives  $N$  equations in  $N$  variables, but they are non-linear
- Use an **iterative** approach: starting with random utilities

$$U_{t+1}(s) := R(s) + \gamma \max_a \sum_{s'} T(s, a, s') U_t(s')$$

Bellman update

... until  $U$  values **converge** to an equilibrium

# Value iteration algorithm

```
function VALUE_ITERATION(mdp,  $\epsilon$ ) returns a  
                                utility function  
    inputs: mdp with states  $S$ , transition model  $T$ ,  
            reward function  $R$ , discount  $\gamma$   
             $\epsilon$  max error allowed  
  
 $U \leftarrow 0, U' \leftarrow 0$   
repeat  
     $U \leftarrow U'; \delta \leftarrow 0$   
    for each state  $s$  in  $S$  do  
         $U'[s] \leftarrow R[s] + \gamma \max_a \sum_{s'} T(s, a, s') U[s']$   
        if  $|U'[s] - U[s]| > \delta$  then  $\delta \leftarrow |U'[s] - U[s]|$   
until  $\delta < \epsilon(1 - \gamma) / \gamma$ 
```

# Value iteration convergence

- Bellman update is a contraction by a factor of  $\gamma$  on the space of utility vectors

# Policy iteration

- If one action is clearly better than all others, the actual state values need not be precise
- Policy iteration
  - 1. **policy evaluation**: given  $\pi_i$ , calculate  $U_i = U^{\pi_i}$
  - 2. **policy improvement**: calculate new policy  $\pi_{i+1}$  that maximizes expected utility using one-step lookahead based on  $U_i$

# Policy evaluation

- Policy evaluation step easier than iteratively calculating  $U^*$  because the policy is fixed at each step

$$U_i(s) = R(s) + \gamma \sum_{s'} T(s, \pi(s), s') U_i(s')$$

equations are linear (no max. operator)

- Exact policy evaluation:  $O(n^3)$  for  $n$  states with linear algebra methods
- For large spaces, **modified policy iteration** performs a few steps of the value iteration algorithm to estimate the values instead
- Policy improvement is straightforward (max. expected utility):

$$\pi_{i+1}(s) = \operatorname{argmax}_a \sum_{s'} T(s, a, s') U_i(s')$$

# Next time

- What happens when the agent doesn't know the transition model?