CS 4649/7649
Robot Intelligence: Planning

Reality

Sungmoon Joo

School of Interactive Computing
College of Computing
Georgia Institute of Technology

*Slides based in part on Dr. Mike Stilman's lecture slides

Administrative – Final Project

• CS7649
- project topic decision, grouping: Due Oct. 23 → update Wiki (group, description)
- project proposal: Due Oct. 30, 2-3page (motivation, technical gap, approach, expected result)
- project final report: Due Dec. 4, 23:59pm, conference-style paper (format is on the course web)
- project presentation: Dec. 11, 11:30am - 2:20pm

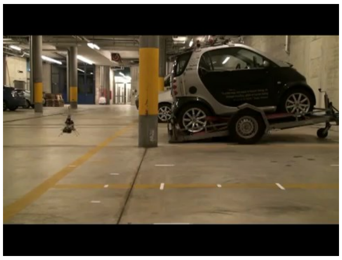
*there may be meetings between project teams and the instructor to see if projects are progressing as scheduled.

• CS4649
- project reviewer assignment: Oct. 28
- proposal review report: Due Nov. 6
- project review report(for the assigned project): Due Dec. 11, 11:30am
- project presentation review*(for all presentation): Due Dec. 11, 2:20pm

*presentation review sheets will be provided

Execution of Plan

"We have a plan. Let's execute the plan"



Execution of Plan

"We have a plan. Let's execute the plan using a controller"

- Reasonable solution in some domains

ex. Planner decides arm trajectory. (servo) Controller tracks it

- Hierarchical decomposition

:High level plan → Low level control

:Not just Classical & Motion, but also Control(execution)

:Allows one to focus on a specific aspect of a robot problem

So, is there any problem?

***sometimes the distinction between planner and controller is subtle**

Two Sources of Error

- **Suppose you try to execute your plan on a robot,**
  -chances are it will not succeed.
  -there are two general reasons

- **What are they?**

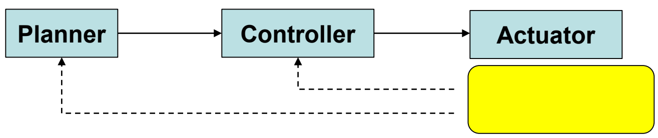
S. Joo (sungmoon.joo@cc.gatech.edu) 10/21/2014 5

Two Sources of Error

"*Principles of control*
...
The scientific formulation of a control problem must be based on two kinds of information: (A) the behavior of the system must be described in a mathematically precise way; (B) the purpose of control (criterion) and the environment (disturbances) must be specified, again in a mathematically precise way.
..."

(www.britannica.com)

Do we have the two kinds of information?



S. Joo (sungmoon.joo@cc.gatech.edu) 10/21/2014 6

Two Sources of Error

Your models are Incorrect!

- **Sensing & State Estimation → Uncertainty**
  - Sensors have noise
  - You don't know exactly what the state is (e.g. mapping, localization,…)

- **Action Execution → Uncertainty**
  - Your actuators do not do what you tell them to
  - The system responds differently than you expect
    : Friction gears, air resistance, etc.

Have we done nothing about uncertainty?

Uncertainties in estimation and execution are not explicitly considered!

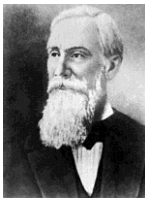
Re-Planning
- Fast Planners make this more feasible

Efficient Plan Reconstruction
- D* and variants make it possible to speed up re-planning

Optimizing Plans for Success
- Voronoi Diagrams maximize distance to obstacles
- Potential fields push us away from obstacles

Universal Plans or Policies
- Provide actions for unexpected or unlikely outcomes

Markov Processes: Formalizing Uncertainty



Pafnuty Chebyshev
(1821~1894)

Andrey Markov
(1856~1922)

Gives us an upper bound on the probability that some random variable is at least a given distance away from the expected value

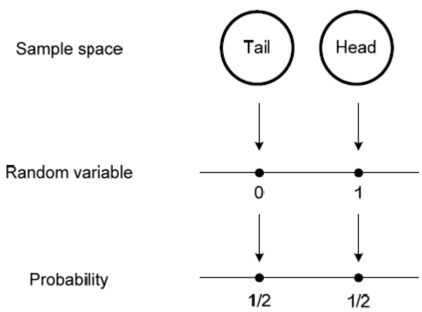
- Chebyshev inequality: $P(|X - E(x)|) \geq a\sigma) \leq \frac{1}{a^2}$
- If X is a R.V. with S.D. = σ :

  - Probability that the outcome of X is more than $a\sigma$ away from its mean is no more than 1/ a^2

  - In any distribution

Random Variable & Probability

(Scalar) Random Variable
- a variable but rather a function that maps events (or outcome) to real numbers, each with an associated probability



$x : \Omega \rightarrow \mathbb{R}$

(i) $x(Head) = 1$, and $x(Tail) = 0$

(ii) $p(x = 1) = 0.5$, and $p(x = 0) = 0.5$.

* A random vector is a vector-valued random variable whose components are scalar random variables.

Markov Processes: Formalizing Uncertainty

A random process is a collection of continuous-valued random variables indexed by a continuous-valued parameter

$$x(t_i), t_0 \leq t_i \leq t_f.$$

A Markov Process is a random process which can be completely specified by giving the density functions $p(x(t)|x(\tau))$ and $p(x(\tau))$ for all $t, \tau \in (t_0, t_f)$

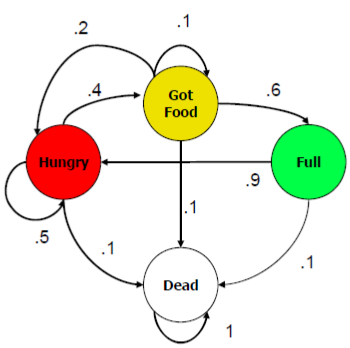
We will, mostly, work on discrete version !
- Markov sequence or Markov chain

A random sequence is a collection of random variables (scalar or vector) indexed by a discrete-valued parameter such as $x_0, x_1, \ldots, x_N$

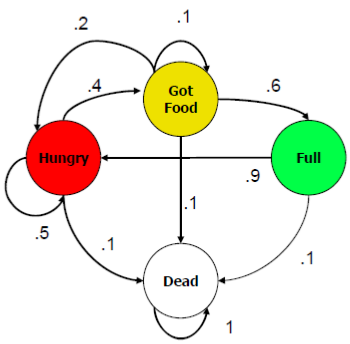
A random sequence is called Markov Sequence if

$$p(x_{k+1}|x_0, \ldots, x_k) = p(x_{k+1}|x_k), \forall k \in 0, 1, \ldots, N$$

Markov Processes: Caveman's World



Markov Processes: Caveman's World

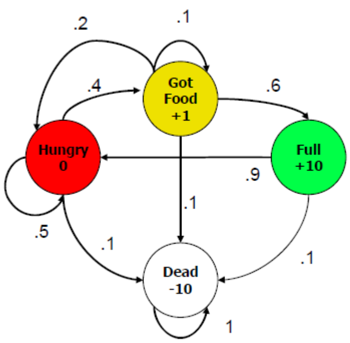


States $\Sigma = \{s_1, ..., s_n\}$

Transition Model

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ & \cdots & & \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

Markov Processes with Rewards



$$p_{ij} = P(x_{k+1} = j | x_k = i)$$

States $\Sigma = \{s_1, ..., s_n\}$

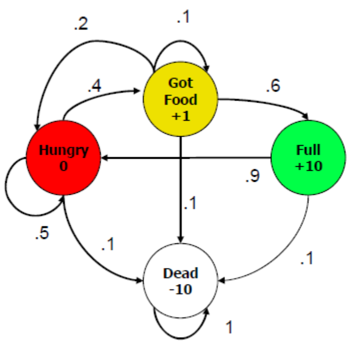
Transition Model

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ & \cdots & & \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

Rewards

$$R = \begin{bmatrix} r_1 & r_2 & \cdots & r_n \end{bmatrix}^T$$

Markov Processes with Rewards



Value

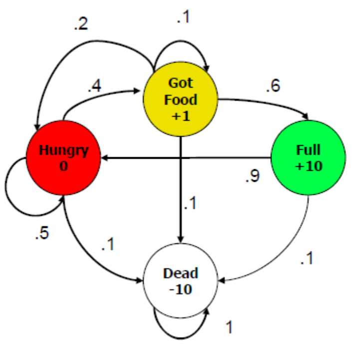
- How good is it to be in a state?
- Sum of DISCOUNTED expected rewards

$$J^k = r + \lambda J^{k-1}$$

λ is a discount factor

Why?

Markov Processes with Rewards



Value

- How good is it to be in a state?
- Sum of DISCOUNTED expected rewards

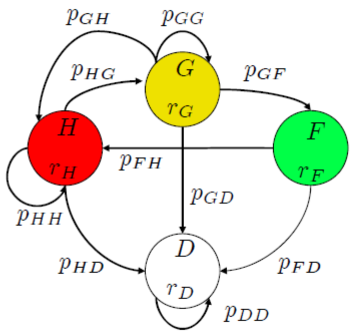
$$J_s = r_s + \lambda(\text{Expected Rewards})$$

$$J_s = r_s + \lambda(p_{s1}J_1 + p_{s2}J_2 \ldots p_{sn}J_n)$$

λ is a discount factor

Why? Infinite Values
Catastrophic Events

Find the Value of a State: J(s)



Value

$$J_s = r_s + \lambda(\text{Expected Rewards})$$
$$J_s = r_s + \lambda(p_{s1}J_1 + p_{s2}J_2 \dots p_{sn}J_n)$$

$$p_{ij} = P(x_{k+1} = j | x_k = i)$$

Find the Value of a State: J(s)

$$J = \begin{bmatrix} J_1 & J_2 & \dots & J_n \end{bmatrix}^T$$

$$P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ & \dots & & \\ p_{n1} & p_{n2} & \dots & p_{nn} \end{bmatrix}$$

$$R = \begin{bmatrix} r_1 & r_2 & \dots & r_n \end{bmatrix}^T$$

Value

$$J_s = r_s + \lambda(\text{Expected Rewards})$$
$$J_s = r_s + \lambda(p_{s1}J_1 + p_{s2}J_2 \dots p_{sn}J_n)$$

$$J = R + \lambda PJ$$
$$J - \lambda PJ = R$$
$$(I - \lambda P)J = R$$
$$J = (I - \lambda P)^{-1}R$$

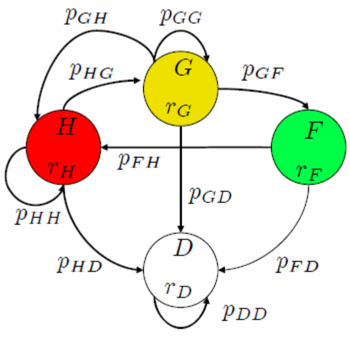
We have an analytic solution for Value!

Actually not so exciting. Why? ~ $O(n^3)$

Summary I

- Markov Processes represent uncertainty in state transitions
- It is possible to determine the overall value of a state
- Value can be found analytically – but this is little bit SLOW
- What's next?

Finding Value with Value Iteration



How do we find the value within k steps?

$$J_s^1 = r_s \quad \text{(1 step)}$$

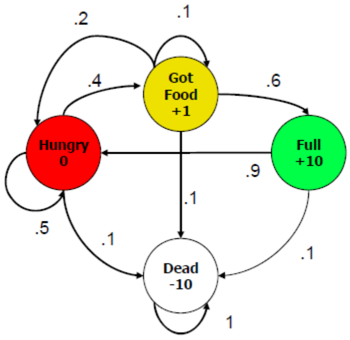
$$J_s^2 = r_s + \lambda \sum_{j=1}^{n} p_{sj} J_j^1 \quad \text{(2 steps)}$$

$$J_s^k = r_s + \lambda \sum_{j=1}^{n} p_{sj} J_j^{k-1} \quad \text{(k steps)}$$

Value Iteration in Caveman's World

$$J_s^k = r_s + \lambda \sum_{j=1}^{n} p_{sj} J_j^{k-1}$$

$$J_H^2 = 0 + .9\,(.5 * 0 + .4 * 1 + .1 * (-10))$$



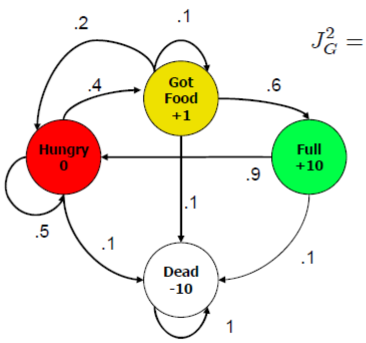
$$\lambda = .9$$

	H	G	F	D
1	0	1	10	-10
2	-.54			

Value Iteration in Caveman's World

$$J_s^k = r_s + \lambda \sum_{j=1}^{n} p_{sj} J_j^{k-1}$$

$$J_G^2 = 1 + .9\,(.2 * 0 + .1 * 1 + .6 * 10 + .1 * (-10))$$

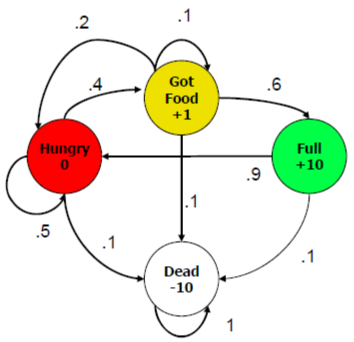


$$\lambda = .9$$

	H	G	F	D
1	0	1	10	-10
2	-.54	5.69		

Value Iteration in Caveman's World

$$J_s^k = r_s + \lambda \sum_{j=1}^{n} p_{sj} J_j^{k-1}$$



$\lambda = .9$

	H	G	F	D
1	0	1	10	-10
2	-.54	5.69	9.1	-19
3	.06	4.61	7.85	-27.1
4	-.75	3.23	7.61	-34.39
		...		
99	-39.08	-34.71	-30.66	-100.0
100	-39.09	-34.71	-30.66	-100.0

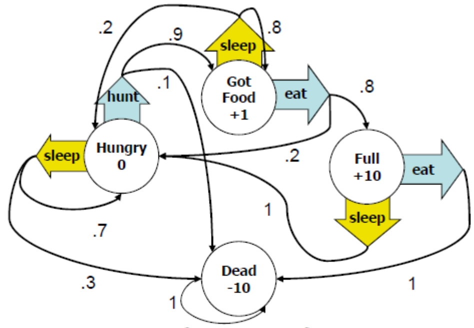
Value Iteration is Guaranteed to Converge

Summary II

- Markov Processes represent uncertainty in state transitions
- It is possible to determine the overall value of a state
- Value iteration is a simple and effective algorithm

What happens when we have actions?

Value of Free Will: Markov Decision Process



States $\Sigma = \{s_1, \dots, s_n\}$ **Actions** $A = \{a_1, \dots, a_m\}$

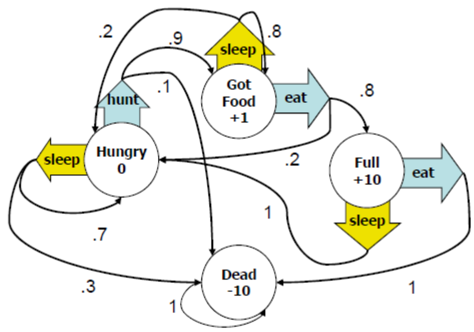
Rewards $R = \{r_1, \dots, r_n\}$

Transition Model

$$P_{ij}^{l} = P(\text{next} = s_j | \text{ current} = s_i \text{ and action} = a_l)$$

* $p_{ij} = P(x_{k+1} = j | x_k = i)$

Value of Free Will: Markov Decision Process



Policy maps states to actions

Optimal Policy gives the best action for any state

J_s^* = Expected value of state **s** when following optimal policy

MDP: Optimal Policies

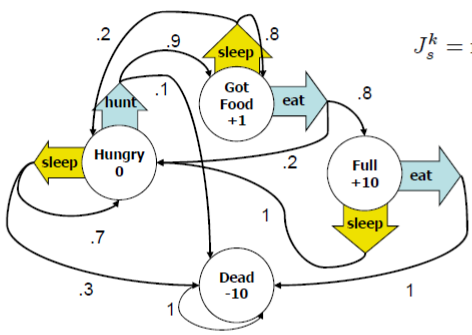
Let J_s^k be the highest expected sum of discounted rewards from **s** in k steps

- Highest expected value in 1 step: $J_s^1 = r_s$
- Highest expected value in 2 steps: $J_s^2 = \max_l \left[r_s + \lambda \sum_{j=1}^{n} p_{sj}^l J_j^1 \right]$
- Highest expected value in t steps: $J_s^k = \max_l \left[r_s + \lambda \sum_{j=1}^{n} p_{sj}^l J_j^{k-1} \right]$

- Bellman's Equation

- Bellman Update

MDP: Value Iteration



$$J_s^k = \max_l \left[r_s + \lambda \sum_{j=1}^{n} p_{sj}^l J_j^{k-1} \right]$$

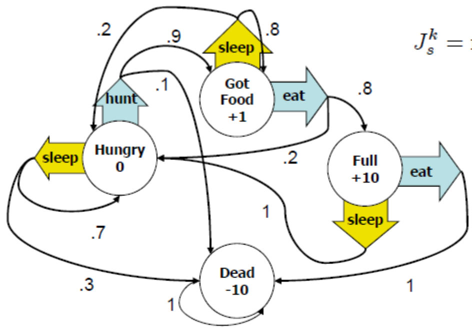
	H	G	F	D
1	0	1	10	-10
2	-.09			

$$J_H^{sleep} = 0 + 0.9(0.7 \times 0 + 0.3 \times (-10)) = -2.7$$

$$J_H^{hunt} = 0 + 0.9(0.9 \times 1.0 + 0.1 \times (-10)) = -0.09$$

Optimal action at H - Hunt

MDP: Value Iteration



$$J_s^k = \max_l \left[r_s + \lambda \sum_{j=1}^{n} p_{sj}^l J_j^{k-1} \right]$$

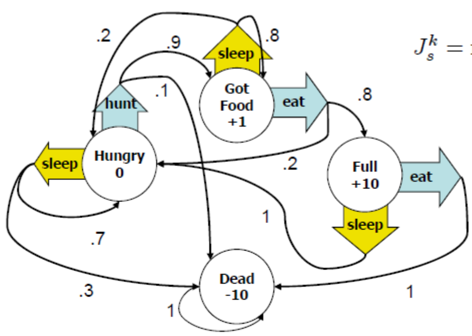
	H	G	F	D
1	0	1	10	-10
2	-.09	8.2		

$$J_G^{sleep} = 1 + 0.9(0.2 \times 0 + 0.8 \times 1) = 1.72$$

$$J_G^{eat} = 1 + 0.9(0.8 \times 10 + 0.2 \times 0) = 8.2$$

Optimal action at G - Eat

MDP: Value Iteration



$$J_s^k = \max_l \left[r_s + \lambda \sum_{j=1}^{n} p_{sj}^l J_j^{k-1} \right]$$

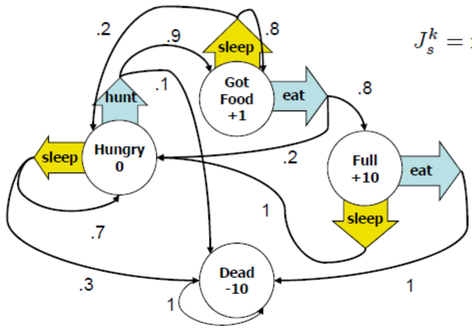
	H	G	F	D
1	0	1	10	-10
2	-.09	8.2	10	-19
		…		
97	-7.16	2.27	3.56	-100

$$J_F^{sleep} = 10 + 0.9(1 \times 0) = 10$$

$$J_F^{eat} = 10 + 0.9(1 \times (-10)) = 1$$

Optimal action at F - Sleep

MDP: Value Iteration



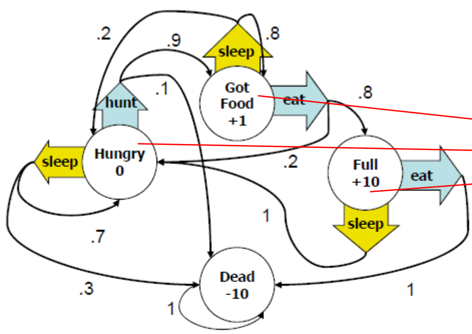
$$J_s^k = \max_l \left[r_s + \lambda \sum_{j=1}^n p_{sj}^l J_j^{k-1}\right]$$

	H	G	F	D
1	0	1	10	-10
2	-.09	8.2	10	-19
		…		
97	-7.16	2.27	3.56	-100

$$J_D = -10 + 0.9(1 \times -10)$$
$$= -19$$

No choice at D

MDP: Optimal Policy



Optimal Policy

$$\pi_s = \text{argmax}_l \left[\sum_{j=1}^n p_{sj}^l J_j^*\right]$$

Hunt - Eat - Sleep

	H	G	F	D
1	0	1	10	-10
2	-.09	8.2	10	-19
100	-7.16	2.27	3.56	-100