Today

Introduction to stochastic dynamic programming

- Refresher on Markov chains
- Definition of policies in the stochastic context
- Examples:
  - Stochastic shortest path

Homework Assignment Grading

MICHIGAN ENGINEERING
UNIVERSITY of MICHIGAN ■ COLLEGE of ENGINEERING

I grade a subset of the solutions to homework assignments

Example: For assignment 1 I graded question 5

Solutions to all problems are on Canvas – carefully review these

Important:

Matlab Code: The code you develop for all assignment questions must be submitted on Canvas. The hardcopy should include a copy of the commented code, a screen shot of the output, and an explanation of the output

Markov Chain Refresher

- A random variable that evolves over time follows a *stochastic process*

- A *Markov chain* is a particular kind of stochastic process in which the states are discrete

- We will be interested mainly in discrete time Markov chains

Stochastic Processes

Suppose we observe some characteristic of a system at discrete points in time…

Let X_t be the value associated with the characteristic at time *t*

A **discrete-time stochastic process** is a description of the relation between the random variables X_0 , X_1, $X_{2,\ldots}$

Markov Chains

A discrete-time stochastic process is a **Markov chain** if, for t = 0,1,2…, and for all states

$$P(X_{t+1} = i_{t+1} | X_t = i_t, X_{t-1} = i_{t-1}, \ldots, X_1 = i_1, X_0 = i_0)$$

$$= P(X_{t+1} = i_{t+1} | X_t = i_t)$$

The probability distribution of the state at time t+1 depends only on the state at time t

The vector $\mathrm{q} = (\mathrm{q}_1, \mathrm{q}_2, \ldots, \mathrm{q}_\mathrm{N})$ is the **initial probability distribution** for the Markov chain at time 0 where $P(\mathbf{X}_0 = i) = q_i$

Transition Probabilities

We refer to $P(X_{t+1} = j | X_t = i) = p_{ij}^t$ as the **transition probability**

If transition probabilities do not change over time the Markov chain is a **stationary Markov chain**

The transition probability matrix for an N-state stationary Markov chain is:

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{bmatrix}$$

n-Step Transition Probabilities

If a stationary Markov chain is in a state *i* at stage *m*, the probability that *n* stages later the Markov chain is in state *j* is

$$P(X_{m+n} = j | X_m = i) = P(X_n = j | X_0 = i) = p_{ij}(n)$$

where $p_{ij}(n)$ is the *ij*th element of P^n, called the ***n*-step probability**.

For example, for n=2

$$p_{ij}(2) = \sum_{k=1}^{N} p_{ik} p_{kj}$$

Example

If for a 2 state stationary system:

$$P = \begin{array}{c} \\ \textit{State 1} \\ \textit{State 2} \end{array} \begin{array}{c} \begin{array}{cc} \textit{State 1} & \textit{State 2} \end{array} \\ \begin{bmatrix} .90 & .10 \\ .20 & .80 \end{bmatrix} \end{array}$$

Then $P(\mathbf{X}_2 = 1|\mathbf{X}_0 = 2) = P_{21}(2)$ = element (2,1) of P^2:

$$P^2 = \begin{bmatrix} .90 & .10 \\ .20 & .80 \end{bmatrix} \begin{bmatrix} .90 & .10 \\ .20 & .80 \end{bmatrix} = \begin{bmatrix} .83 & .17 \\ .34 & .66 \end{bmatrix}$$

Important Definitions

- Given two states, *i* and *j*, a **path** from *i* to *j* is a sequence of transitions that begins in *i* and ends in *j*, such that each transition has a positive probability
- A state *j* is **reachable** from state *i* if there is a path leading from *i* to *j*
- Two states, *i* and *j,* **communicate** if *j* is reachable from *i*, and *i* is reachable from *j*.
- A set of states *S* in a Markov chain is a **closed set** if no state outside of *S* is reachable from any state in *S*

Classification of States

States of a Markov chain can be classified into different **types:**

- A state *i* is an **absorbing state** if $p_{ii}=1$
- A state *i* is a **transient state** if there exists a state *j* that is reachable from *i*, but the state *i* is not reachable from state *j*
- All other states are **recurrent states**

Exercise: For the following transition probability matrix draw a graphical representation and identify the "type" for each state.

$$P=\begin{bmatrix} .5 & .5 & 0 & 0 & 0 \\ .7 & .3 & 0 & 0 & 0 \\ 0 & 0 & .4 & .6 & 0 \\ 0 & 0 & .5 & .2 & .3 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Today

Reading: Today we will start covering topics discussed in Chapter 4.1-4.3 of Puterman (on Canvas)

Following are applications we will cover in future classes:

- Inventory control
- Medical treatment decisions
- Finance
- Zombies
- Others….

Markov Decision Process (MDP)

Following is the standard form of a finite horizon MDP:

- Time horizon: $t \in \{1,2,\dots,N\}$
- States: $s_t \in S$
- Actions: $a_t \in A$
- Rewards: $r_t(s_t, a_t)$
- Discount rate: λ
- Transition Probabilities: $\mathrm{p}(\mathrm{s_{t+1}}|\mathrm{s_t}, \mathrm{a_t})$
- Optimality Equations:

$$v_t(s_t) = \min_{a_t \in A}\left\{r_t(s_t, a_t) + \lambda \sum_{s_{t+1} \in S} p(s_{t+1}|s_t, a_t)\, v_{t+1}(s_{t+1})\right\}, \quad \forall s_t$$

$$v_N(s_N) = R(s_N), \quad \forall s_N$$

Decision Rules and Policies

- A **policy** is a collection of decision rules $\pi = (d_1, d_2, \dots, d_{N-1})$
- A **decision rule**, $d_t(s_t) \in A_{s_t}$ defines the **action** to take in a given state, s_t, and stage, t
- The **history** at stage t is defined as:

$$h_t = (s_1, a_1, s_2, a_2, \dots, s_{t-1}, a_{t-1}, s_t) = (h_{t-1}, a_{t-1}, s_t)$$

There are four types of decision rules:

History Dependence	**Action Choice**	
	Deterministic	Randomized
Markovian	$d_t(s_t) \in A_{s_t}$ D_t^{MD}	$q_{d_t(s_t)} \in P(A_{s_t})$ D_t^{MR}
History Dependent	$d_t(h_t) \in A_{s_t}$ D_t^{HD}	$q_{d_t(h_t)} \in P(A_{s_t})$ D_t^{HR}

Induced Stochastic Process

- A policy **induces** a particular stochastic process. In a finite horizon MDP the set of possible sample paths is

$$\Omega = S \times A \times S \times A \times \cdots \times A \times S = \{S \times A\}^{N-1} \times S$$

- A **sample path,** $\omega \in \Omega$, defines the states and actions for a realization of the induced stochastic process

$$\omega = (s_1, a_1, s_2, a_2, \ldots, \mathrm{a_{N-1}}, \mathrm{s_N})$$

that occurs with **path probability** $\mathrm{p}(\omega)$

Evaluating Policies

Policies are often compared on the basis of the mean reward process

$$W(\omega) = \sum_{t=1}^{N-1} r_t(s_t, a_t) + r_N(s_N)$$

$$E_{\omega}^{\pi}[W(\omega)] = \sum_{\omega=1}^{|\Omega|} p(\omega) W(\omega)$$

Alternative policies can be compared on the basis of this criteria to ascertain which policy is better.

Question: What other ways can policies be compared?

General Formulation of Stochastic DPs

Evaluation Problem: For policy π what is the expected discounted reward?

State Action

$$u_N^{\pi}(s) = E_s^{\pi}\left[\sum_{t=1}^{N-1} \lambda^{t-1} r_t(X_t, Y_t) + r_N(X_N)\right]$$

Total discounted expected future rewards over all N stages given policy π is applied when system starts in state s.

Optimization Problem: An optimal policy π^* satisfies

$$u_N^{\pi^*}(s) \geq u_N^{\pi}(s), for\ all\ s \in S\ and\ for\ all\ polices\ \pi \in \Pi$$

To find π^* solve the following problem:

$$u_N^{\pi^*}(s) = \max_{\pi \in \Pi} u_N^{\pi}(s)$$

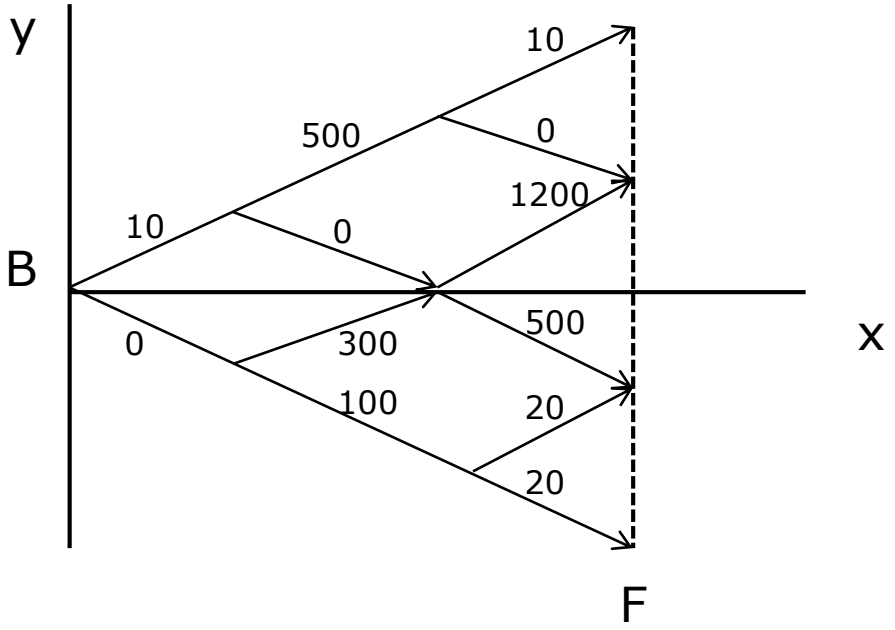
Stochastic Shortest Path

Consider the following stochastic shortest path problem

Assume the state transition outcomes depend probabilitically on the choice of action:

$$\Pr(u|a=u) \equiv p^u \text{ and } \Pr(d|a=d) \equiv p^d$$

where u is the decision to move up to the right and d is the decision to move down to the right.



Stochastic Shortest Path MDP

We want to know the optimal solution to this problem, but let's start with the easier question: what is the expected path length for a particular policy?

Policy Example:

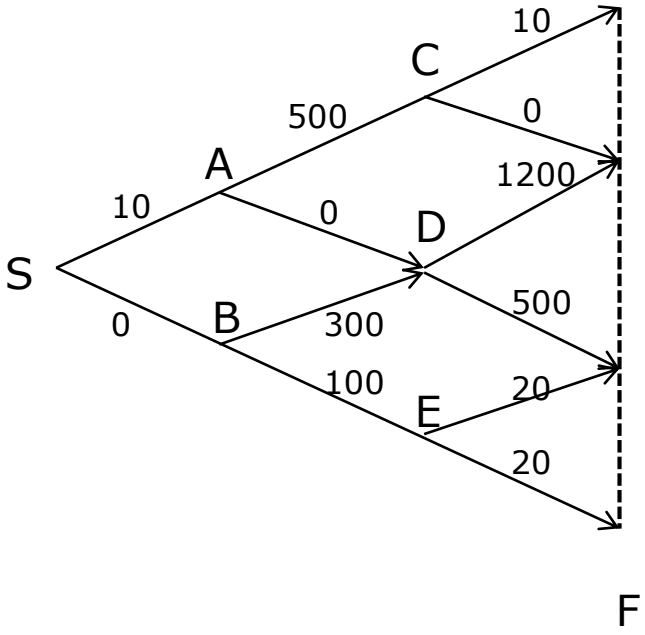
Consider the following policy:

1) At vertex S choose the action "up"
2) At all future states choose the action "up" if you have ever gone up in the past; otherwise choose the action "down" (remember choosing actions "up" or "down" do not guarantee you will go up in this **stochastic problem**)

What is the expected path length for this policy?

Example: Stochastic Shortest Path

- Problem data: Assume outcomes of actions have the following probabilities: $\Pr(u|a=u) = 0.8$ and $\Pr(d|a=d) = 0.7$
- Assume the following graph in which you start at node S and the goal is to reach the "finish line" F



- What is the expected distance travelled under policy π ?

Assumptions

For now we will make the following assumptions:

- The set of actions, A, and states, S, are finite
- The rewards, r(s,a), are bounded, i.e., $r(s,a) \leq M$
- The decision maker's goal can be represented by linear additive rewards

Policy Evaluation Algorithm

Definition: expected **value to go**, at stage t , given history h_t, and policy π:

$$v_t^{\pi}(h_t) = E_{h_t}^{\pi}\left[\sum_{n=t}^{N-1} \lambda^{n-1} r_n(X_n, Y_n) + r_N(X_N)\right]$$

Algorithm (Policy Evaluation):

1. Set $\mathrm{t} = \mathrm{N}, v_N^{\pi}(h_N) = r_N(s_N), for\ all\ h_N$
2. $If\ t = 1\ stop, otherwise\ go\ to\ step\ 3$
3. $Substitute\ t-1\ for\ t\ and\ compute\ v_t^{\pi}(h_t), for\ all\ h_t, as$:

$$v_t^{\pi}(h_t) = \mathrm{r_t}\left(\mathrm{s_t}, \mathrm{d_t}(\mathrm{h_t})\right) + \lambda \sum_{j \in S} p_t(j|s_t, d_t(h_t)) v_{t+1}^{\pi}(h_{t+1})$$

$return\ to\ step\ 2$

Stage t "decision rule" defined by policy π

Policy Evaluation Algorithm

Theorem (≈4.2.1 Puterman): For $\pi \in \Pi^{HD}$ at each stage t the policy evaluation algorithm generates $\mathrm{v}_{\mathrm{t}}^{\pi}(\mathrm{h}_{\mathrm{t}})$ for all h_t and $v_1^{\pi}(s) = u_N^{\pi}(s)$ for all $s \in S$.

Proof: By induction (completed in class)

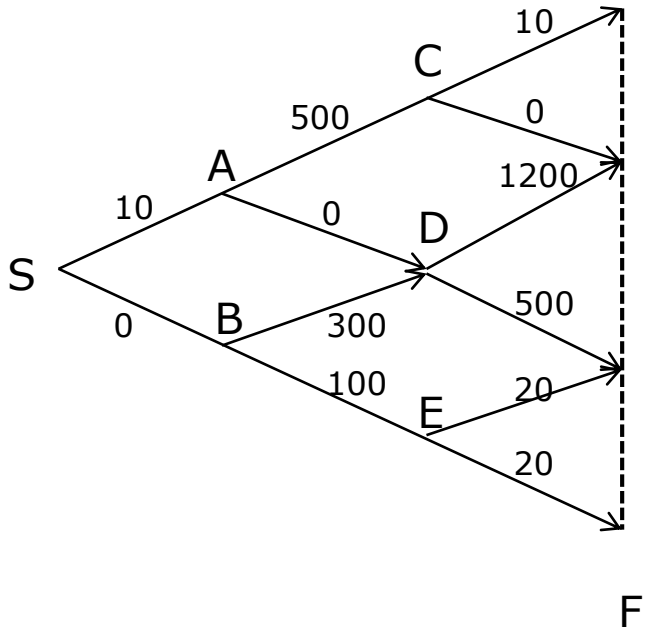
Note:

(1) Linear additive rewards are assumed in the proof

(2) The proof is for HD but is easily extended to HR

Example

- $\Pr(u|a = u) = 0.8$ and $\Pr(d|a = d) = 0.7$
- Policy π: At vertex S go up. At all future *states* go up if you have ever gone up in the past; otherwise go down.
- What is $v_1^{\pi}(S)$ According to the HD policy evaluation algorithm?



Complexity of Policy Evaluation

Consider the effort in evaluating policies:

- For policies of type HD or HR, if there are $|S|$ states and $|A|$ actions then at decision epoch t there are $|S|^t|A|^t$ histories

- For policies of type MD or MR each decision epoch requires evaluation of only $|S|$ value functions

Summary of Policy Evaluation

- The policy evaluation algorithm provides a method for evaluating policies

- For Markovian policies it is efficient in the sense that it requires a number of value function evaluations that is:

  - Linear in the state space
  - Linear in the number of stages

Next Time: Optimization of Stochastic DPs

For policy π the expected discounted reward is:

$$u_N^{\pi}(s) = E_s^{\pi}\left[\sum_{t=1}^{N-1} \lambda^{t-1} r_t(X_t, Y_t) + r_N(X_N)\right]$$

Optimization Problem: An optimal policy π^* satisfies

$$u_N^{\pi^*}(s) \geq u_N^{\pi}(s), for\ all\ s \in S\ and\ for\ all\ polices\ \pi \in \Pi$$

To find π^* solve the following problem:

$$u_N^{\pi^*}(s) = \max_{\pi \in \Pi} u_N^{\pi}(s)$$

Next Class

Next time we will cover **optimality equations** for MDPs

Read Sections 4.1 – 4.3 of Puterman in advance