CS 520: Introduction to Artificial Intelligence

Prof. Louis Steinberg

Lecture 19:

Sequential Decision Problems

Machine Learning

Acting under uncertainty

- **Probability: methods for representing and reasoning about uncertainty of knowledge**
- **Utility theory: a way to represent and reason about the** *desirability* **of different outcomes**
- **Decision theory: A way to combine probability and utility to decide what to do.**

Review:Joint Probability

- **The state of a single situation / domain may be represented by a *set* of random variables.**
- **Joint probability distribution (or just “the joint”): probability distribution over all possible sets of values for the state variables.**

Review: Building a Belief Net

1. Choose a set of variables.
2. Choose an ordering of the variables.
3 While (Some variables are left)
   a. Choose a variable, ``Xi''.
   b. Add a node for Xi to the network.
   c. Set Parents(i) to be the minimal set of nodes already in the network satisfying the conditional independence property.
   d. Define the conditional probability table for node Xi.

Review: Decision Theory

- **Basic concepts:**
  - **Lottery**

    [.4, pizza; .6 ice cream]

    Represents partial control over future states of the world
  - **Preferences**
    - **Relationship on partial states of the world**
  - **Rules of rational preferences**
    - **E.g. transitivity**
  - **Utility function U:partial-state =>**

    U(A)>U(B) <=> A preferred to B

Acting under uncertainty

- **Probability: methods for representing and reasoning about uncertainty of knowledge**
- **Utility theory: a way to represent and reason about the** *desirability* **of different outcomes**
- **Decision theory: A way to combine probability and utility to decide what to do.**

Representing an Agent's Preferences

- **Let A and B be states:**

$A \succ B$ **Agent prefers A to B.**

$A \sim B$ **Agent is indifferent between A and B.**

Representing Lotteries

- **If A and B are states, then [p,A ; 1-p,B] is the lottery with two possible outcomes.**
  - **State A is the outcome with probability p.**
  - **State B is the outcome with probability 1-p.**

Generalize to N possible outcomes.

Generalize to lotteries of lotteries.

Constraints on Rational Preferences

- **Orderable:**
  - (A < B) v (B < A) v (A = B)
- **Transitive:**
  - (A < B) ^ (B < C) => (A < C)
- **Continuous:**
  - A < B < C => (\exists p) [p,A;1-p,C] = B
- **Substitutable:**
  - A = B => [p,A;1-p,C] = [p,B;1-p,C]
- **Monotonic:**
  - A < B => (p > q <=> [p,A;1-p,B] < [q,A;1-q,B])
- **Decomposable:**
  - [p,A;1-p,[q,B;1-q,C]] = [p,A;(1-p)q,B;(1-p)(1-q),C]

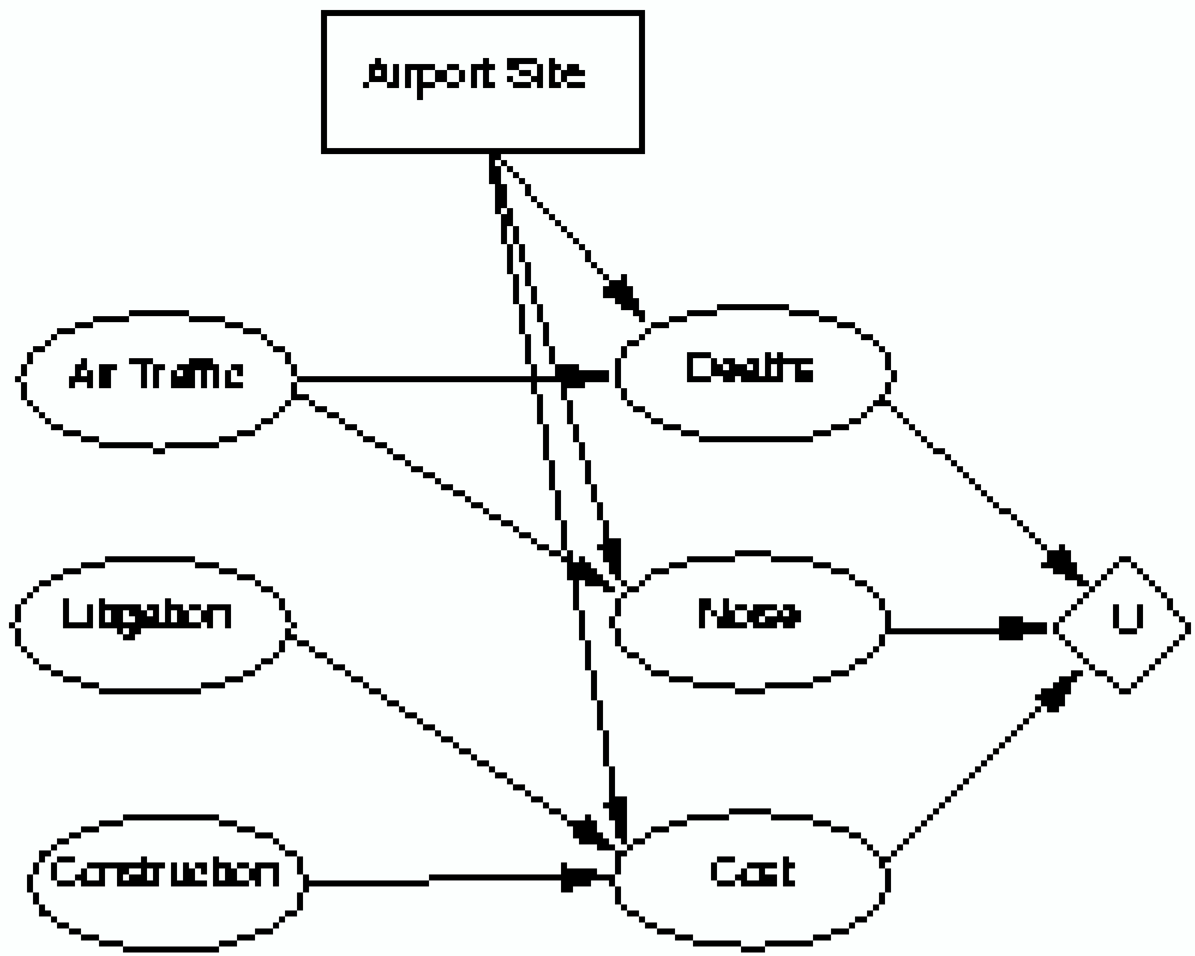
Representing Preferences by Utility Functions

- If an agent's preferences satisfy the constraints on rationalpreferences, then his preferences can be represented by a
- utility function U:States => Reals.
  - U(A)>U(B) <=> A < B
  - U(A)=U(B) <=> A = B
- The utility of a lottery is the expected utility of the outcome

One-Shot Decision Networks

- **Decision Nodes: (Rectangle) Controllable variables.**
- **Chance Nodes: (Ellipse) Random variables.**
- **Utility Node: (Diamond) Special random variable.**

Decision Network for Airport Siting Problem



Evaluating Decision Networks

- Set evidence variables from observations.
- For each possible action do:
  - Set decision nodes for the action.
  - Calculate posteriori probability for each value of
  - each parent of the utility node.
  - Calculate expected utility of the action.
- Return an action with highest expected utility.

Sequential Decision Problems

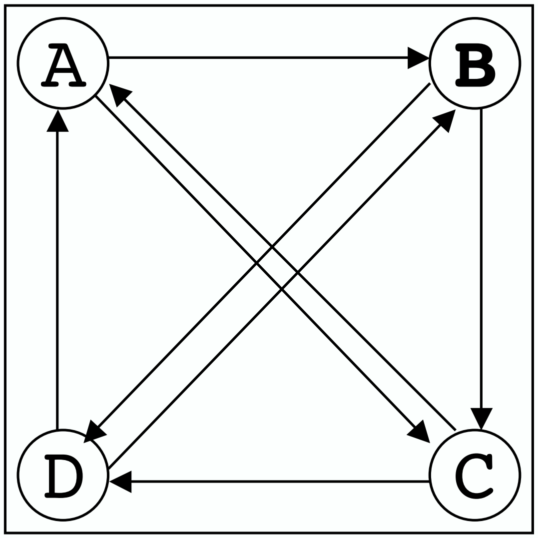
- Agent's utility depends on a sequence of decisions.
- Early decisions may effect choices available for later decisions.
- Early decisions may effect outcomes of later decisions.

Types of Environments

- **Deterministic: If agent knows the current state, he can predict the effect of each action.**
- **Stochastic: Even when agent knows the current state, he cannot predict the of each action.**
- **Accessible: The agent always knows or can determine the current state.**
- **Inaccessible: The agent has only partial information about the current state.**

Markov Decision Problem

- **Finding an optimal policy for an accessible, stochastic environment.**
- **A ``policy'' is a mapping from states to actions.**
- **Assume environment has ``Markov Property'', i.e., the transition probabilities don't depend on history.**



A	F	A	0
		B	.7
		C	.2
		D	.1
	X	A	0
		B	.1

• • •

- **States: A, B, C, D**
- **Operations:**
  - **F Forward, X across**
- **Transition matrix**

  OState X Operation X NState => P(choosing operation in ostate results in nstate)
- **Reward**

  U(state-sequence)

  Special case □ $U(s_i)$
- **Policy: state=>operation**

Finite and Infinte Horizons

- Finite Horizon: The agent makes a finite sequence of N decisions
- taking him through a series (S_0,…S_n) states:
  - Utility = U(S_1,…S_n)
- Infinite Horizon: The agent makes an infinite sequence of decisions
- taking him through a potentially infinite series S_0,…S_n,…states.
- Utility usually a discounted sum

  □$(1\text{-}d)^i U(s_i)$

Reward and Utility

- Reward for a state is immediate value
- Utility includes future value of states you get to from here
- Problem: optimal policy depends on utilities but utilities depend on policy
- Solution: relaxation (value iteration)
  - Estimate values
  - Loop until values don't change
    - Compute optimal policy for values
    - Recompute values

Policy Iteration

- Estimate policy
- Loop until policy does not change
  - Compute utilities given policy
  - Update optimal policy