RN, Chapter 16

Making Simple Decisions

Decision Theoretic Agents

- Introduction to Probability [Ch13]
- Belief networks [Ch14]
- Dynamic Belief Networks [Ch15]
- Single Decision [Ch16]
  - Decision Networks
  - Value of Information
  - Diagnosis
- Sequential Decisions [Ch17]
- Game Theory [Ch17.6 – 17.7]

Decision Analysis

- Def'n: ision ≡ irrevocable allocation of resources
  Shoul n
  - informat
  - alternatives
  - preferences

- NOW: quality of decision known at time of decision
- LATER: … determined by eventual outcome

- Simplest case:
  - **completely-specified**, **unique** “curre tate s_i
  - **complete knowledge of action outcome**
    … $\text{Result}(a, s)$ is single value, $\forall\ s, a$
  - **unambiguous, totally-ordered utility function**
    $U(s') \in \Re \quad \forall\ s' = \text{Result}(a, s)$

  $\Rightarrow$ optimal action
  $a^*(s) = \text{argmax}_a \{ U(\text{Result}(a, s)) \}$

- Issues: Current status is not known precisely
  Effects of actions not deterministic
  Utility function ill-defined (underspecified, $\in \Re^n$, …)

Utility Function

- Need to take actions

  ⇒ need preferences to compare different options
- Outcomes $\{\alpha_i\}$
  - Best outcome: $\alpha_\perp$
  - Worst outcome: $\alpha_\top$
- Lottery: $\alpha_i \sim [p : \alpha_\perp, 1- p : \alpha_\top]$
  - Agent indifferent between
  - taking α_i , vs
  - playing lottery with prize

$$\left\{\begin{array}{ll} \alpha_\perp & \text{w/prob } p \\ \alpha_\top & \text{w/prob } 1-p \end{array}\right.$$

- . . . with simple assumptions

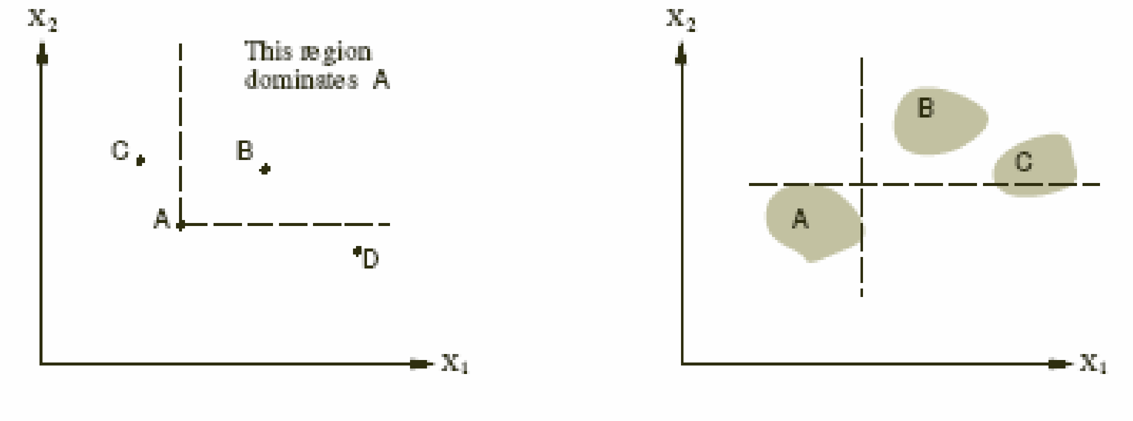
  (ordered, transitive, continuity, substitutability, monotonicity, …)

⇒ real-valued utility function $U(\alpha_i) \in \mathfrak{R}$

(in fact, $U(\alpha_i) \in [0, 1]$)

Dealing with Multiple Attributes

- Strict Domination:



(a) (b)

- If "mutually preferentially independent"
  - For all features...

$$\langle A1, B1, x, y \rangle \succsim \langle A2, B2, x, y \rangle$$
$$\Rightarrow \langle A1, B1, x', y' \rangle \succsim \langle A2, B2, x', y' \rangle$$

  - \+ deterministic

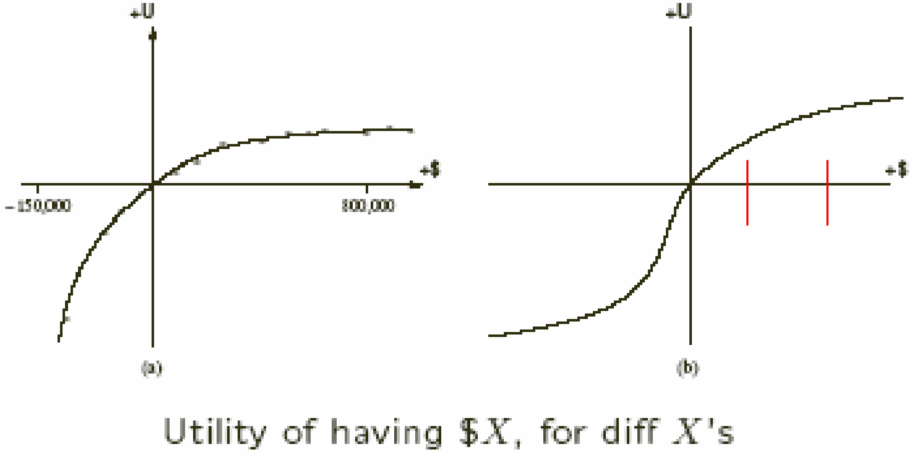
$U(S) = \sum_i U_i(S_i)$!

Utility ≈ Money ?

- Utility ≈ Money ...but not “linear”

$1M vs ½: $3M + ½: $0

$U(1M) >^{?} \frac{1}{2}\ U(0) + \frac{1}{2}\ U(3M)$



Utility of having $\$X$, for diff X's

- Typical pattern

  Risk Adverse (concave) for large X

  Risk Seeking (convex) for small (negative) X

  ... why insurance companies are rich!

- VERY SUBJECTIVE

  depends on person, resources, time, ...

Utility-Based Agents

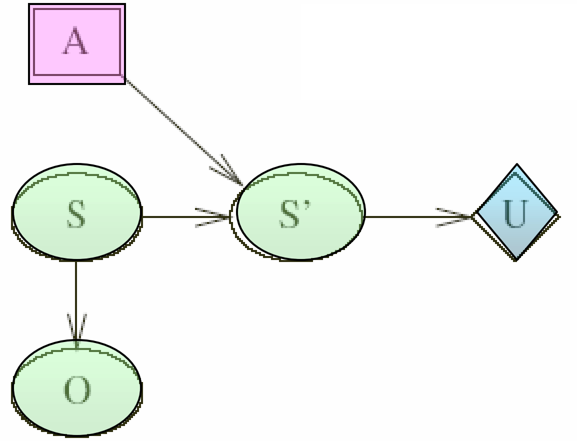
- MEU Principle:
  Agent should act to maximize expected utility
- Choose action that maximizes
  expected utility of state after A,
  given prior observations O:

$$\boxed{A^{*} = \text{argmax}_{A} \{ EU(A|O) \}}$$

$$
\begin{aligned}
&EU(A|O) = \\
&= \sum_{S'} P(S'|A,O)\, U(S') \\
&= \sum_{S'} \sum_{S} P(S \mid O)\, P(S' \mid S,A)\, U(S') \\
&= \sum_{S'} \sum_{S} [\alpha\, P(O \mid S)\, P(S)]\;\; P(S' \mid S,A)\, U(S')
\end{aligned}
$$

- Given simple assumptions, this is best possible action!
  (Average of utility, not of ~~utility^2~~, not ~~minimaxing…~~)
- Good decision, bad outcome.

Decision Network



- Chance Nodes: S, O, S'
  - Bayesian net ≡ decision diagram w/ only chance nodes
  - Specify: P(S), P(O | S), P(S' | S, A)
  - Here: S ≡ Current State   O ≡ Observation
    S' ≡ Resulting State
- Decision Nodes: A
  - represents decision/action to make.
  - Specify: set of possible actions a ∊ Dom(A)
- Utility Node(s): U
  - represents utility of each value-set of its parent chance variables
  - Specify: set of U(s') for each s' ∊ Dom(S')

Perform a Medical Treatment?

- $EU(T = 1) = \sum_r P(R = r \mid T = 1)\ U(R = r)$

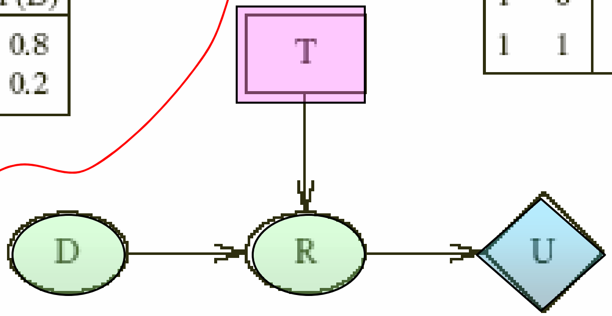
  $EU(T = 0) = \sum_r P(R = r \mid T = 0)\ U(R = r)$

- $P(R = 1 \mid T = 1) = \sum_d P(R = 1, D = d \mid T = 1)$

  $= \sum_d P(R = 1 \mid D = d, T = 1)\ P(D = d)$

  $= P(R = 1 \mid D = 0, T = 1)\ P(D = 0) + P(R = 1 \mid D = 1, T = 1)\ P(D = 1)$
  $= (0.001 \times 0.8) + (0.01 \times 00.2) = 0.0028$

- $P(R = 0 \mid T = 1) = 1 – P(R = 1 \mid T = 1) = 0.9972$
- Similarly:
  - $P(R = 1 \mid T = 0) = 0.1908$
  - $P(R = 0 \mid T = 0) = 0.8092$



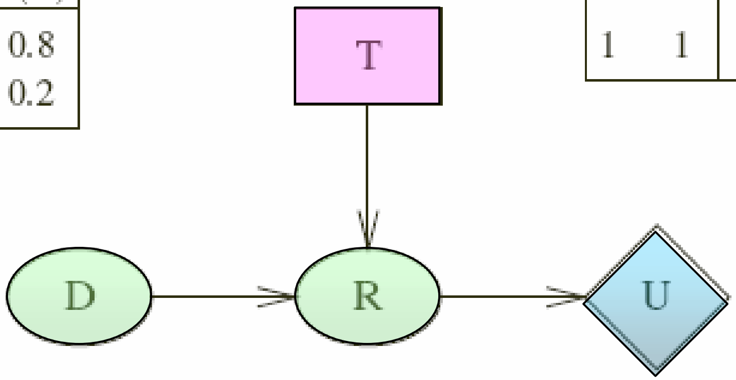
D	P(D)
0	0.8
1	0.2

D	T	P(R|D,T)
0	0	.001
0	1	.001
1	0	.950
1	1	.010

R	U(R)
0	0
1	-1000

Medical Treatment (con't)

D	P(D)
0	0.8
1	0.2



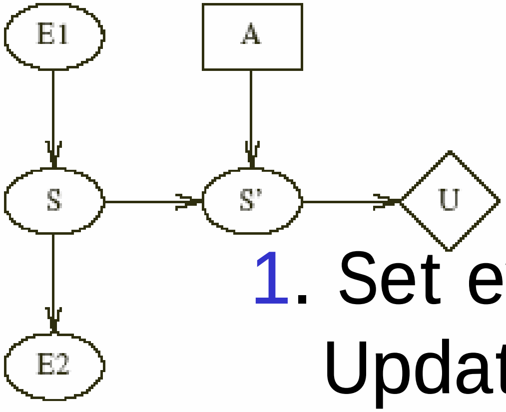
D	T	P(R|D,T)
0	0	.001
0	1	.001
1	0	.950
1	1	.010

R	U(R)
0	0
1	-1000

	$P(R\|T)$		$U(R)$		
T	0	1	0	1	$EU(T)$
0	.8092	.1908	0	−1000	−190.8
1	.9972	.0028	0	−1000	−2.8

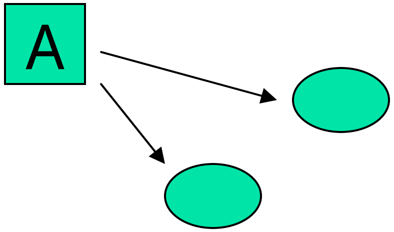
⇐ chosen action

Evaluating a Decision Network



1. Set evidence variables E_1, E_2
   Update distribution over current state S
2. For each possible action a of decision node A
   (a) Set decision node A to a
   (b) For each parent { S' } of utility node U:
       Calculate posterior probability of S
       Here, just $P(S' \mid E_1, E_2, A = a)$
   (c) Calculate expected utility for action a:
       $EU(A \mid E_1, E_2) = \sum_{S'} P(S' \mid E_1, E_2, a) U(S')$
3. Choose action $a^* = \arg\max_a \{ EU(a \mid \ldots) \}$
   with highest expected utility

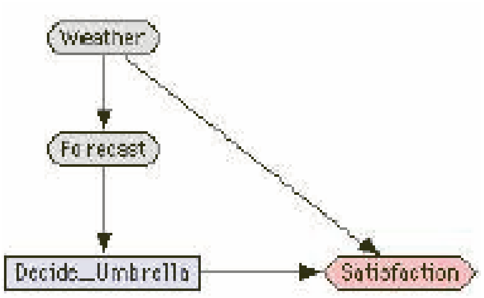
Domain of Decision Node A ...



If A's parents are . . .

1. None:

   Dom(A) = possible actions

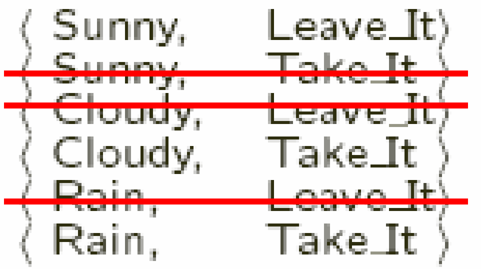


2. Chance node X:

   Dom(A) = Set of ⟨x, a⟩ pairs

   “Take action a if observe x”

   $x \in$ Dom(X) $a \in$ Actions



⟨ Sunny, Leave_It ⟩
~~⟨ Sunny, Take_It ⟩~~
~~⟨ Cloudy, Leave_It ⟩~~
⟨ Cloudy, Take_It ⟩
~~⟨ Rain, Leave_It ⟩~~
⟨ Rain, Take_It ⟩

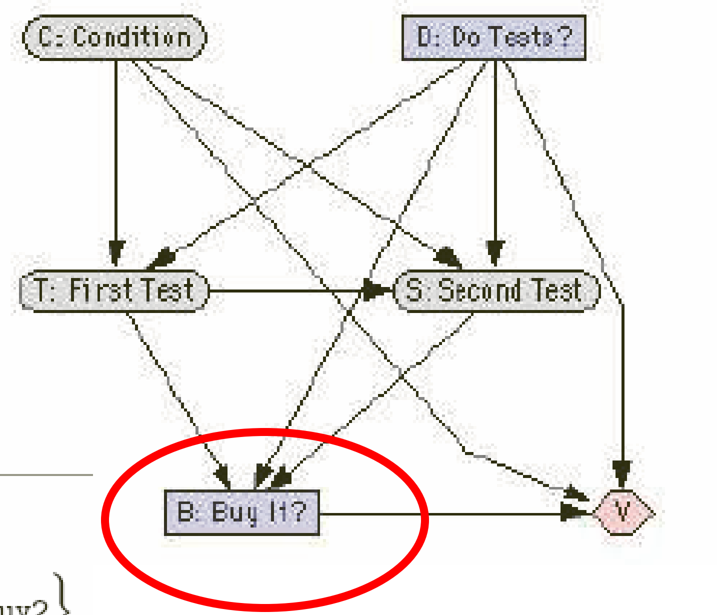
Domain of Decision Node A ...

If DecisionNode's parent is ...

3. another Decision Node D,

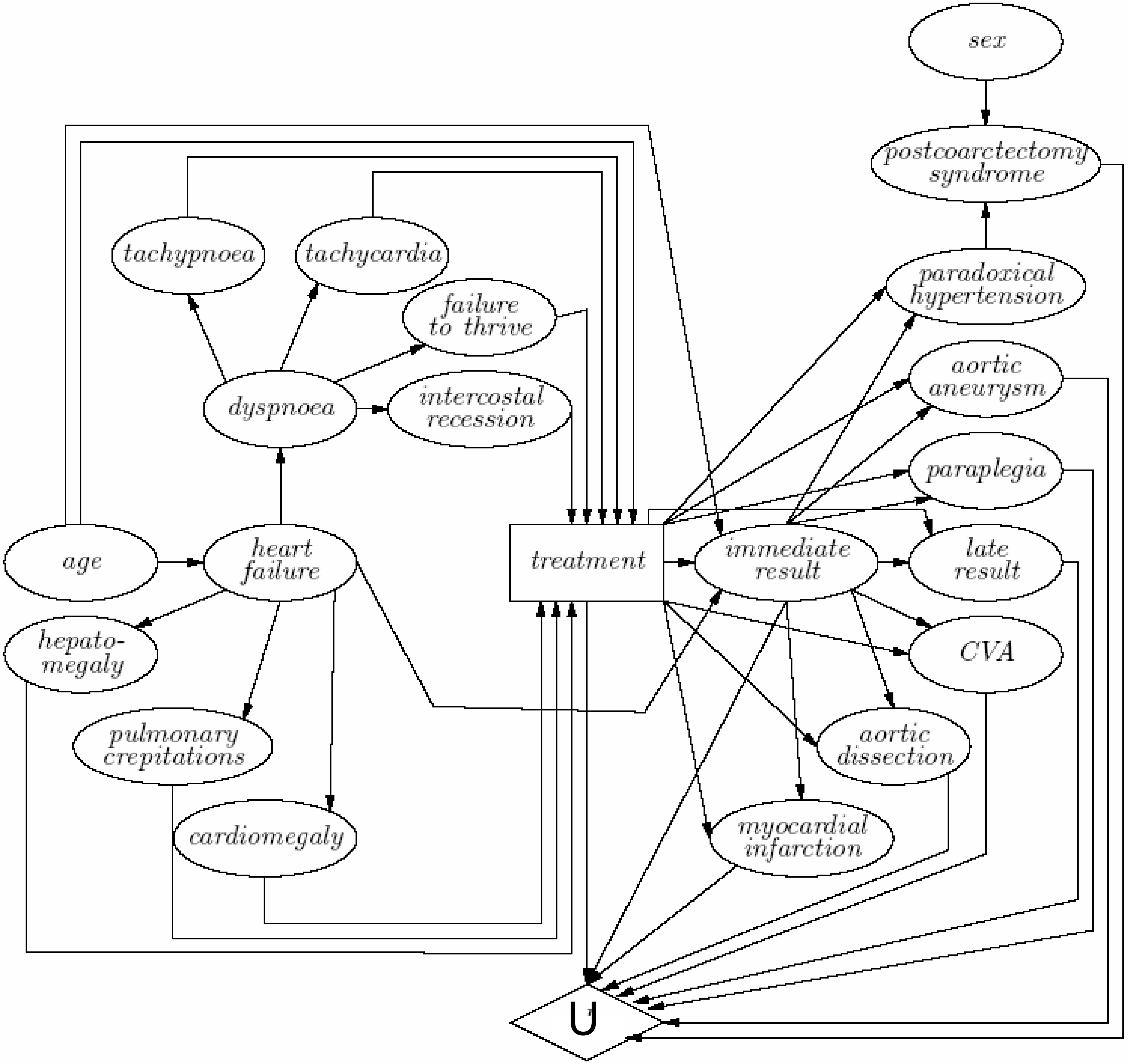
for whether to test B prior to action:

$$Dom(D) = \left\{ \begin{array}{ll} \bar{t}: & \text{Do NO tests} \\ t_1: & \text{test car \#1} \\ t_2: & \text{test car \#2} \end{array} \right\}$$



$Dom(B)$:

D	Options at B
$\bar{t}$	$\{\text{Buy1}, \text{Buy2}\}$
t_1	$\left\{ \begin{bmatrix} \text{Buy1 if } t_1=\text{pass} \\ \text{Buy2 if } t_1=\text{fail} \end{bmatrix}, \begin{bmatrix} \text{Buy2 if } t_1=\text{pass} \\ \text{Buy1 if } t_1=\text{fail} \end{bmatrix}, \text{Buy1}, \text{Buy2} \right\}$
t_2	$\left\{ \begin{bmatrix} \text{Buy1 if } t_2=\text{pass} \\ \text{Buy2 if } t_2=\text{fail} \end{bmatrix}, \begin{bmatrix} \text{Buy2 if } t_2=\text{pass} \\ \text{Buy1 if } t_2=\text{fail} \end{bmatrix}, \text{Buy1}, \text{Buy2} \right\}$


sex
postcoarctectomy syndrome
tachypnoea
tachycardia
paradoxical hypertension
failure to thrive
dyspnoea
intercostal recession
aortic aneurysm
paraplegia
age
heart failure
treatment
immediate result
late result
hepato-megaly
CVA
pulmonary crepitations
aortic dissection
cardiomegaly
myocardial infarction
U

Value of Information

- Probabilistic Reasoning can determine
  “Bayesian Optimal Action"
  even given only partial information.

  P(Cancer | Age, Gender)

  not

  P(Cancer | Age, Gender, T#1, T#2, ...)

- but... spse agent is capable of acquiring MORE info

  Eg running tests, issuing requests, . . .

- Which tests? How much is a test worth?

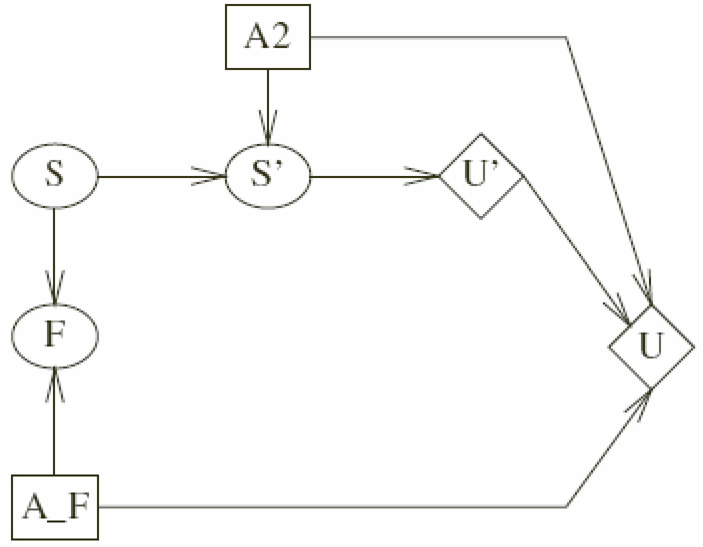
SLhT ≡ “Selected Lot has Treasure”

Example: Which Site?

- One of *n* lots has treasure, worth *$C*
- Can buy one lot, for C/n
  Expected return is
  $$P(+SLhT) \times U(+SLhT) + P(\neg SLhT) \times U(\neg SLhT)$$
  $$= \quad 1/n \quad [C - C/n] \ + (n-1)/n \ [-C/n] \quad = 0$$
- Spy knows whether lot#3 has treasure
  $L3$ ≡ Lot #3 has treasure"
  How much is this information worth?
- If know $L3$: *buy it!*
  Profit: $U(+L3) = C - C/n$
  If know $\neg L3$: *buy another lot!*
  Profit: $U(\neg L3) = C/(n-1) - C/n$
  Expected return, with this information:
  $$P(+L3)\ U(+L3) \quad + P(\neg L3)\ U(\neg L3) =$$
  $$1/n\ [n-1]/n\ C + (n-1)/n\ \ C\ [1/(n-1)n\ ] = C/n$$
- Value of info: difference between
  (return using info) – (return without info) = C/n

  (. . . ie, cost of block itself!)

Example#2: Perform Test



- A_F is “observation action"

  If performed:

  provides value of $F = f_i$

  costs reward = –10
- A2 is ordinary action

  If performed:

  costs reward = –100

Should we perform A_F?

- If we . . .
  - do not perform A_F,
    we will *NOT* know value of test (r.v.) F
    $EU(A_F = 0) = \max_{A2} [\sum_{S'} P(S' \mid S, A_2) U(S') – 100\ A_2]$
  - do perform A_F , we will know value of $F = f$

$$EU_{F=f}(A_F = 1) = \max_{A2} \Big[\sum_{S'} P(S' \mid S, A_2, F = f) U(S') – 100\ A_2 – 10\ A_F \Big]$$

$$EU(A_F = 1) = \sum_f P(F = f \mid S)\ EU_{F=f}(A_1 = 1)$$

- Value of Perfect Information wrt F
  ≡ difference between expected utility of
  optimal policy (with / without) F information

$$VPI(F) = EU(\underset{\substack{\uparrow \\ \text{Knowing F}}}{A_F} = 1) – EU(\underset{\substack{\uparrow \\ \neg\ \text{Knowing F}}}{A_F} = 0)$$

Value of Information (General)

- Initially, agent knows E but not (r.v.) F

$\Rightarrow$ (Prior) utility of action A is

$$EU(A|E) = \sum_S U(S)\ P(\ S = Result(A)\ |\ E, Do(A)\ )$$

Utility of best action: $UBA(E) = \max_A \{\ EU(A|E)\ \}$

- Knowing $F = f$, utility of action A is

$$EU(A|\ E, F = f_k) = \sum_S U(S)\ P(\ S = Result(A)\ |\ E, Do(A),\ \ F = f_k\ )$$

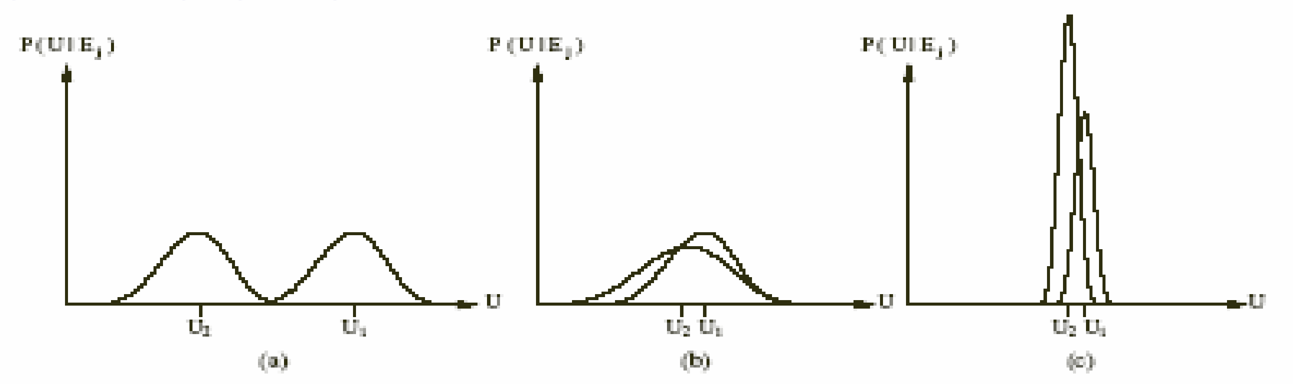
. . . $UBA(E, F = f_k) = \max_A \{\ EU(A\ |\ E, F = f_k)\ \}$

- But don't know that $F = f_k$. . .

$$VPI_E(F) = [\ \sum_k P(\ F = f_k\ |\ E\ )\ UBA(E,\ \ F = f_k)]\ \ – UBA(E)$$

- <u>V</u>alue of <u>P</u>erfect <u>I</u>nformation

Properties of VPI



(a): A1 almost certainly better than A2, no more info needed

(b): Choice unclear, get more info!

(c): Choice unclear, but who cares… (too little at stake)

- Information never hurts: $\forall$ E,U $VPI_E(U) \geq 0$
- Information doesn't add: $VPI_E(U, V) \neq VPI_E(U) + VPI_E(V)$
- Information is order-independent:

$$VPI_E(U; V) = VPI_E(U) + VPI_{E,U}(V) = VPI_E(V) + VPI_{E,V}(U)$$

The Diagnosis Problem

- **State**: assignment of values to each variable describing device
  SparkPlugs = Bad, Distributor = Ok, Starter = Ok,
  BatteryAge = new, EngineCranks = NoCrank, Starts = no, ...
- **Start state**: Unknown state where Starts = no
- **Actions:**
  - **Observe component**: can be applied to
    SparkPlugs, Distributor, …
  - **Repair component**: can be applied to
    SparkPlugs, Distributor, …
  - **Reward function**: Negative reward (cost) for each action
    Positive reward for getting into state where “Starts = yes"

- Note: “States” states of device
  ...not of troubleshooter!
  - Observation actions don't change state of device
  - Repair actions do change state of device.

Special Case Where Optimal Policy can be Computed

(A device with 20 components has 2^{20} states;
dynamic programming can't work on problems of that size)

- ∃ single problem-defining node.
  - E.g., engine-starts
- Device is malfunctioning in start state
- Single Fault Assumption:
  - Exactly one component is broken
- After each repair, problem-defining node is always observed
- Only two kinds of components:

  (a) observable and repairable
  (b) unobservable and repairable
- Only two kinds of actions:
  - For (a): observe, if broken, repair.
  - For (b): repair.
- Costs of actions are fixed and independent of order.
- No other observations are permitted

Computing Value of Policy

- Cost of observing component c_i: *Obs*(c_i)
  If c_i not observable, Obs(c_i) = cost of repairing c_i
- Cost of repairing component c_i : *Rep*(c_i)
  (includes cost of re-observing problem-defining variable)
  If c_i not observable, Rep(c_i) = 0)
- Probability that c_i is broken: p_{c_i}

- Task: Evaluate policy $\rho = \langle c_1, c_2, \ldots, c_n \rangle$
  where $U_\rho(s_0)$ = value of start state s_0 under policy ρ

$$U_\rho(s_0) \quad = \quad \sum_{i=1}^{n}\left[\left(1-\sum_{j=1}^{i-1} p_{c_j}\right) Obs(c_i) + p_{c_i} Rep(c_i)\right]$$

i^{th} term =

prob that first i –1 components are ok × cost of observing i^{th} component, c_i

plus

prob that c_i is broken × cost of repairing c_i

Computing Best Policy

- Compare policies:

$$\rho_1 = \langle\, c_1, \ldots, c_{k-1},\ \ c_k,\ c_{k+1}, \ldots, c_n \,\rangle$$

$$\rho_2 = \langle\, c_1, \ldots, c_{k-1},\ c_{k+1},\ \ c_k, \ldots, c_n \,\rangle$$

⇒ Prefer policy ρ_1 if $U\rho_1 (s_0) - U\rho_2(s_0) > 0$

... which is true if $\dfrac{p_{c_k}}{Obs(c_k)} > \dfrac{p_{c_{k+1}}}{Obs(c_{k+1})}$

- Ie, optimal policy processes components in descending order of $\dfrac{p_{c_i}}{Obs(c_i)}$

Extensions

- Multiple Faults

- Auxiliary Observation Actions

  Observing components that cannot be repaired or
  do not lie along a causal pathway to problem-defining node.

- Examples:
  - EngineCranks is observable,
    but not repairable
  - Lights is observable,
    but does not lie along causal path to Starts

Solving Multiple Faults Iteratively

1. Compute p_{c_i}, given current information.
2. Observe the (as yet unobserved) component with highest ratio $p_c/Obs(c_i)$
   - (If this c_i not observable, simply repair it.)
3. If c_i not faulty, *goto 1*.
4. If c_i faulty, then repair it.
   - If the device is working: *Terminate*
   - Otherwise go to step 1.

Two big differences:

- Doesn't assume 1st fault repaired fixes problem
- Recompute probabilities (and hence policy) after each repair.

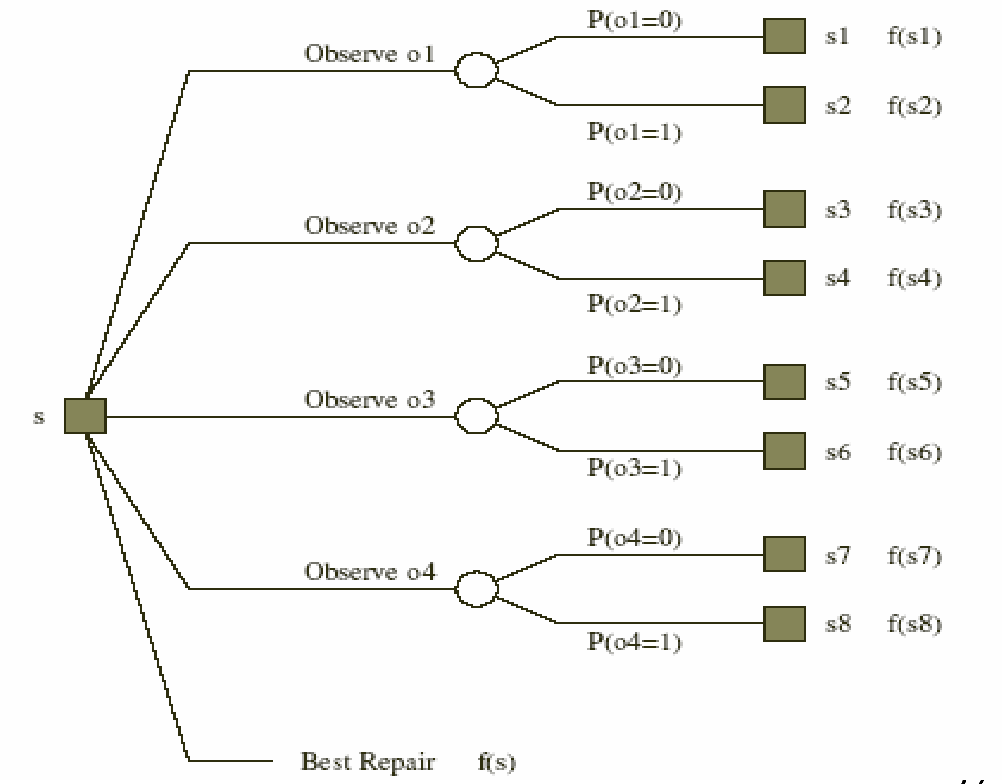
Note $\sum_i p_{c_i} \geq 1$

Auxiliary Observation Actions

- Perform one-step lookahead search, evaluate resulting states

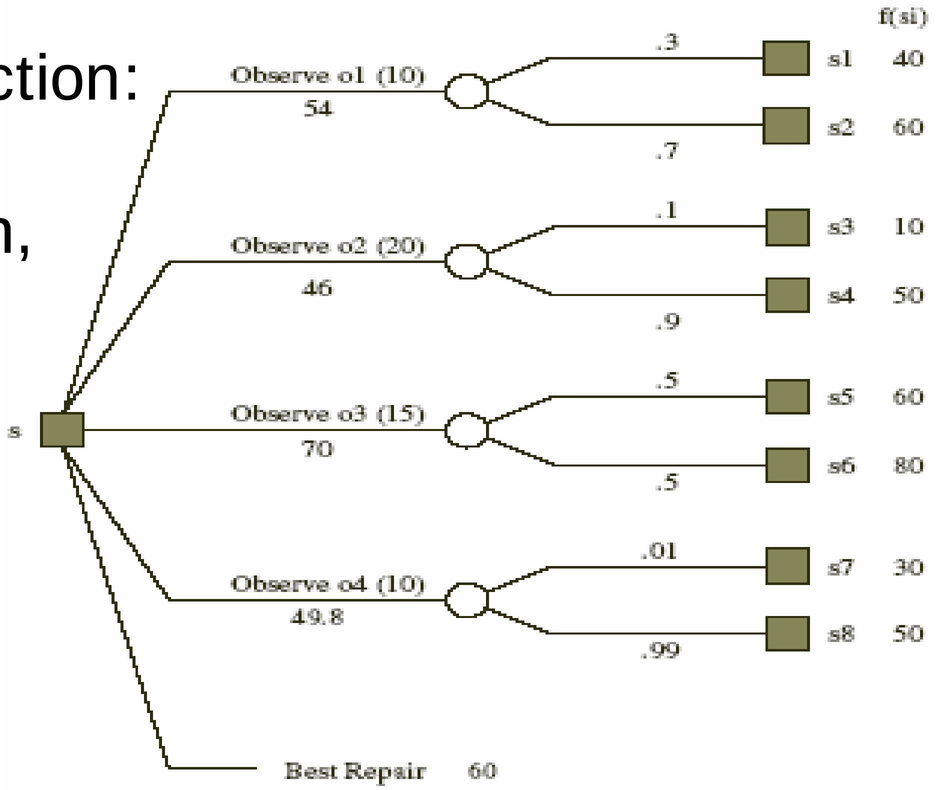
  assuming all subsequent actions are repairs

(⇒ we can easily compute values of those states)



Problem with Assumptions

- best greedy action:
  Observe o2
- but… misses optimal action:
  attempt repair,
  then make observation,
  then other repairs. . .



Summary of Terms

- Probabilistic Inference
  - Factorization: $P(X,Y) = P(X)\,P(Y \mid X)$
  - Marginalization: $P(X) = \sum_y P(X, Y = y)$
  - Conditionalization: $P(X \mid Y) = P(X, Y) / P(Y)$
- Decision Making
  - Utility: $U(S)$ = “happiness” at being in state S
  - Expected Utility: $EU(A \mid E)$ = expected “happiness" at taking action A, given evidence E
  - Maximize Expected Utility: $MEU(E) = \text{argmax}_A\, EU(A \mid E)$
  - Value Perfect Information: $VPI_E(F)$ for determining value of r.v. F, given evidence E

Summary

- Rational Action =
  Action that maximizes Expected Utility
- Depends on
  - (probabilistic) knowledge about current state
  - (stochastic) effects of actions
  - (subjective) utilities

  Modeled using Decision Nets
- ACTIONs can include
  - Actions that affect the world
  - SENSING actions, that provide information