CS 486 — Lecture 17: Decision Networks

1 The Weather DN

- Suppose we want to ask — do we take our umbrella or not?
- Depending on the weather, we may bring our umbrella.
- But we don't actually see the weather directly, we see the *forecast*.
- We can draw it like this:

```
(Weather) ->  <Utility>
    |             A
    V             |
(Forecast) -> [Umbrella]
```

- A policy specifies what the agent should do under all contingencies.
- For each decision variable, a policy specifies a value for the decision variable for each assignment of values to its parents.
- So, for our weather decision network, how many policies are there?
- Forecast has 3 possible values, and for each value of forecast, there are 2 possible decisions. This gives $2^3 = 8$ possible policies.
- We have two approaches to solving the network problem:
  - Brute force: solve every policy's expected utility, and choose the best one.
  - VEA.
- Obviously we prefer VEA since brute force can result in having to calculate a very large number of policies compared to the number of nodes!
- Consider the policy π_1:
  - Take the umbrella if the forecast is cloudy
  - Leave the umbrella at home otherwise

  What is the expected utility of this policy?
- In this case, using a brute force approach:

$$\begin{aligned} EU(\pi_1) &= P(norain) * P(sunny|norain) * u(norain, leaveit) \\ &+ P(norain) * P(cloudy|norain) * u(norain, takeit) \\ &+ P(norain) * P(rainy|norain) * (norain, leaveit) \\ &+ P(rain) * P(sunny|rain) * u(rain, leaveit) \\ &+ P(rain) * P(cloudy|rain) * u(rain, takeit) \\ &+ P(rain) * P(rainy|rain) * u(rain, leaveit) \\ &= 0.7 * 0.7 * 100 + \cdots + 0.l3 * 0.6 * 0 \\ &= 64.05 \end{aligned}$$

- We see that this could take too long. We'll now investigate VEA.

1.1 VEA

- The general algorithm is as follows:
  1. Remove all variables that are not ancestors of the utility node.
  2. Create factors.
  3. While there are decision nodes remaining:
     (a) Sum out each RV that is not a parent of a decision node
     (b) Find the optimal policy for the last decision
  4. Return the optimal policies.
  5. Sum out all the remaining RVs to get the expected utility of the optimal policy.
- Let's apply this to our weather example:
  1. Nothing to be done.
  2. Define three factors, f1(Weather), f2(Forecast, Weather), f3(Weather, Umbrella)
  3. Now:
     (a) Weather is not a parent of any decision node. Sum out Weather. For example, f1 * f2 * f3 = f4(Weather, Forecast, Umbrella), then sum out f4 to get f5(Forecast, Umbrella).
     (b) Now let's find the optimal policy for, say, Umbrella.