Final Exam Study Guide - CS440
By: Fauzan Amjad

Intro to Decision Theorcretic Agents and Probability Theory

1. What is the focus of decision theory? What are the basic steps of a decision theoretic agent?
    a. Decision Theory = Probability Theory + Utility Theory
        i. The main focus of decision theory is how we make decisions when have uncertainty.
        ii. Assigning appropriate utilities to end states is the objective of utility theory.
        iii. Think about airplane example where if being late is catastrophic, then coming to the airport as early as possible is appropriate.
    b. Basic steps
        i. Update a belief-state distribution based on actions/percepts (answering the question: what is the most probable state the world is currently in?)
        ii. Calculate outcome of actions given action descriptions and current belief state (compute the utility of each action)
        iii. Select action with the highest expected utility given probabilities of outcomes and utility information
        iv. Return action
2. You might be asked to provide any of the rules of probability theory discussed in the class:
    a. Kolmogorov's axioms,
        i. 0 <= P(a) <= 1, ∀a
        ii. P(true) = 1, P(false) = 0
        iii. P(a ∨ b) = P(a) + P(b) − P(a ∧ b)
    b. Product rule - with and w/o normalization,
        i. P(a ∧ b) = P(a | b)P(b)
    c. Bayes rule - with and w/o normalization,

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)}$$

        i.
        ii. Normalization Factor: < P(b|a), P(¬b|a) >=< αP(a|b)P(b), αP(a|¬b)P(¬b) >
            1. Basically, P(a) can be ignored if we want to find P(¬b|a)
    d. marginalization and conditioning,
        i. Marginalization Rule
            1. When we have available joint probabilities (P(x, y) = P(X ∧ y)), we can calculate a particular prior probability by applying marginalization:

$$P(x) = \sum_{y \text{ value of } Y} P(x, y)$$

            2.

ii. Conditioning Rule

1. Marginalization Rule + Product Rule

2. $$P(x|z) = \sum_{y \text{ value of } Y} P(x|y,z)P(y|z)$$

e. independence and conditional independence properties.

i. Independence

1. P(a|b) = P(a)
2. P(b|a) = P(b)
3. P(a ^ b) = P(a)P(b)

ii. Conditional Independence Properties

1. P(x, y|z) = P(x|z)P(y|z)
2. P(x|y, z) = P(x|z)
3. P(y|x, z) = P(y|z)

3. You might be asked to compare the value of P(a|b) and P(¬a|b) in a way that minimizes the number of probabilities that we have to be aware of.

a. 3

b. P(B | a), P(a), P(B| !a),

4. Assume two random binary variables α, β. What is the minimum number of distinct probabilities that you need to know to compare P(α|β) and P(¬α|β) using the Bayes rule? [Clarification of "distinct": if you know P(γ) then you also know P(¬γ) = 1 − P(γ).]

a. 3

b. P(B | a), P(a), P(B| !a),

Bayesian Networks: Properties and Exact Inference

1. What is the definition of a Bayesian network?

a. The Bayesian network formalism was invented to allow efficient representation of, and rigorous reasoning with, uncertain knowledge.

b. Bayesian networks is a graphical way to represent dependencies between variables and hopefully allow us to detect independencies.

c. It is a graph where

i. the nodes correspond to the random variables of a problem,

ii. directed links connect pairs of nodes so that:

1. X → Y (X is a parent of Y)

iii. each Xi has a conditional probability distribution. The conditional distribution expresses the effects of the parents to the node:

1. P(Xi|P arents(Xi)),

iv. the graph has no directed cycles, hence it is a directed, acyclic graph or a DAG

2. Why can we answer any query in a probabilistic domain given the Bayesian network that involves all the variables in this domain and their independence properties?

a. A Bayesian network can answer any probabilistic query that can be answered by a joint distribution.

b. P(X1, . . . , Xn) = $\prod_{i=1}^{n} P(X_i|X_{i-1}, \dots, X_1)$

   c. If the Bayesian network correctly represents the independence properties of the problem then the only nodes that influence each random variable Xi are the parent random variables Parents(Xi) in the Bayesian network. Which means the following is true:
      i. P(Xi|Xi−1, . . . , X1) = P(Xi|Parents(Xi))
      ii. Parents(Xi) ⊂ {Xi−1, . . . , X1}
   d. The joint distribution can be expressed as a product of the conditional probabilities stored on the Bayesian network
      i. $$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|Parents(X_i))$$
   e. Since the joint distribution can be used to answer every probabilistic query involving its random variables, then so does the information available on the Bayesian network of those variables.

3. How can we compute the joint probability distribution P(X1, . . . , Xn) from the conditional probabilities P(X |Prents(X)) stored on the Bayesian network that involves variables X1, . . . , Xn?
   a. If a bayesian network is a representation of the joint distribution, then it can solve any query, by summing all the relevant joint entries.
   b. A Bayesian network can answer any probabilistic query that can be answered by a joint distribution.
   c. P(X1, . . . , Xn) = $\prod_{i=1}^{n} P(X_i|X_{i-1}, \ldots, X_1)$
   d. If the Bayesian network correctly represents the independence properties of the problem then the only nodes that influence each random variable Xi are the parent random variables Parents(Xi) in the Bayesian network. Which means the following is true:
      i. P(Xi|Xi−1, . . . , X1) = P(Xi|Parents(Xi))
      ii. Parents(Xi) ⊂ {Xi−1, . . . , X1}
   e. The joint distribution can be expressed as a product of the conditional probabilities stored on the Bayesian network
      i. $$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|Parents(X_i))$$
   f. Since the joint distribution can be used to answer every probabilistic query involving its random variables, then so does the information available on the Bayesian network of those variables.

4. You may be provided with a Bayesian network and asked to compute the joint probability distribution. You can also then be asked to compute conditional probabilities that involve the variables of the Bayesian network. The conditional probability might:
   a. Involve all the variables in the Bayesian network
   b. Involve a subset of the variables in the Bayesian network
   c. Look at a bunch of problems on YouTube

5. How can we compute exactly a conditional probability of the form P(A|B) (without a normalization factor) from full joint probability distributions of the form: P(A, B) and P(¬A, B).
    a. Via enumeration
6. How does exact inference by enumeration work in Bayesian networks? You can be given a Bayesian network (e.g., such as the Burglary-Alarm problem) and asked to compute a conditional probability by applying exact inference.
    a. One possible way to approach exact inference in Bayesian networks is to take advantage of the fact that the Bayesian network is equivalent to full joint probability distribution.
    b. ▶ Bayesian Network - Exact Inference Example (With Numbers, FULL Walk…
    c. ▶ 1. Bayesian Belief Network | BBN | Solved Numerical Example | Burglar …
7. What are the basic ideas behind Variable Elimination? How is it advantageous over Inference by enumeration? You might be provided a Bayesian network and asked to compute a conditional probability by applying Variable Elimination.
    a. Basic Ideas behind variable elimination
        i. do the calculations once and save the results for later.
        ii. Variable elimination evaluates probabilistic expressions by following a bottom-up approach along the enumeration tree. Intermediate results are stored and summations over each variable are done only for those portions of the expression that depend on the variable.
    b. Advantages over inference by enumeration
        i. It eliminates repeated calculations
        ii. Enumeration is so slow because it joins whole joint distribution before sum out the hidden variables
        iii. Variable elimination marginalizes earlier, so it's much faster than inference by enumeration.
    c. How to do variable elimination:
        i. ▶ Variable Elimination
        ii. ▶ Bayesian Networks: Inference using Variable Elimination

Approximate Inference in Bayesian networks

1. Which techniques do you know for approximate inference in Bayesian networks? What is the basic idea/methodology behind approximate methods for inference in Bayesian networks?
    a. Techniques for approximate inference in Bayesian networks
        i. Direct Sampling
        ii. Markov Chain Simulation
    b. Direct Sampling
        i. The basic idea is that when you have a probability it is possible to sample from it.
        ii. So given a Bayesian network and its conditional probabilities, it is possible to sample an "atomic event".
    c. Markov Chain Simulation
        i. In MCMC, we do not create each sample from scratch. Instead we change the previous sample.

        ii. To achieve that, we sample a value for one of the non-evidence variables Xi.
        iii. But since all the neighbors in the Bayesian network of this variable Xi already have assignments, we must take their values into accoount before sampling Xi.

2. How does direct sampling work? You might be provided with a Bayesian network and a series of “random” numbers and asked to sample an atomic event by employing direct sampling.
    a. The basic idea is that when you have a probability it is possible to sample from it. So given a Bayesian network and its conditional probabilities, it is possible to sample an “atomic event”. An atomic event is an assignment of values to the random variables.
3. What direct sampling algorithms do you know to compute conditional probabilities? What are the basic ideas behind them? How do they work? You might also be provided a network and “random” numbers and asked to imitate the operation of these algorithms?
    a. Rejection Sampling
        i. The idea in rejection sampling is that initially we will produce the samples with the same approach as we described above with direct sampling. Then we will reject those samples that do not match the evidence.
    b. Likelihood Weighting
        i. The idea in likelihood weighting is to fix the evidence variables E and sample only the remaining variables X and Y .
        ii. This time, each sample also stores a weight. The weight expresses the probability the sample could have been produced if we had not fixed the evidence variables.
        iii. Becomes problematic when there’s too many evidences
4. What is the Markov Blanket of a random variable in a Bayesian network? Why is it important?
    a. A node is conditionally independent of all other nodes in the network, given its parents, children, MARKOV BLANKET and children’s parents
    b. A Markov blank of a node is its parents, childrens, and the parents of its children, excluding itself.
    c. Important Property
        i. The sampling process settles into a “dynamic equilibrium”, where the fraction of time spent in each state is proportional to its posterior probability.

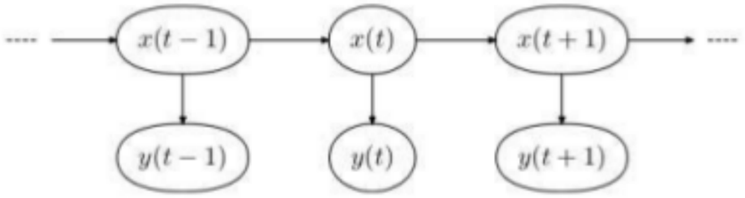
Dynamic Bayesian Networks: Temporal State Estimation

1. What assumptions do we typically employ in a Dynamic Bayesian network? Give a graphical representation of a network that involves state and evidence variables.
    a. Assumptions
        i. Stationary Process: The process with which the variables change over time does not change:
            1. P(Xt|Parents(Xt)) is the same $\forall$ t
        ii. Markov Assumption: The current state depends only on a finite history of previous states. If the current state depends only on the previous state then we have a first-order Markov process:

$$P(X_t|X_{0:t-1}) = P(X_t|X_{t-1}) \text{ transition model}$$

1. $$P(E_t|X_{0:t}) = P(E_t|X_t) \text{ observation model}$$

b. Graphical Representation

i. 

L

2. What problems can we answer by employing temporal models (Dynamic Bayesian Networks)? Provide the mathematical representation of these problems.
    a. Filtering: Compute the current belief state given all evidences so far
        i. $$P(X_t|E_{1:t}) = \alpha \cdot P(E_t|X_t) \cdot \sum_{X_{t-1}} P(X_t|X_{t-1}) \cdot P(X_{t-1}|E_{1:t-1})$$
    b. Prediction: Predict a future state based on all the evidences up to this point in time t
        i. $$P(X_{t+k}|E_{1:t}) = \sum_{X_{t+k-1}} P(X_{t+k}|X_{t+k-1}) \cdot P(X_{t+k-1}|E_{1:t})$$
    c. Smoothing: Given all the evidence up to time t, what is the most likely value of a previous state?
        i. $$P(X_k|E_{1:t}) = P(X_k|E_{1:k}, E_{k+1:t}) = \alpha * \overbrace{P(X_k|E_{1:k})}^{\text{filtering}} * P(E_{k+1:t}|X_k, E_{1:k})$$
    d. Most Likely Explanation: Given the observations up to this point, what is the sequence of states most likely to generate these observations?
        i. $$argmax_{X_1 \dots X_t}\{P(X_1 \dots X_t, X_{t+1}|E_{1:t+1})\} =$$ $$\alpha \cdot P(E_{t+1}|X_{t+1}) \cdot argmax_{X_1 \dots X_t}\{P(X_{t+1}|X_t) * argmax_{X_t}\{P(X_1 \dots X_t|E_{1:t})\}\}$$
3. What information/input should we have available in order to solve a problem with a Dynamic Bayesian Network?
    a. Prior (initial) probability: P(X0)
    b. The transition model: P(Xt|Xt−1)
    c. The observation model: P(Et|Xt)
4. Derive the filtering equation in Dynamic Bayesian Networks.

$$
\begin{aligned}
P(X_t \mid e_{1:t}) &= P(X_t \mid e_{1:t-1}, e_t) \\
&= \frac{P(e_t \mid X_t, e_{1:t-1}) P(X_t \mid e_{1:t-1})}{P(e_t|e_{1:t-1})} \text{ (Bayes' Rule)} \\
&= \frac{P(e_t \mid X_t) P(X_t \mid e_{1:t-1})}{P(e_t|e_{1:t-1})} \\
&= \frac{P(e_t \mid X_t) \sum_{x_{t-1}} P(X_t \mid x_{t-1}, e_{1:t-1}) P(x_{t-1} \mid e_{1:t-1})}{P(e_t|e_{1:t-1})} \\
&= \frac{P(e_t \mid X_t) \sum_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}{P(e_t|e_{1:t-1})} \text{ (Markov)} \\
&= \frac{P(e_t \mid X_t) \sum_{x_{t-1}} P(X_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}{\sum_{x_t} P(e_t \mid x_t) \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1} \mid e_{1:t-1})}
\end{aligned}
$$

   a.

5. What is the filtering equation? Explain its various elements. You can be provided with a small temporal state estimation problem and asked to apply filtering for one or two steps in order to update the belief distribution. You will be provided a transition, an observation model and an initial probability distribution. For example, consider the rain-umbrella example provided in the notes, or a robot moving in a grid world with obstacles

$$P(X_{t+1}|e_{1:t+1}) = \alpha \underbrace{P(e_{t+1}|X_{t+1})}_{\text{observation model}} \sum_{X_t} \underbrace{P(X_{t+1}|X_t)}_{\text{transition model}} \underbrace{P(X_t|e_{1:t})}_{\text{prior belief}}$$

   a. Filtering has constant time and space requirements per update.
   b. Look at rain umbrella example from notes

6. What is the backwards message used for solving the smoothing problem in temporal models?

$$P(X_k|E_{1:t}) = P(X_k|E_{1:k}, E_{k+1:t}) = \alpha * \overbrace{P(X_k|E_{1:k})}^{\text{filtering}} * \underline{P(E_{k+1:t}|X_k, E_{1:k})}$$

   a.
   b. It is used for implementing our observation, our recursive step, and our transition model into solving the smoothing problem in temporal models.

7. Which state estimate do you expect to be more accurate, a filtering estimate, a smoothing estimate or a predictive estimate?
   a. Smoothing because we are reinforcing our current beliefs.

8. Assume that for a specific system the predictive estimate is more accurate than the filtering estimate? What can you infer about the properties of this system for which we are trying to estimate its state?
   a. If predictive estimate is more accurate, we are trying to evaluate a state which we have no observation for. This is because the predictive estimate predicts from all the evidence up to this point.

9. What is the difference between the "most likely explanation" problem formulation in temporal models compared to filtering, smoothing and prediction?
   a. This problem is different from the previous ones in that the query is to return an entire sequence of states.

10. What is the advantage of the “most likely explanation” formulation over executing smoothing for all the states visited so far?
    a. Most likely explanation runs in linear time O(t) and smoothing runs in O(t^2)

Continuous Temporal State Estimation Problems: Kalman and Particle Filtering

1. What are the typical approaches for dealing with temporal state estimation problems that involve continuous variables?
    a. Kalman filter
    b. Particle filter
2. What is the advantage of using a Gaussian distribution to model a continuous temporal state estimation problem? How many numbers do you need to represent an n-dimensional state with a Gaussian distribution?
    a. Advantage
        i. The advantage of a Gaussian distribution is that it can be represented (for multi-dimensional problems) just by 1 its mean vector μ and its covariance matrix Σ. This means that if the state has n variables: {X1, X2, . . . , Xn}, then the mean is a vector of n values that stores the mean values for the variables:
        ii.

$$\mu = \begin{bmatrix} \mu_1 & \mu_2 & \dots \mu_n \end{bmatrix}^T = \begin{bmatrix} \bar{X_1} \\ \bar{X_2} \\ \dots \\ \bar{X_n} \end{bmatrix}$$

    b. 2n + nC2 numbers
3. What are the requirements so that the Kalman filter is the optimal solution to the Bayesian filtering problem?
    a. If the current distribution P(Xt|e1:t) is Gaussian and the transition model P(Xt+1|Xt) is linear Gaussian then the one-step prediction:
        i.

$$\underbrace{P(X_{t+1}|e_{1:t})}_{\text{is also Gaussian}} = \int_{X_t} P(X_{t+1}|X_t)P(X_t|e_{1:t})dt$$

    b. If the predicted distribution P(Xt+1|e1:t) is Gaussian and the observation model P(et+1|Xt+1) is linear Gaussian then the updated distribution:
        i.

$$\underbrace{P(X_{t+1}|e_{1:t+1})}_{\text{is also Gaussian}} = \alpha \cdot P(e_{t+1}|X_{t+1}) \cdot P(X_{t+1}|e_{1:t})$$

4. Under which circumstances does the Kalman filter fail? What approaches can be used to address these challenges?
    a. Circumstances where it fails
        i. The underlying processes are non-linear
        ii. The distribution is multi-modal.
    b. Approaches

i. When non-linear process
   1. Extended Kalman filter is an approximate version of Kalman filter

ii. When distribution is multi-modal
   1. “switching” Kalman filter or a Gaussian Sum filter.

iii. Particle filter can address both challenges at the expense of computational power

5. Describe the basic particle filtering algorithm. What is the property provided by a particle filter in terms of computing the correct probability distribution?
   a. Basic particle filtering algorithm
      i. A population of N samples is constructed by sampling from the prior distribution P(x0)
      ii. Then the following update cycle is repeated for each time step:
         1. Each sample if propagated forward by sampling the next state value xt+1 given the current value xt using the transition model P(xt+1|xt).
         2. Each sample is weighted by the likelihood it assigns to new evidence according to the observation model P(et+1|xt+1).
         3. The population is resampled to generate a new population of N samples. The probability that a sample is selected, is proportional to its weight. The new samples produced are assigned equal weights.
   b. Property Provided
      i.

$$
\begin{aligned}
\frac{N(x_{t+1}|e_{1:t+1})}{N} &= \alpha \cdot W(x_{t+1}|e_{1:t+1}) && \text{(Equation 3)} \\
&= \alpha \cdot P(e_{t+1}|x_{t+1}) \cdot N(x_{t+1}|e_{1:t}) && \text{(Equation 2)} \\
&= \alpha \cdot P(e_{t+1}|x_{t+1}) \cdot \textstyle\sum_{x_t} P(x_{t+1}|X_t) \cdot N(x_t|e_{1:t}) && \text{(Equation 1)} \\
&= \alpha \cdot N \cdot P(e_{t+1}|x_{t+1}) \cdot \textstyle\sum_{x_t} P(x_{t+1}|X_t) \cdot P(x_t|e_{1:t}) \\
&= \alpha' P(e_{t+1}|x_{t+1}) \cdot \textstyle\sum_{x_t} P(x_{t+1}|X_t) \cdot P(x_t|e_{1:t}) \\
&= P(x_{t+1}|e_{1:t+1})
\end{aligned}
$$

      ii. The property provided: P(x_{t+1} | $e_{1:t+1}$)

Utility Theory

1. What is the basic principle of utility theory?
   a. Agents pick states based on their utilities.
   b. A utility function assigns a single number to express the desirability of a state.
   c. U : S ⇒ R is used to denote the utility of a state, where S is the state space of a problem and R is the set of real numbers.
   d. These utilities are used in combination with probabilities of outcomes to get expected utilities for every action.
   e. Agents choose the actions that maximize their expected utility.
2. How can we compute the expected utility of an action A in a probabilistic setup (probabilistic transition model) given evidence variables?
   a. Agents choose the actions that maximize their expected utility.
   b. Consider a non-deterministic action A, which has possible outcome states Resulti(A). Index i spans over the different outcomes. Each outcome has a probability assigned to it by the agent before the action is performed:

i. $P(Result_i(A)|Do(A), E)$

c. E corresponds to the evidence variables. Now we want to maximize the expected utility. If U(Resulti(A)) is the utility of state (Resulti(A)), then the expected utility is:

i. $EU(A|E) = \sum_i P(Result_i(A)|Do(A), E)U(Result_i(A))$

3. Specify the rules of utility theory regarding preference of lottery outcomes.

a. A Lottery is a probability distribution over a set of outcomes. These outcomes can be called prizes of the lottery.

b. Rules

i. Orderability:

1. Given any two lotteries, a rational agent must either prefer one to the other or else rate the two as equally preferable.
2. That is, the agent cannot avoid deciding.
3. Exactly one of (A > B), (B > A), or (A ~ B) holds.

ii. Transitivity:

1. Given any three lotteries, if an agent prefers A to B and prefers B to C, then the agent must prefer A to C.
2. (A > B) ∧ (B > C) ⇒ (A > C) .

iii. Continuity:

1. If some lottery B is between A and C in preference, then there is some probability p for which the rational agent will be indifferent between getting B for sure and the lottery that yields A with probability p and C with probability 1 − p.
2. A > B > C ⇒ ∃ p [p, A; 1 − p, C] ~ B .

iv. Substitutability:

1. If an agent is indifferent between two lotteries A and B, then the agent is indifferent between two more complex lotteries that are the same except that B is substituted for A in one of them.
2. This holds regardless of the probabilities and the other outcome(s) in the lotteries.
3. A ~ B ⇒ [p, A; 1 − p, C] ~ [p, B; 1 − p, C] .

v. Monotonicity:

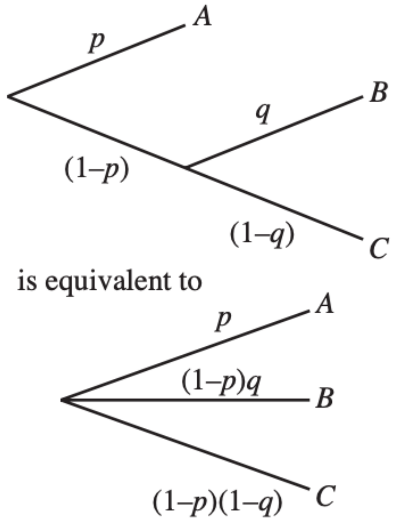
1. Suppose two lotteries have the same two possible outcomes, A and B.
2. If an agent prefers A to B, then the agent must prefer the lottery that has a higher probability for A (and vice versa).
3. $A \succ B \;\Rightarrow\; (p > q \;\Leftrightarrow\; [p, A;\; 1 - p, B] \succ [q, A;\; 1 - q, B])$.

vi. Decomposability:

1. Compound lotteries can be reduced to simpler ones using the laws of probability.

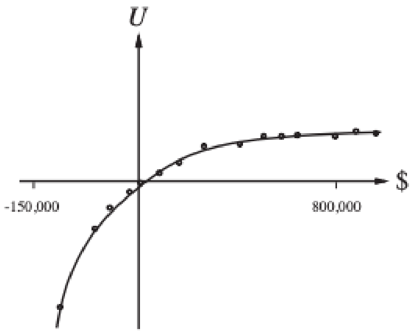
        2. This has been called the “no fun in gambling” rule because it says that two consecutive lotteries can be compressed into a single equivalent lottery, as shown in Figure 16.1(b).
        3. $[p, A;\ 1-p, [q, B;\ 1-q, C]] \sim [p, A;\ (1-p)q, B;\ (1-p)(1-q), C]$

4. What does the “no fun in gambling” rule of utility theory specify? Why does it have this name, what is its meaning?
    a. Compound lotteries can be reduced to simpler ones using the laws of probability.
    b. This has been called the “no fun in gambling” rule because it says that two consecutive lotteries can be compressed into a single equivalent lottery, as shown in Figure 16.1(b).3



    c.

5. What are the effects of risk aversion in the behavior of a utility function? Can you provide an example relating to the utility of money?
    a. Risk aversion can lead to a a sure payoff that is less than the expected monetary value of the gamble.
    b. Consider the case where you can either take $1,000,000 or gamble between winning $0 and $3,000,000 by flipping a coin. So our expected monetary value of the gamble looks like this:
        i. 0.5 · ($0) + 0.5 · ($3, 000, 000) = $1, 500, 000
    c. And the expected monetary value for the first case is $1,000,000. This does not necessarily mean that gambling is the better deal. Some people would rather take $1,000,000 if it is worth a lot to them, while people with billions of dollars may gamble because the $1,000,000 will probably not make a huge difference to them. Studies have shown that the utility of money is proportional to the logarithm of the amount, shown in the first graph below



(a)

   d.

6. How can we deal with multi-attribute utility functions?
   a. Multi-attribute problems are problems in which there is more than one attribute
      i. Example: If an airport needs to be built, then cost of land, distance from centers of population, noise levels, and safety issues are all attributes associated with this problem.
   b. **Strict dominance** is how we can deal with multi-attribute multi utility function.
      i. Suppose that an airport site S1 costs less, generates less noise, and is safer than an airport site S2. Since S2 is worse on all attributes, it does not even need to be considered. S1 has a strict dominance over S2 in this case.
      ii. Strict dominance can be useful in narrowing choices.

Decision Networks and Value of Information

1. How is a decision network different than a Bayesian network? Describe the purpose of the additional nodes.
   a. Decision networks are different than a Bayesian network in that it combines Bayesian networks with additional nodes for types of actions and utilities.
   b. Additional Nodes
      i. Chance Nodes(ovals):
         1. as in a typical Bayesian network they represent the possible attributes that effect the problem’s state.
      ii. Decision Nodes(rectangles):
         1. They correspond to the different actions that the agent can take to solve a problem.
      iii. Utility nodes(diamonds):
         1. They are used to show which attributes affect the utility
2. Describe the algorithm for computing the best action given a decision network.
   a. Set evidence variables for the current state
   b. For each possible value of the decision node
      i. Set decision node to that value
      ii. Calculate posterior probabilities for the state nodes that are parents to the utility node
      iii. Compute expected utility for this action
   c. Return action that maximizes the expected utility
3. How can we compute the value of information that we can acquire in order to make a decision?
   a. If we do not acquire the evidence Ej:

i. $$E(U)(a|E) = \max_A \sum_i U(Result_i(A)) * P(Result_i(A)|D_0(A), E)$$

ii. Where A is the action space for the agent

b. If we do acquire the evidence Ej

i. $$E(U)(a_{E_j}|E, E_j) = \max_A \sum_i U(Result_i(A)) * P(Result_i(A)|(A), E, E_j)$$

c. Value of Perfect Information:

i. $$VPI_E(E_j) = (\sum_k P(E_j = e_{jk}|E) * E(U)(a_{E_{jk}}|E, E_j = e_{jk})) - EU(a|E)$$

4. You might be given an example similar to the “oil company - seismologist example” in class and asked to compute the value of a piece of information.

a. Consider the following example: An oil company has n indistinguishable blocks of ocean drilling rights

i. Only one of them will contain oil worth $C

ii. The price of each block is $ C / n

b. A seismologist offers to survey block 3 and will definitely indicate whether the block has oil or not. How much should the company pay the seismologist?

i. In other words, what is the “value of information” that the seismologist offers?

c. There are two outcomes when using the seismologist:

i. $\exists$ oil in block 3, which has a probability of $\frac{1}{n}$
Then the best action is to buy block 3
Because we want profit (C) - cost ($\frac{C}{n}$), the total profit is $C - \frac{C}{n} = (n-1)\frac{C}{n}$

ii. Does not $\exists$ oil in block 3, which has a probability of $\frac{n-1}{n}$
Then the best action is to buy a block other than 3
Because expected profit is $\frac{C}{n-1}$ and since we choose among n-1 blocks, cost is $\frac{C}{n}$ of buying a block.
The total profit is therefore $\frac{C}{n-1} - \frac{C}{n} = \frac{C}{n(n-1)}$

d. The expected utility in the case that we use the seismologist is:

i. $$EU = \frac{1}{n} * \frac{(n-1)C}{n} + \frac{n-1}{n} * \frac{C}{n(n-1)} = \frac{C}{n}$$

e. If we do not use the seismologist:

i. $$EU = \text{ expected profit - cost } = \frac{C}{n} - \frac{C}{n} = 0$$

(Partially Observable) Markov Decision Processes

1. What do we need to define in order to formulate a Markov Decision Process?

a. Transition model: T(s, α, s’)

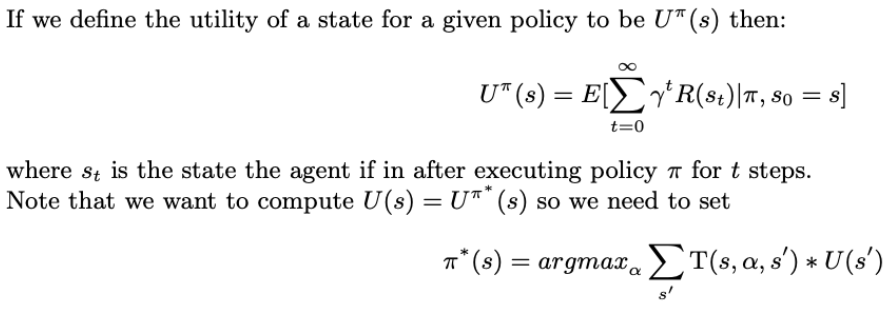
i. is the probability of going from state s to state s’ by applying the action α.

b. Rewards: R(s)

i. typically assign some positive value to the desired goal state and some negative value to a finish node that is undesirable

        ii. we assign a small negative value to each cell that the agent visits in hopes to entice the agent to find the goal quickly
            1. (-0.04 for every cell that is not the goal)
    c. Initial state: s0

2. What do we have to compute in order to solve a Markov Decision Process?
    a. Compute an optimal policy π * , a policy that yields the highest expected utility
    b. A policy tells the agent the best action to execute at each state so as to maximize expected utility.
3. How do we compute utilities of state sequences (i.e., paths)?
    a. There are two ways to compute the reward of a path.
    b. (1) Simply add up the rewards of the state along the path:
        i. $$U([s_0, s_1, ...]) = R(s_0) + R(s_1) + ...$$
        ii. Danger: Infinite path could mean each infinite rewards for each path
    c. (2) Use discounting
        i. U([s0, s1, ...]) = $\frac{R_{max}}{(1-\gamma)}$
4. How does value iteration work? What is the main idea behind the algorithm?
    a. Calculate the utility of each state and then use the state utilities to select an optimal action in each state.
    b. The utility of a state is the expected utility of the state sequences that might follow it.
    c. State sequences depend upon a policy, therefore we must define a random policy in order to begin applying the algorithm.
5. Derive the Bellman equation and describe the value iteration algorithm.
    a. Bellman Equation Derivation

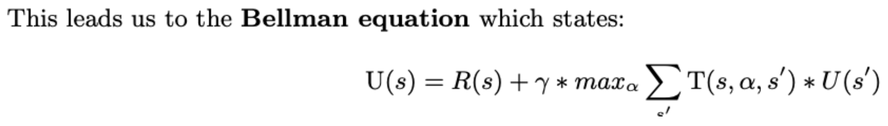


If we define the utility of a state for a given policy to be $U^{\pi}(s)$ then:

$$U^{\pi}(s) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t) | \pi, s_0 = s]$$

where s_t is the state the agent if in after executing policy π for t steps. Note that we want to compute $U(s) = U^{\pi^*}(s)$ so we need to set

$$\pi^*(s) = argmax_{\alpha} \sum_{s'} \mathrm{T}(s, \alpha, s') * U(s')$$



This leads us to the **Bellman equation** which states:

        i. $$\mathrm{U}(s) = R(s) + \gamma * max_{\alpha} \sum_{s'} \mathrm{T}(s, \alpha, s') * U(s')$$

    b. Value Iteration Algorithm
        i. Make an initial assignment.
        ii. Calculate the right hand side of the Bellman equation.
        iii. Plug the values into the Left hand side:
            1. $$U_{i+1}(s) = R(s) + \gamma * max \sum_{s'} T(s, \alpha, s') * U_i(s')$$

iv. Repeat until you reach equilibrium.

6. What are the properties of value iteration?

a. The algorithm always converges to an optimal solution.

b. At each iteration, you always know your error from the actual solution.

c. Unfortunately it might take a very long time to reach a solution.

7. How does policy iteration work? What is the main idea behind the algorithm? Describe the algorithm. What is the algorithm's advantage over value iteration?

a. How does it work?

i. Starts with randomly chosen πt at t = 0.

ii. Alternates between the policy evaluation and the policy improvement operations until convergence.

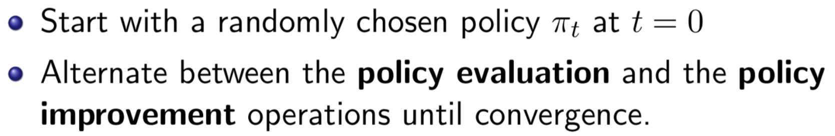
b. Main idea

i. The basic principle of policy iteration is the following:

ii. Given a policy πi, calculate Ui = U^(πi) , the utility of each state if πi were to be executed. Then given Ui, we can calculate a new policy πi+1 that maximizes expected utility using one-step look-ahead.

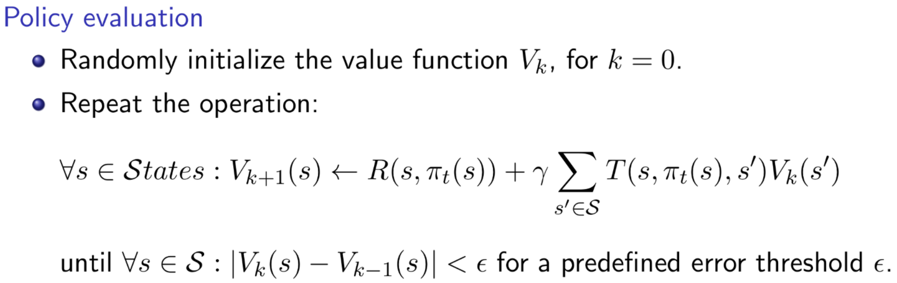
c. Describe algorithm

i.



- Start with a randomly chosen policy π_t at $t = 0$
- Alternate between the **policy evaluation** and the **policy improvement** operations until convergence.

ii.



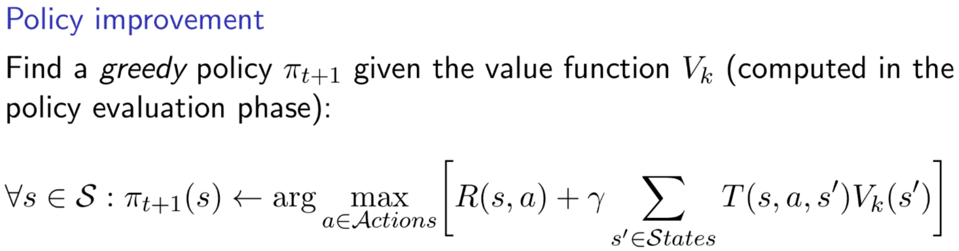
Policy evaluation

- Randomly initialize the value function V_k, for $k = 0$.
- Repeat the operation:

$$\forall s \in \mathcal{S}tates : V_{k+1}(s) \leftarrow R(s, \pi_t(s)) + \gamma \sum_{s' \in \mathcal{S}} T(s, \pi_t(s), s') V_k(s')$$

until $\forall s \in \mathcal{S} : |V_k(s) - V_{k-1}(s)| < \epsilon$ for a predefined error threshold ϵ.

iii.



Policy improvement

Find a *greedy* policy π_{t+1} given the value function V_k (computed in the policy evaluation phase):

$$\forall s \in \mathcal{S} : \pi_{t+1}(s) \leftarrow \arg \max_{a \in \mathcal{A}ctions} \left[R(s, a) + \gamma \sum_{s' \in \mathcal{S}tates} T(s, a, s') V_k(s') \right]$$

d. Advantage over value iteration

i. The important advantage of policy iteration is that once you assume the first policy, computing the utilities for the cells is easier than in value iteration.

ii. Instead of the operand max in the equations, we have the operand Σ, which implies that the equations are now linear in nature.

iii. Thus, we end up with n linear equations and n unknowns.

iv. Solving this linear system of equations has an O(n^3) complexity. Policy evaluation is often more efficient in small state spaces, however, for large state spaces O(n^3) may still be prohibitive.

8. You might be provided a small Markov Decision Process and asked to solve it by applying either policy iteration or value iteration.
    a. ▶ Policy and Value Iteration
    b. ▶ 7 POLICY ITERATION
9. What do we need to define in order to formulate a Partially Observable Markov Decision Process? What makes POMDPs a challenging problem?
    a. Define
        i. Initial probability distribution: b0
        ii. Transition model: T(s, a, s')
        iii. Rewards function: R(s)
        iv. Observation model: O(s, o)
    b. Challenging
        i. When the environment is only partially observable, the situation is, one might say, much less clear
        ii. The agent does not necessarily know which state it is in, so it cannot execute the action π(s) recommended for that state.
        iii. Furthermore, the utility of a state s and the optimal action in s depend not just on s, but also on how much the agent knows when it is in s.
        iv. For these reasons, partially observable MDPs are usually viewed as much more difficult than ordinary MDPs
10. Describe how to turn a POMDP into an MDP over a belief state space.
    a. Compute Transitionn model for new version of MDP
        i.

$$\begin{aligned} \tau(b,a,b') &= P(b'|a,b) \\ &= \textstyle\sum_o P(b'|o,a,b)\cdot P(o|a,b) \quad \text{(conditioning)} \end{aligned}$$

    b. The second probability is something that we have to deal explicitly. It can be computed as follows:
        i.

$$\begin{aligned} P(o|a,b) &= \textstyle\sum_{s'} P(o|a,s',b)\cdot P(s'|a,b) && \text{(conditioning)} \\ &= \textstyle\sum_{s'} O(s',o)P(s'|a,b) && \text{(rewriting observation model)} \\ &= \textstyle\sum_{s'} O(s',o)\sum_s T(s,a,s')b(s) && \text{(conditioning)} \end{aligned}$$

    c. Now that we have an expression of P(o|a, b) that depends on the state-level observation O(s' , o) and transition models T(s, a, s'), which are available to us as input, we can replace this expression to the computation of the belief-level transition model:
        i.

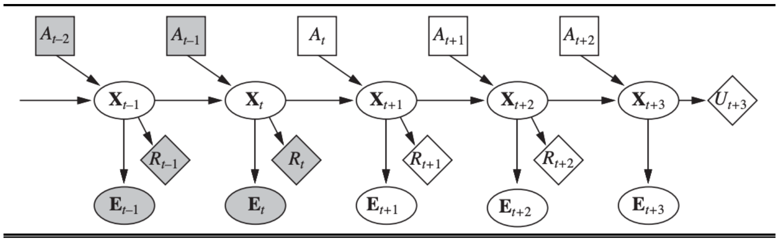
$$\tau(b,a,b') = \sum_o P(b'|o,a,b)\sum_{s'} O(s',o)\sum_s T(s,a,s')b(s)$$

    d. The last element that we have to define for this type of MDPs is how is the reward function computed over the belief states:
        i.

$$\rho(b) = \sum_s b(s)\cdot R(s)$$

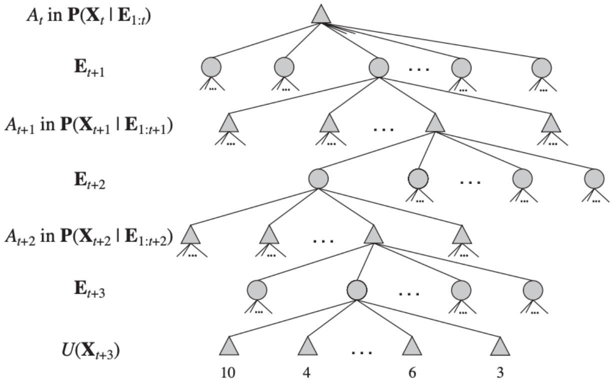
    e. Together, P(b’ | b, a) and ρ(b) define an observable MDP on the space of belief states.
    f. Furthermore, it can be shown that an optimal policy for this MDP, π*(b), is also an optimal policy for the original POMDP.
    g. In other words, solving a POMDP on a physical state space can be reduced to solving an MDP on the corresponding belief-state space

11. Describe a general version of a decision-theoretic agent that solves POMDPs by employing a look-ahead approach. Give the corresponding graphical representation and describe the algorithm for estimating the best action at each time step.
    a. General version of a decision-theoretic agent that solves POMDPs
        i. The transition and sensor models are represented by a dynamic Bayesian network (DBN), as described in Chapter 15.
        ii. The dynamic Bayesian network is extended with decision and utility nodes, as used in decision networks in Chapter 16. The resulting model is called a dynamic decision network, or DDN.
        iii. A filtering algorithm is used to incorporate each new percept and action and to update the belief state representation.
        iv. Decisions are made by projecting forward possible action sequences and choosing the best one.
    b. Graphical Representation
        i. Dynamic Decision Network



            1. **Figure 17.10** The generic structure of a dynamic decision network. Variables with known values are shaded. The current time is t and the agent must decide what to do—that is, choose a value for A_t. The network has been unrolled into the future for three steps and represents future rewards, as well as the utility of the state at the look-ahead horizon.

        ii. Look-ahead solution for DDN


A_t in $\mathbf{P}(\mathbf{X}_t \mid \mathbf{E}_{1:t})$
$\mathbf{E}_{t+1}$
A_{t+1} in $\mathbf{P}(\mathbf{X}_{t+1} \mid \mathbf{E}_{1:t+1})$
$\mathbf{E}_{t+2}$
A_{t+2} in $\mathbf{P}(\mathbf{X}_{t+2} \mid \mathbf{E}_{1:t+2})$
$\mathbf{E}_{t+3}$
$U(\mathbf{X}_{t+3})$
. . .
10
4
6
3

            1.

12. You might be provided a small POMDP like the tiger problem and asked to compute an appropriate policy
    a. Lecture 15 Partially Observable MDPs (POMDPs) -- CS287-FA19 Advanc…