

Probabilistic Reasoning

Notes

Nijat Mursali

22nd January 2022

Table of Contents

Table of Contents	1
Uncertainty	3
What is a qualification problem?	3
What is a rational decision?	3
What is the advantage of using probability over first-order logic (FOL)?	3
What is the difference between causal and diagnostic rules?	3
What does the probability of 0 and 1 mean?	4
Explain the decision-making process with MEU.	4
What are random variables and their types?	4
Short definitions of the following terms:	4
Associate a rule with the following equations.	6
What is Bayes' Rule?	8
Write down axioms of Probability which are also called Kolmogorov's theorem.	9
Example of FJD (Full Joint Distribution) inference.	9
Probabilistic Reasoning	10
What are the parts of a bayesian network?	10
What does one row of a conditional probability table represent? Does it represent an atomic event?	10
Is the graph below a polytree? When is node F independent from node E?	11
What are the semantics of Bayesian Networks?	11
Discuss the compactness of BN over FJP.	12
What does it mean to create a correct Bayesian network from the full joint distribution?	12
What is a Chain Rule?	12
What is the correct way to add nodes and what happens if you don't?	13
Explain the two types of independence criteria used in bayesian networks.	13
Discuss deterministic and uncertain node relationships.	13
How to work with random variables that have continuous states?	14
What is Hybrid Bayesian Network?	14
Explain polytrees, what are their problems and how to solve these.	15
What are the types of inferencing?	15
Discuss each direct method separately.	15
Discuss approximate inference methods.	17
Probabilistic Reasoning over Time	18

What PR over time is about? Give a short introduction. State the difference between this chapter and the previous chapter.	18
What are the conventions used in this?	19
What are the problems and their solutions to the unbounded variables?	19
What are the inference methods in DBN (and why can't we use the static Bayesian network inference methods here)?	21
What is Dynamic Bayesian Network (DBN)? How to construct DBNs?	22
Is it possible to translate DBN to HMM and HMM to DBN? What is the difference between HMM and DBN?	23
What is Kalman Filter? Can you convert Kalman Filter to DBN and DBN to Kalman Filter?	24
Hidden Markov Models	24
What are Hidden Markov Models?	24
What is the Viterbi algorithm and how does it work?	25
Making Simple Decisions	26
What is utility?	26
Explain expected utility.	26
Why can't AI solve these problems?	26
What is a lottery?	27
What are the constraints on making a rational preference?	27
Discuss important utility principles.	29
What is a utility function?	29
How to estimate the nonlinearity of money?	30
Map the graph and talk about risk aversion.	30
What are Multivariate Utility Functions?	31
What is dominance? What are the differences between Strict Dominance and Stochastic Dominance?	31
Talk about preference without uncertainty.	32
What are Decision Networks?	33
What is the value of information? What is the value of perfect information (VPI)?	34
Making Complex Decisions	34
In the lecture slides the questions about an agent trying to find its path from start to end why don't we solve it as an AI search problem instead use the Markov decision process?	34
What is Markov Decision Process?	34
What are the three components of a Markov decision process?	34
Define the following terms.	34
What is a discount factor?	35
What are the two algorithms (taught in class) that are used to solve Markov decisions processes and describe them?	36

Uncertainty

1. What is a qualification problem?

The problem of reasoning about the conditions to meet for an event (required for an event) to have a given consequence. We had an example of reaching the airport at 10:30 on time if you leave 30 minutes earlier to reach there. In this case, the traffic should be minimum, the car should not break down and other kinds of things (enumeration of possible events).

2. What is a rational decision?

It is a combination of likelihood and relative importance. The right thing to do - the rational decision - therefore depends on both the relative importance of various goals and the likelihood that, the degree to which, they will be achieved.

3. What is the advantage of using probability over first-order logic (FOL)?

Let's say we have given the rule: $\forall p \text{ symptoms} \rightarrow \text{Disease}(p, \text{Cavity})$.

This rule is wrong because not all patients with all the symptoms always get Cavity. Thus:

- In FOL like above, the rule is given as 100% certainty and it cannot handle uncertainty.
- Lazy (we cannot list all causes and results). It is too much work.
- Theoretical ignorance (we don't know all causes any symptoms).
- Practical ignorance (we have not done any tests to collect all results).

4. What is the difference between causal and diagnostic rules?

Causal inference happens when some event is the cause of another event (we can denote it as $P(A|B)$ where A is the cause of B). For diagnostic inference, as understandable from its name, we use effect (symptom) to infer a cause (We can denote it as $P(B|A)$). For this case, we have to use Baye's Rule.

- If we are doing diagnostic models (from symptom to causes/effects), we have to specify additional dependencies in otherwise independent ones
- But if we have a causal model, then we specify less

- A causal model is a more natural way to think.

5. What does the probability of 0 and 1 mean?

The main tool for dealing with degrees of belief will be probability theory which assigns to each sentence a numerical degree of belief between 0 and 1.

Assigning probability of 0 ~ unequivocal belief that the sentence is **false**.

Assigning probability of 1 ~ unequivocal belief that the sentence is **true**.

Thus, we say that probabilities between 0 and 1 correspond to intermediate degrees of belief in the truth of the sentence. The sentence is itself in fact either true or false.

Without a doubt the degree of belief a statement is false or true. The belief depends on the sensors of the agent **not the actual truth of the world** (precepts of the world)

- The true event is the event that always happens
- Before percepts are called prior and after is called posterior or conditional probability

6. Explain the decision-making process with MEU.

We use utility theory to represent and reason with preferences. Utility theory says that every state has a degree of usefulness, or utility, to an agent and that the agent will prefer states with higher utility.

Preferences, as expressed by utilities, are combined with probabilities in the general theory of rational decision called decision theory:

$$\text{Decision theory} = \text{probability theory} + \text{utility theory}$$

The fundamental idea of decision theory is that an agent is rational if and only if it chooses the action that yields the highest expected utility, averaged over the possible outcomes of the action. This is called the principle of Maximum Expected Utility (MEU).

7. What are random variables and their types?

A random variable is referred to as a part of the world whose status is initially unknown. We may have different types of random variables such as boolean, discrete, and continuous types of random variables.

8. Short definitions of the following terms:

- **Probability theory**

Probability theory uses an extension of propositional logic for its sentences. The language is slightly more expressive than propositional logic.

Probability theory, a branch of mathematics concerned with the analysis of random phenomena. The outcome of a random event cannot be determined before it occurs, but it may be any one of several possible outcomes

- **Utility theory**

Utility theory says that every state has a degree of usefulness, or utility, to an agent and that the agent will prefer states with higher utility

- **Decision theory**

The fundamental idea of decision theory is that an agent is rational if and only if it chooses the action that yields the highest expected utility, averaged over the possible outcomes of the action.

- **Atomic event**

Assignment of values to all the variables of which the world is composed.

Properties:

1. **mutually exclusive** - two events can not occur at the same time,
 $(cavity \wedge toothache) \vee (cavity \wedge \neg toothache)$
2. **exhaustive** - at least one atomic event has to occur when performing the experiment,
3. any proposition is logically equivalent to the disjunction of all atomic events that entail the truth of the proposition. For instance, the cavity is equivalent to $(cavity \wedge toothache) \vee (cavity \wedge \neg toothache)$

- **Prior probability**

Unconditional or prior probability associated with a proposition a is the degree of belief accorded to it in the absence of any other information written as $P(a)$. Shortly, The degree of belief before any percept information is available.

It is important to note that **P(a) can be used only when there is no other information**. As soon as some new information is known, we must reason with the conditional probability of a given that new information.

$P(Weather)$ describes the prior probability distribution for the random variable Weather.

- **Joint probability distribution**

P(Weather, Cavity) denotes probabilities of all combinations of the values of a set of random variables and is represented by a 4x2 table of probabilities called the joint probability distribution of Weather and Cavity.

- **Full joint probability distribution**

A full joint probability is **exhaustive** it **contains all possible atomic events that can occur** and **one of them must be true at one instance**. Specify the probability of every atomic event. It is a complete specification of one's uncertainty of the world in question.

In other words, sometimes it will be useful to think about the **complete set of random variables used to describe the world**: a joint probability distribution that covers this complete set is called the full joint probability distribution.

An example can be if the world consists of just the variables *Cavity*, *Toothache*, and *Weather*, then the full joint probability distribution is given by

$$P(Cavity, Toothache, Weather)$$

This joint distribution can be represented as a 2x2x4 table with 16 entries.

A full joint distribution specifies the probability of every atomic event and is, therefore, a complete specification of one's uncertainty about the world in question.

9. Associate a rule with the following equations.

- **Product rule**

The product rule is equal to

$$P(a \wedge b) = P(a|b) P(b)$$

which comes from the conditional probability

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

Thus, the final equation for product rule would be something like the following:

$$P(X, Y) = P(X|Y) P(Y)$$

- **Marginalization rule**

We can write the following general marginalization rule for any sets of variables Y and Z:

$$P(Y) = \sum_z P(Y, z)$$

- **Conditioning rule**

A variant of marginalization role involves conditional probabilities instead of joint probabilities, using the product rule:

$$P(Y) = \sum_z P(Y | z) P(z)$$

- **Independence (marginal/absolute independence)**

Let's say we have an example of $P(\text{toothache}, \text{catch}, \text{cavity}, \text{Weather} = \text{cloudy})$. We can use the product rule:

$$\begin{aligned} & P(\text{toothache}, \text{catch}, \text{cavity}, \text{Weather} = \text{cloudy}) = \\ & = P(\text{Weather} = \text{cloudy} | \text{toothache}, \text{catch}, \text{cavity}) P(\text{toothache}, \text{catch}, \text{cavity}) \end{aligned}$$

As we saw from the example, one's dental problems typically don't influence the weather. Therefore,

$$P(\text{Weather} = \text{cloudy} | \text{toothache}, \text{catch}, \text{cavity}) = P(\text{Weather} = \text{cloudy})$$

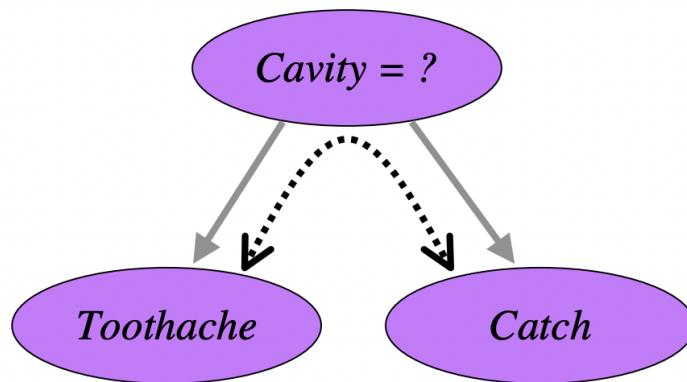
Thus, the property we used in writing the above equation is called independence (also marginal and absolute independence).

Independence between variables X and Y can be written as follows:

$$P(X | Y) = P(X) P(Y | X) = P(Y) P(X \wedge Y) = P(X)P(Y)$$

- **Conditional independence**

Let's give an example as follows:



Without evidence on Cavity, our beliefs regarding Toothache and Catch may affect each other. However, given evidence on Cavity, Toothache and Catch have no effect on each other.

Thus, we say that variables are independent, however, given the presence or absence of a cavity. Each is directly caused by the cavity, but neither has a direct effect on the other.

$$P(X, Y | Z) = P(X | Z) P(Y | Z)$$

- **Chain Rule**

We may rewrite the joint distribution in terms of a conditional probability, using the product rule:

$$P(x_1, \dots, x_n) = P(x_n | x_{n-1}, \dots, x_1) P(x_{n-1}, \dots, x_1)$$

Repeat the process, reducing each conjunctive probability to a conditional probability and smaller conjunction:

$$\begin{aligned} P(x_1, \dots, x_n) &= P(x_n | x_{n-1}, \dots, x_1) P(x_{n-1} | x_{n-2}, \dots, x_1) \dots P(x_2 | x_1) P(x_1) = \\ &= \prod_{i=1}^n P(x_i | x_{i-1}, \dots, x_1) \end{aligned}$$

Identity holds true for any set of random variables and is called the chain rule.

- **Inference rule (General inference procedure)**

The fundamental idea of inferencing is to compute the posterior probability distribution for a set of query variables, given some observed event (i.e., some assignment of values to a set of evidence variables).

Let X be the query variable (Cavity in the example), let E be the set of evidence variables (just Toothache in the example), let e be the observed values for them, and let Y be the remaining unobserved variables (just Catch in the example). The query is $P(X | e)$ and can be evaluated as

$$P(X | e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y)$$

10. What is Bayes' Rule?

Product rule can be written in two forms because of the commutativity of conjunction:

$$P(a \wedge b) = P(a | b)P(b) \text{ and } P(a \wedge b) = P(b | a)P(a)$$

Equating the two right-hand sides and diving by $P(a)$, we get:

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

Known as Bayes'rule (also Bayes' law or Bayes' theorem).

The more general case of multi-valued variables can be written in the P notation (probability distribution) as

$$P(Y | X) = \frac{P(X | Y)P(Y)}{P(X)} \text{ or } P(Y | X, e) = \frac{P(X | Y, e)P(Y, e)}{P(X | e)}$$

11. Write down axioms of Probability which are also called Kolmogorov's theorem.

Basic axioms defining probability scale and endpoints

1. All probabilities are between 0 and 1. For any proposition a,

$$0 \leq P(a) \leq 1$$

2. Necessarily true (i.e., valid) propositions have probability 1, and necessarily false (i.e., unsatisfiable) propositions have probability 0.

$$P(true) = 1 \\ P(false) = 0$$

3. The probability of disjunction is given by

$$P(a \vee b) = P(a) + P(b) - P(a \wedge b)$$

12. Example of FJD (Full Joint Distribution) inference.

We had an inference example in our lecture slides regarding $P(Toothache, Cavity, Catch)$ which is shown as follows:

	<i>toothache</i>		\neg <i>toothache</i>	
	<i>catch</i>	\neg <i>catch</i>	<i>catch</i>	\neg <i>catch</i>
<i>cavity</i>	0.108	0.012	0.072	0.008
\neg <i>cavity</i>	0.016	0.064	0.144	0.576
A full joint distribution for the <i>Toothache, Cavity, Catch</i> world.				

Thus, we can find the FJD using Marginalization (this process is called marginalization, or summing out - because the variables other than Cavity are summed out).

We can write the following general marginalization rule for any sets of variables Y and Z:

$$P(Y) = \sum_z P(Y, z)$$

Probabilistic Reasoning

1. What are the parts of a bayesian network?

A Bayesian network is a **directed graph** in which **each node is annotated with quantitative probability information**.

The full specification is:

1. A set of *random variables* (boolean, discrete, or continuous) are the nodes of the network
2. A set of *directed links* or arrows **connects pairs of nodes**. If there is an arrow from node X to node Y, **X is said to be the parent of Y**.
3. Each node X_i has a **conditional probability distribution**:

$$P(X_i | \text{Parents}(X_i))$$

that quantifies the effect of the parents on the node.

4. The graph has **no directed cycles** (and hence is a directed, acyclic graph, or DAG).

The **topology of the network** is the **set of nodes and links** which **specifies the conditional independence relationships** that hold in the domain. The intuitive meaning of an **arrow** in a properly constructed network is usually that **X has a direct influence on Y**. Once the topology of the Bayesian network is laid out, we have to specify a conditional probability distribution for each variable, given its parents.

A combination of the *topology and the conditional distributions* suffices to specify the *full joint distribution for all the variables*.

2. What does one row of a conditional probability table represent? Does it represent an atomic event?

Each row in a CPT contains the **conditional probability of each node** value for a **conditioning case**. The conditioning case is just a **possible combination of values for the parent nodes** (miniature atomic event). Each **row must sum to 1** because the entries represent an exhaustive set of cases for the variable.

In general, a table for a Boolean variable with k Boolean parents contains 2^k **independently specifiable probabilities**. **Node with no parents has only one row**, **representing the prior probabilities** of each possible value of the variable.

3. Is the graph below a polytree? When is node F independent from node E?

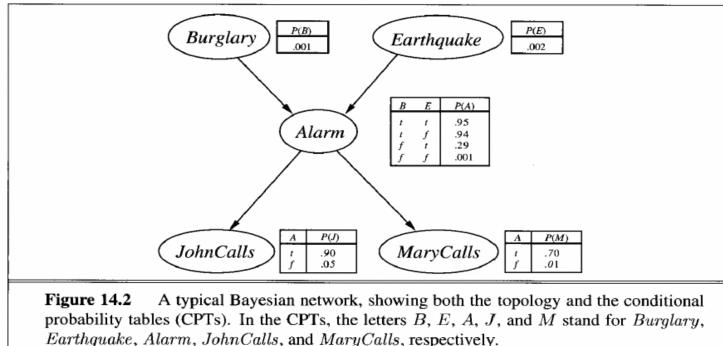


Figure 14.2 A typical Bayesian network, showing both the topology and the conditional probability tables (CPTs). In the CPTs, the letters *B*, *E*, *A*, *J*, and *M* stand for *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, and *MaryCalls*, respectively.

It is not a polytree. **If a node can be reached to all its nodes only by one path**, then it's a polytree. Node F is independent of node E when another parent node D is given.

4. What are the semantics of Bayesian Networks?

Two ways in which one can understand the **semantics of Bayesian networks**:

1. See the network as a **representation of the joint probability distribution**.
2. View it as an encoding of a **collection of conditional independence statements**.

These two views are equivalent, but the **first** turn out to be helpful in understanding **how to construct networks**, whereas the **second** is helpful in designing inference procedures.

The Bayesian network provides a complete description of the domain. Every entry in the full joint probability distribution can be calculated from the information in the network. General entry in the join distribution is the probability of the conjunction of particular assignments to each variable, such as $P(X_1 = x_1 \wedge \dots \wedge X_n = x_n)$ abbreviated as

$P(x_1, \dots, x_n)$ as an abbreviation for this.

The value of this entry is given by the formula

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$$

where $\text{parents}(X_i)$ denotes the specific values of the variables in $\text{Parents}(X_i)$.

Each entry in joint distribution is represented by the product of the appropriate elements of the conditional probability tables (CPTs) in the Bayesian network. **CPTs, therefore, provide a decomposed representation of the joint distribution.**

5. Discuss the compactness of BN over FJP.

The Bayesian network can often be far more compact than the full joint distribution. In the case of Bayesian networks, it is reasonable to suppose that in most domains each random variable is directly influenced by at most k others.

If we assume n Boolean variables for simplicity, then the amount of information needed to specify each conditional probability table will be at most 2^k numbers, and the complete network can be specified by $n2^k$ numbers. In contrast, the joint distribution contains 2^n numbers.

Example:

Suppose we have $n = 30$ nodes, each with five parents ($k = 5$); the Bayesian network requires 960 numbers, but full joint distribution requires over a billion.

6. What does it mean to create a correct Bayesian network from the full joint distribution?

Equation 2.1 says that the value of the entry is given by the formula:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$$

Comparing the Chain rule formula with Equation 2.1, we see that the specification of the joint distribution is equivalent to the general assertion that, for every variable X_i in the network,

$$P(X_i | X_{i-1}, \dots, X_1) = P(X_i | \text{Parents}(X_i))$$

provided that $\text{Parents}(X_i) \subset \{X_{i-1}, \dots, X_1\}$

The above equation says that the Bayesian network is a correct representation of the domain only if each node is conditionally independent of its predecessors in the node order, given its parents.

Intuitively, the parents of node X_i should contain all those nodes in X_1, \dots, X_{i-1} that directly influence X_i .

7. What is a Chain Rule?

We may rewrite the joint distribution in terms of a conditional probability, using the product rule:

$$P(x_1, \dots, x_n) = P(x_n | x_{n-1}, \dots, x_1)P(x_{n-1}, \dots, x_1)$$

Repeat the process, reducing each conjunctive probability to a conditional probability and smaller conjunction:

$$\begin{aligned} P(x_1, \dots, x_n) &= P(x_n | x_{n-1}, \dots, x_1)P(x_{n-1} | x_{n-2}, \dots, x_1) \dots P(x_2 | x_1)P(x_1) = \\ &= \prod_{i=1}^n P(x_i | x_{i-1}, \dots, x_1) \end{aligned}$$

7. What is the correct way to add nodes and what happens if you don't?

We first need to add the "root causes", then the variables they influence, and so on until we reach the "leaves", which have no direct causal influence on the other variables. If we don't do like that, there will have more links, hence more probabilities, and some probabilities will be difficult or weird to define.

8. Explain the two types of independence criteria used in bayesian networks.

- A node is **conditionally independent** of its **non-decedents** given its parents
eg: *JohnCall* is independent of *Burglary* and *Earthquake* given *alarm*
- A node is **conditionally independent** of **all other nodes** in the network given **its parents**, children, and parents of its children i.e. **Markov blanket**

9. Discuss deterministic and uncertain node relationships.

1. Certain relationships (deterministic nodes)

A deterministic node has **its value specified exactly by the values of its parents, with no uncertainty**. The relationship can be a logical one. For example, the relationship between the parent nodes Canadian, US, Mexican, and the child node NorthAmerican are simply that the child is a **disjunction (V)** of the parents. Relationships can also be numerical. For example, if the parent nodes are the prices of a particular model of car at several dealers, and the child node is the price that a **bargain hunter ends up paying**, then the **child node is the minimum** of the parent values.

2. Uncertain relationships (nondeterministic nodes)

Uncertain relationships are often characterized by so-called "**noisy**" logical relationships. A standard example can be **noisy-OR** relation which is the **generalization of the logical OR**. In propositional logic: Fever is true if and only if Cold, Flu, or Malaria is true.

The noisy-OR model allows for **uncertainty about the ability of each parent to cause the child to be true**.

The causal relationship between parent and child may be inhibited so a patient could have a cold, but not exhibit a fever.

Individual inhibition probabilities:

$$P(\neg\text{fever} \mid \text{cold}, \neg\text{flu}, \neg\text{malaria}) = 0.6$$

$$P(\neg\text{fever} \mid \neg\text{cold}, \text{flu}, \neg\text{malaria}) = 0.2$$

$$P(\neg\text{fever} \mid \neg\text{cold}, \neg\text{flu}, \text{malaria}) = 0.1$$

The model makes two assumptions:

- all possible causes are listed (not as strict as it seems, because we can always add a so-called leak node that covers "miscellaneous causes.")
- inhibition of each parent is independent of inhibition of any other parents: for example, whatever inhibits Malaria from causing a fever is independent of whatever inhibits Flu from causing a fever.

Given these assumptions, Fever is false if and only if all its true parents are inhibited, and the probability of Fever=false is the product of the inhibition probabilities for each parent.

10. How to work with random variables that have continuous states?

Many real-world problems involve continuous quantities, such as height, mass, temperature, and money. Continuous variables have an infinite number of possible values, so it is impossible to specify conditional probabilities explicitly for each value. One possible way to handle continuous variables is using **discretization**, i.e. dividing up the possible values into a fixed set of intervals:

example: $< 0^\circ\text{C}$, $0^\circ\text{C} - 100^\circ\text{C}$ and $> 100^\circ\text{C}$

Discretization is sometimes an adequate solution but often results in a considerable loss of accuracy and very large CPTs.

Another solution is to define standard families of **probability density functions** that are specified by a finite number of parameters (Gaussian).

11. What is Hybrid Bayesian Network?

A hybrid Bayesian network is a network with both discrete and continuous variables. To specify a hybrid network, we have to specify two new kinds of distributions:

1. the conditional distribution for a continuous variable given discrete or continuous parents,
2. the conditional distribution for a discrete variable given continuous parents.

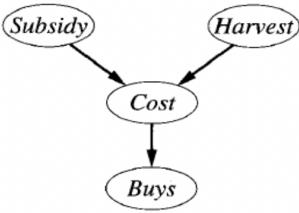


Figure 14.5 A simple network with discrete variables (*Subsidy* and *Buys*) and continuous variables (*Harvest* and *Cost*).

12. Explain polytrees, what are their problems and how to solve these.

Clustering algorithms

The basic idea of clustering is to **join individual nodes of the network to form cluster nodes** in such a way that the **resulting network is a polytree**. **If a node can be reached to all its nodes only by one path**, then it's a polytree. If this is not the case, to make the inference tractable we form cluster nodes creating a mega node,

12. What are the types of inferencing?

There are two types of inferencing: exact inference and approximate inference.

The basic task for any probabilistic inference system is to **compute the posterior probability distribution** for a set of query variables, **given some observed event**. We denote it as $P(X | e)$. For exact inference, we learned two algorithms: **enumeration** and **variable elimination**.

Given the **intractability** (impossible to manage, control) of exact inference in large, multiply connected networks, it is **essential to consider approximate inference methods**. We have learned several approximate inference algorithms which are Markov Chain and Direct Sampling. Markov Chain has **Monte Carlo Markov chain** method and direct sampling has **rejection sampling** and **likelihood weighting**.

13. Discuss each direct method separately.

1. Enumeration

As we know any **conditional probability** can be computed by **summing terms from the full joint distribution**.

$$P(X | e) = \alpha P(X, e) = \alpha \sum_y P(X, e, y)$$

The equation above showed that the terms $P(x, e, y)$ in the join distribution can be written as **products of conditional probabilities from the network**. A query can

be answered using a Bayesian network by **computing sums of products of conditional probabilities from the network**.

Example: $P(\text{Burglary} \mid \text{JohnCalls} = \text{true}, \text{MaryCalls} = \text{true})$

Hidden variables: Earthquake and Alarm.

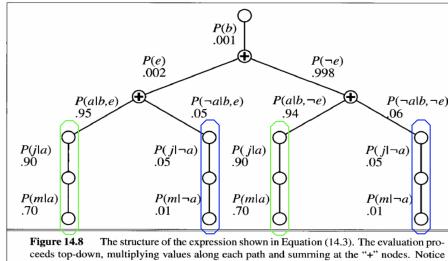
$$P(B \mid j, m) = \alpha P(B, j, m) = \alpha \sum_{e \in a} \sum_{b \in b} P(b) P(e) P(a \mid b, e) P(j \mid a) P(m \mid a)$$

In this case, we have 20 terms because we have 2 summations and 5 equations and these 2 summations can be true or false ($e, \neg e, a, \neg a$).

In the worst case, where we have to sum out almost all the variables, the complexity of the algorithm for a network with n Boolean variables is $O(n2^n)$.

In the above example, we can move the $P(b)$ (because it is constant) outside the summations over a and e , and the $P(e)$ term can be moved outside the summation over a .

$$P(B \mid j, m) = \alpha P(B, j, m) = \alpha P(b) \sum_e P(e) \sum_a P(a \mid b, e) P(j \mid a) P(m \mid a)$$



```

function ENUMERATION-ASK( $X, e, bn$ ) returns a distribution over  $X$ 
  inputs:  $X$ , the query variable
   $e$ , observed values for variables  $E$ 
   $bn$ , a Bayes net with variables  $\{X\} \cup E \cup Y$  /*  $Y$  = hidden variables */
   $Q(X) \leftarrow$  a distribution over  $X$ , initially empty
  for each value  $x_i$  of  $X$  do
    extend  $e$  with value  $x_i$  for  $X$ 
     $Q(x_i) \leftarrow \text{ENUMERATE-ALL(VARS}[bn], e)$ 
  return NORMALIZE( $Q(X)$ )

function ENUMERATE-ALL( $vars, e$ ) returns a real number
  if  $EMPT(Y)$  then return 1.0
   $Y \leftarrow \text{FIRST}(vars)$ 
  if  $Y$  has value  $y$  in  $e$ 
    then return  $P(y \mid \text{parents}(Y)) \times \text{ENUMERATE-ALL}(\text{REST}(vars), e)$ 
  else return  $\sum_y P(y \mid \text{parents}(Y)) \times \text{ENUMERATE-ALL}(\text{REST}(vars), e_y)$ 
    where  $e_y$  is  $e$  extended with  $Y = y$ 

```

Figure 14.9 The enumeration algorithm for answering queries on Bayesian networks.

ENUMERATION-ASK algorithm evaluates trees using **depth-first recursion**.

The space complexity of ENUMERATION-ASK is only **linear** in the number of variables effectively, the algorithm sums over the full joint distribution without ever constructing it explicitly.

The time complexity for a network with n Boolean variables is always $O(2^n)$

2. Variable Elimination

The enumeration algorithm can be improved substantially by **eliminating repeated calculations**. Thus, we need to do the **calculation once** and **save the results for later use**.

Variable elimination works by:

1. Evaluation expressions we showed in Enumeration in right-to-left order

2. Intermediate results are stored, and summations over each variable are done only for those portions of the expression that depend on the variable.

$$P(B \mid j, m) = \alpha \underbrace{P(B)}_B \sum_e \underbrace{P(e)}_E \sum_a \underbrace{P(a \mid B, e)}_A \underbrace{P(j \mid a)}_J \underbrace{P(m \mid a)}_M$$

Parts B, E, A, J, M are called **factors** (which are storage places).

To conclude, two basic computational operations are required

- pointwise product of factors
- summing out a variable from a product of factors

1. summing out a variable from a product of factors:

any factor that does not depend on the variable to be summed out can be moved outside the summation process,

2. pointwise product inside the summation is computed, and the variable is summed out of the resulting matrix.

Variable elimination is **more efficient than enumeration** because it avoids repeated computations. Time and space requirements of variable elimination are dominated by the size of the largest factor constructed during the operation of the algorithm; this in turn is determined by the order of elimination of variables and by the structure of the network.

14. Discuss approximate inference methods.

Given the **intractability** (impossible to control, manage) of exact inference in large, multiply connected networks, it is **essential to consider approximate inference methods**.

Monte Carlo algorithms are **randomized sampling algorithms**, that provide **approximate** answers **whose accuracy depends on the number of samples generated**.

There are two families of algorithms: **direct sampling** and **Markov chain sampling**.

The basic element in any sampling algorithm is the **generation of samples from the known probability distribution**. The simplest kind of random sampling process for Bayesian networks **generates events from a network that has no evidence associated with it**.

The idea here is to sample each variable in turn, in topological order.

1. Rejection sampling

It generates samples from the prior distribution specified by the network. It starts from the root and goes till to the leaf. It rejects all those that do not match the evidence. Estimate $P(X = x \mid e)$ is obtained by counting how often $X=x$ occurs in

the remaining samples. The biggest problem with rejection sampling is that it **rejects too many samples**.

2. Likelihood weighting

Likelihood weighting avoids the inefficiency of rejection sampling by generating only events that are consistent with the evidence e . It fixes the values for the evidence variables E and samples only the remaining variables X and Y . This guarantees that each event generated is consistent with the evidence.

3. Markov Chain Monte Carlo (MCMC) algorithm

Unlike other sampling algorithms, which generate each event from scratch, MCMC generates each event by making random changes to the preceding event/state. The next event/state is generated by randomly sampling a value for one of the nonevidence variables X_i , conditioned on the current values of the variables in the

Markov blanket. MCMC, therefore, wanders randomly around the state space - the space of complete assignments - flipping one variable at a time, but keeping the evidence variables fixed.

To summarize, it does not always start from scratch and unlike the other two does not always go from root to leaf node. It starts with a random variable that it has already sampled. Instead of sampling an entirely new random variable, it samples from the Markov blankets of the last sampled node. It randomly initializes the hidden variables then starts moving through the **Markov blanket**.

Probabilistic Reasoning over Time

1. What PR over time is about? Give a short introduction. State the difference between this chapter and the previous chapter.

In previous chapters, we have talked and studied causal reasoning in static environments. probabilistic reasoning over time is not only about uncertainty, but also to uncertainty about how the environment changes over time. Thus, in this chapter, we we not only consider the noise and uncertainty of the current world we also look at the past events.

We try to predict state in present, future or past based on historical events. We will also talk about three specific kinds of models: DBNs, HMMs, Kalman Filters.

An example can be about treating a diabetic patient. We have evidence such as insulin doses, food intake, blood sugar measurements, and other physical signs. Our task is to assess the current state of the patient, including the actual blood sugar level and insulin level. Given this information, the doctor (or patient) makes a decision about the patient's food intake and insulin dose. Unlike the case of earthquake-burglary-alarm, here the dynamic aspects of the problem are essential: blood sugar levels and measurements thereof can change rapidly over time.

2. What are the conventions used in this?

The interval between time slices also depends on the problem. We will generally assume a fixed, finite interval, i.e. times can be labeled by integers. We will assume that the state sequence starts at $t = 0$, e.g. R_0, R_1, R_2, \dots . We will assume that evidence starts arriving at $t = 1$ rather than $t = 0$ e.g. U_1, U_2, \dots . We will use the notation $a:b$ to denote the sequence of integers from a to b (inclusive), and notation $X_{a:b}$ to denote the corresponding set of variables from X_a to X_b (i.e. $U_{1:3}$ corresponds to the variables U_1, U_2, U_3)

3. What are the problems and their solutions to the unbounded variables?

For the next step we need to specify dependencies among the variables. We could follow the procedure laid down in previous chapter placing the variables in some order and asking questions about conditional independence of predecessors, given some set of parents. However: the set of variables is unbounded, because it includes the state and evidence variables for every time slice.

This creates two problems:

1. we might have to specify an unbounded number of conditional probability tables, one for each variable in each slice

This is solved by assuming that changes in the world state are caused by a stationary process (process of change that is governed by laws that do not themselves change over time). Given the assumption of stationarity, we need specify conditional distributions only for the variables within a "representative" time slice.

2. each one might involve an unbounded number of parents

Handling the potentially infinite number of parents is solved by making what is called a **Markov assumption**:

The current state depends on only a finite history of previous states

First-order Markov process: current state depends only on the previous state and not on any earlier states.

$$P(X_t | X_{0:t-1}) = P(X_t | X_{t-1})$$

in a first-order Markov process, laws describing how the state evolves over time are contained entirely within the conditional distribution $P(X_t | X_{t-1})$, which we call the **transition model for first-order processes**.

Transition model for a **second-order Markov process** is the conditional distribution $P(X_t | X_{t-2}, X_{t-1})$

Typically, we assume that the **evidence variables** at time t depend only on the current state:

$$P(E_t | X_{0:t-1}, E_{0:t-1}) = P(E_t | X_t)$$

Conditional distribution $P(E_t | X_t)$ is called the **sensor model** (or sometimes the observation model), because it describes how the “sensors” (evidence variables) are affected by the actual state of the world.

Direction of the dependence (the “arrow”) goes from state to sensor values because the state of the world causes the sensors to take on particular values.

In addition to transition model and sensor model, we need to specify a prior probability $P(X_0)$ over the states at time 0.

First-order Markov assumption says that the state variables contain all the information needed to characterize the probability distribution for the next time slice. Sometimes assumption is only approximate, as in the case of predicting rain only on the basis of whether it rained the previous day.

There are two possible fixes if the approximation proves too inaccurate:

1. Increasing the order of the Markov process model; e.g. make a second-order model by adding $Rain_{t-2}$ as parent of $Rain_t$, which might give slightly more accurate predictions,

2. Increasing the set of state variables; e.g. add Season_t to allow us to incorporate historical records of rainy seasons, or add Temperature_t , Humidity_t and Pressure_t to allow us to use a physical model of rainy conditions.

Adding state variables might improve the system's predictive power but also increases the prediction requirements: we now have to predict the new variables as well.

4. What are the inference methods in DBN (and why can't we use the static Bayesian network inference methods here)?

1. Filtering or monitoring: $P(X_t | e_{1:t})$

It is the task of computing the posterior distribution over the current state, given all the evidence up to date, assuming that evidence arrives in continuous stream beginning at $t = 1$.

As an example, we need to compute the probability of rain today, given all the observations of the umbrella carrier made so far.

Given the result of filtering up to time t , one can compute the result for $t + 1$ from the new evidence e_{t+1} .

$$P(X_{t+1} | e_{1:t+1}) = f(e_{t+1}, P(X_t | e_{1:t}))$$

process is often called recursive estimation.

This calculation composed of two parts:

1. the current state distribution is projected forward from t to $t + 1$;
2. it is updated using the new evidence e_{t+1} .

$$\begin{aligned} P(X_{t+1} | e_{1:t+1}) &= \\ &= P(X_{t+1} | e_{t+1}, e_{1:t}) \quad (\text{dividing up the evidence}) \\ &= \alpha P(e_{t+1} | X_{t+1}, e_{1:t}) P(X_{t+1} | e_{1:t}) \quad (\text{using Bayes' rule}) \\ &= \alpha P(e_{t+1} | X_{t+1}) P(X_{t+1} | e_{1:t}) \quad (\text{by the Markov property of evidence}) \end{aligned}$$

α is a normalizing constant. second term, $P(X_{t+1} | e_{1:t})$ represents a one-step prediction of the next state and $P(e_{t+1} | X_{t+1})$ is obtainable directly from the sensor model.

2. Prediction: $P(X_{t+k} | e_{1:t})$

It is the task of computing the **posterior distribution over the future state**, given all evidence to date umbrella.

As an example, we need to compute the probability of rain three days from now, given all the observations of the umbrella-carrier made so far.

Prediction is useful for evaluating possible courses of action.

Task of prediction can be seen simply as **filtering without the addition of new evidence**.

3. **Smoothing or hindsight:** $P(X_k | e_{1:t})$ for some k such that $0 \leq k < t$

It is the task of computing the **posterior distribution over the past state**, given all evidence up to the present.

As an example, we need to compute the probability that it rained last Wednesday, given all the observations of the umbrella carrier made up to today.

Hindsight provides a better estimate of the state than was available at the time, because it incorporates more evidence.

4. **Most likely explanation:** $\operatorname{argmax}_{x_{1:t}} P(x_{1:t} | e_{1:t})$

This algorithm says that, **given a sequence of observations**, we might **wish to find the sequence of states that is most likely to have generated those observations**.

As an example, if an umbrella appears on each of the first three days and is absent on the fourth, then the most likely explanation is that it rained on the first three days and did not rain on the fourth.

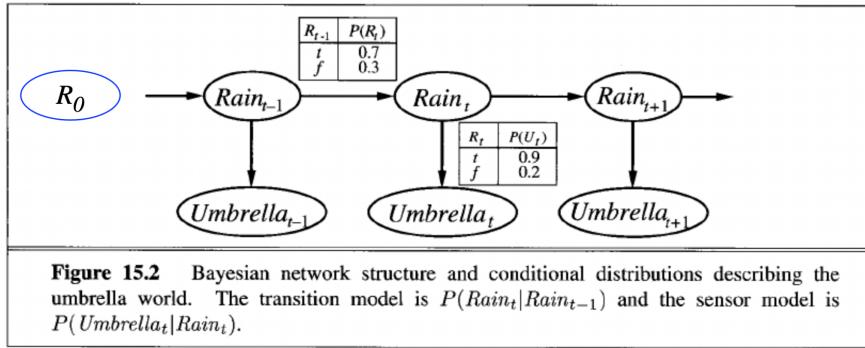
It is useful in many applications, including speech recognition (the aim is to find the **most likely sequence of words**, given a series of sounds) and the reconstruction of bit strings transmitted over a noisy channel.

The algorithm for computing the most likely sequence is similar to filtering: it **runs forward along the sequence**, computing the m message at each time step. At the end, it will have the **probability for the most likely sequence** reaching **each of the final states**. One can thus **easily select the most likely sequence overall**. The **algorithm we have just described is called the Viterbi algorithm, after its inventor**.

5. What is Dynamic Bayesian Network (DBN)? How to construct DBNs?

A dynamic Bayesian network, or DBN, is a **Bayesian network** that represents a **temporal probability model**.

Example can be as follows:



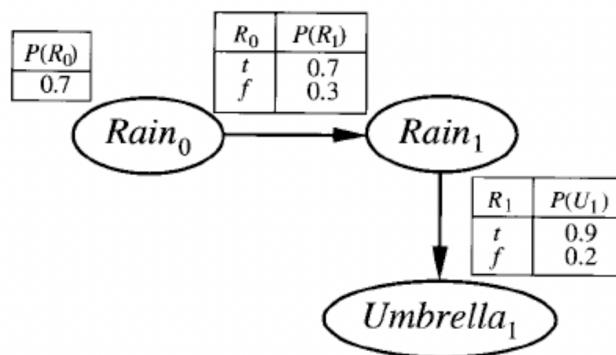
In general:

1. each slice of a DBN can have any number of state variables X_t and evidence variables E_t .
2. for simplicity, we will assume that the variables and their links are exactly replicated from slice to slice and the DBN represents a first-order Markov process.

To construct a DBN, one must specify three kinds of information:

- the prior distribution over the state variables $P(X_0)$;
- the transition model $P(X_{t+1} | X_t)$;
- and the sensor model $P(E_t | X_t)$;
- one must also specify the topology of the connections between successive slices and between the state and evidence variables.

Example for DBN can be as follows:



6. Is it possible to translate DBN to HMM and HMM to DBN? What is the difference between HMM and DBN?

Every hidden Markov model can be represented as a DBN with a single state variable and a single evidence variable.

Every discrete variable DBN can be represented as an HMM. We can combine all the state variables in the DBN into a single state variable whose values are all possible tuples of values of the individual state variables.

Difference is that by decomposing the state of a complex system into its constituent variables, DBN is able to take advantage of **sparseness** in the temporal probability model.

Example:

DBN with 20 Boolean state variables, each of which has three parents in the preceding slice

DBN transition model has $20 \times 2^3 = 160$ probabilities

However: corresponding HMM has 2^{20} states and therefore 2^{40} , or roughly a trillion, probabilities in the transition matrix.

Bad for at least three reasons:

- first, HMM itself requires **much more space**
- second, huge transition matrix makes **HMM inference much more expensive**
- third, the **problem of learning such a huge number of parameters** makes the pure HMM model **unsuitable for large problems**.

7. What is Kalman Filter? Can you convert Kalman Filter to DBN and DBN to Kalman Filter?

Every Kalman filter model can be represented in a DBN with **continuous variables** and **linear Gaussian conditional distributions**. However: not every DBN can be represented by a Kalman filter model.

The reason is that, in a Kalman filter, the **current state distribution is always a single multivariate Gaussian distribution**-that is a single "bump" in a particular location. DBNs can model arbitrary distributions. For many real-world applications, this flexibility is essential.

Hidden Markov Models

1. What are Hidden Markov Models?

A Markov System

It has N states, called s_1, s_2, \dots, s_N . There are discrete timesteps $t = 0, t = 1, \dots$ On the t'th timestep the system is in exactly one of the available states. Call it

q_t where $q_t \in \{s_1, s_2, \dots, s_N\}$. Between each timestep, the next state is chosen randomly.

The current state determines the probability distribution for the next state.

Markov Property

q_{t+1} is conditionally independent of $\{q_{t-1}, q_{t-2}, \dots, q_1, q_0\}$ given q_t . In other words:

$$P(q_{t+1} = s_j | q_t = s_i) = P(q_{t+1} = s_j | q_t = s_i, \text{any earlier history})$$

An HMM, λ , is a 5-tuple consisting of

1. N — the number of states
2. M — the number of possible observations
3. $\{\mu_1, \mu_2, \dots, \mu_N\}$ the starting of possible observations $P(q_0 = S_i) = \mu_i$
4. the state transition probabilities $P(q_{t+1} = S_j | q_t = S_i) = a_{ij}$
5. The observation probabilities $P(O_t = k | q_t = S_i) = b_i(k)$

The Hidden Markov model is a probabilistic model which is used to explain or derive the probabilistic characteristic of any random process. It basically says that an observed event will not be corresponding to its step-by-step status but related to a set of probability distributions. Let's assume a system that is being modeled is assumed to be a Markov chain and in the process, there are some hidden states. In that case, we can say that hidden states are a process that depends on the main Markov process/chain.

The main goal of HMM is to learn about a Markov chain by observing its hidden states. Considering a Markov process X with hidden states Y here the HMM solidifies that for each time stamp the probability distribution of Y must not depend on the history of X according to that time.

2. What is the Viterbi algorithm and how does it work?

The Viterbi algorithm is a dynamic programming algorithm for obtaining the maximum a posteriori probability estimate of the most likely sequence of hidden states—called the Viterbi path—that results in a sequence of observed events, especially in the context of Markov information sources and hidden Markov models (HMM).

The Viterbi algorithm is used to find the most likely hidden state sequence an observable sequence, when the probability of a unobservable sequence can be decomposed into a product of probabilities. In theory, for a sequence of N words with K possible tags, we could compute all the probabilities of all K^N possible sequences, and find the max, but we'd have exponential running time in the length of the list. What the

Viterbi algorithm leverages is that the only operation in finding these probabilities is multiplication. Thus, for all paths ending in to a specific category (e.g. V), the path with greatest probability ending in V will bound above all other paths ending in V for all future expansions from V .

Put another way, take two possible paths $D N$ and $V N$, say with running probabilities 0.9 and 0.1, respectively, as analyses for the first two words of token sequence "the man ate the lasagna". No possible sequence of best tags (say $V D N$), can make $D V V D N$ more probable than $N V V D N$. Since we only care about the most likely tag sequence, with Viterbi, we only need to keep K possible analyses with running totals around at each step, because we know the most probable of the K final analyses will be greater than all other paths, even paths we didn't explicitly calculate. The result is an optimized algorithm with $N * (K^2)$ instead of K^N running time.

Making Simple Decisions

1. What is utility?

Utility function assigns a single number to express desirability of the state. It can be difficult to assign utility to every state.

$U(S)$: utility of state S according to decision making agent where state is complete snapshot of the whole world.

2. Explain expected utility.

$$\text{expected utility} = \text{utility} + \text{outcome probabilities}$$

It is the expected utility given agent takes and action for given state evidence. Action can have multiple results so you multiply the probability of that happening conditioned by action and evidence multiple by the utility of each result. Thus, expected utility of action A given evidence E

$$EU(A|E) = \sum_i P(\text{Result}_i(A) | Do(A), E) U(\text{Result}_i(A))$$

Principle of maximum expected utility (MEU)

A rational agent should choose an action that maximizes the agent's expected utility.

3. Why can't AI solve these problems?

Knowing the initial state of the world requires perception, learning, knowledge representation, and inference. Computations involved can be prohibitive, and it is

sometimes difficult even to formulate the problem completely. Computing $P(\text{Result}_i(A) | \text{Do}(A), E)$ requires a complete causal model of the world and, as we saw in, *NP-hard inference in Bayesian networks*. Computing the utility of each state, $U(\text{Result}_i(A))$, often requires searching or planning, because an agent does not know how good a state is until it knows where it can get to from that state. So, **decision theory is not a panacea that solves the AI problem.**

4. What is a lottery?

Let's first give some notations:

1. $A > B$ when A is preferred to B
2. $A \sim B$ when the agent is indifferent between A and B
3. $A \gtrsim B$ when agent is indifferent between A and B or A preferred to B

What are A and B?

1. if agent's actions are **deterministic**, A and B will typically be the **concrete, fully specified outcome states of those actions**.
2. in the more general, **nondeterministic** case, A and B will be **lotteries**.

The lottery is a probability distribution over the possible outcomes and their probabilities.

A lottery L with possible outcomes C_1, \dots, C_n that can occur with probabilities

p_1, \dots, p_n is written

$$L = [p_1, C_1; p_2, C_2; \dots; p_n, C_n]$$

A lottery with only one outcome can be written either as A or as [1, A]. In general, each outcome of a lottery can be either an **atomic state** or **another lottery**.

5. What are the constraints on making a rational preference?

1. Transitivity

Given any three states, if an agent prefers A to B and prefers B to C, then the agent must prefer A to C.

$$(A > B) \wedge (B > C) \Rightarrow (A > C)$$

2. Orderability

Given any two states, a rational agent must either prefer one to the other or else rate the two as equally preferable. That is, the agent cannot avoid deciding.

$$(A \prec B) \vee (B \prec A) \vee (A \sim B)$$

3. Continuity

If some state 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.

$$(A > B > C) \Rightarrow \exists p [p, A; 1 - p, C] \sim B$$

4. Substitutability

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. This holds regardless of the probabilities and the other outcome(s) in the lotteries.

$$A \sim B \Rightarrow [p, A; 1 - p, C] \sim [p, B; 1 - p, C]$$

5. Monotonicity

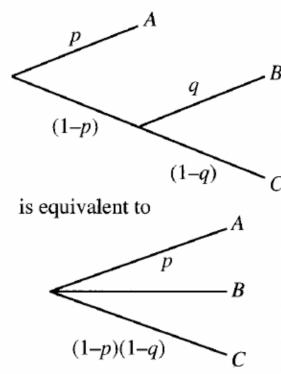
Suppose there are two lotteries that have the same two outcomes, A and B. If an agent prefers A to B, then the agent must prefer the lottery that has a higher probability for A (and vice versa).

$$A > B \Rightarrow (p \geq q \Leftrightarrow [p, A; 1 - p, B] \succsim [q, A; 1 - q, B])$$

6. Decomposability

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

$$[p, A; 1 - p, [q, B; 1 - q, C]] \sim [p, A; (1 - p)q, B; (1 - p)(1 - q), C]$$



Axioms of utility theory do not say anything about utility. They talk only about preferences.

Preference is assumed to be a basic property of rational agents.

6. Discuss important utility principles.

Utility principle

If an agent's preferences obey the axioms of utility, then there exists a real-valued function U that operates on states such that $U(A) > U(B)$ if and only if A is preferred to B , and $U(A) = U(B)$ if and only if the agent is indifferent between A and B .

$$U(A) > U(B) \Leftrightarrow A > B \text{ and } U(A) = U(B) \Leftrightarrow A \sim B$$

Maximum Expected Utility principle

The utility of a lottery is the sum of the probability of each outcome times the utility of that outcome.

$$U([p_1, S_1; \dots; p_n, S_n]) = \sum_i p_i U(S_i)$$

Once probabilities and utilities of possible outcome states are specified, the utility of a compound lottery involving those states is completely determined.

7. What is a utility function?

Utility is a function that maps from states to real numbers. An agent can have any preferences it likes even irrational one.

However, if utility functions were arbitrary, then utility theory would not be of much help because we would have to observe the agent's preferences in every possible combination of circumstances before being able to make any predictions about its behavior.

Economics provides one obvious candidate for a utility measure: **money**. Almost universal exchangeability of money for all kinds of goods and services suggests that money plays a significant role in human utility functions. If we restrict our attention to actions that only affect the amount of money that an agent has, then agent usually prefers more money to less. Agent exhibits a monotonic preference for definite amounts of money.

Expected monetary value is calculated by the sum over the number of items which are calculated by multiplying the amount and probability of that amount.

$$EMV = \sum_i amount_i P(amount_i)$$

The utility is not directly proportional to monetary value, because the utility for your first million is very high, whereas the utility for an additional million is much smaller.

8. How to estimate the nonlinearity of money?

In a pioneering study of actual utility functions, Grayson (1960) found that the utility of money was almost exactly **proportional to the logarithm of the amount**.

9. Map the graph and talk about risk aversion.

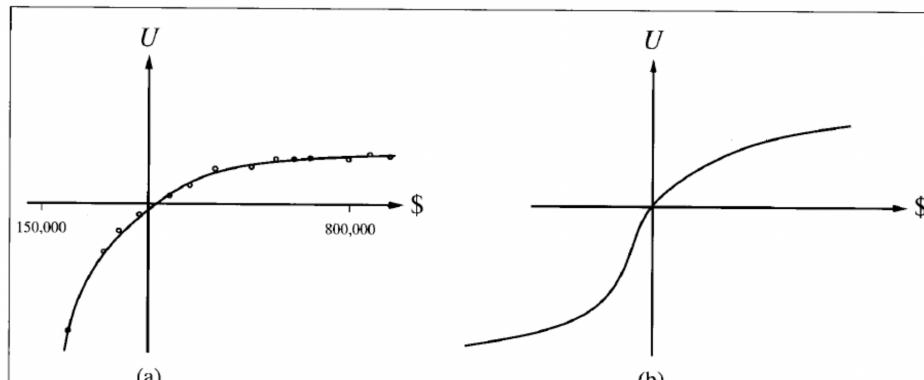


Figure 16.2 The utility of money. (a) Empirical data for Mr. Beard over a limited range. (b) A typical curve for the full range.

People operating in the region of the **curve with decreasing slope** are **risk-averse**, in other words, they **prefer a sure thing with a payoff that is less than the expected monetary value of a gamble**.

In the "desperate" region at large negative wealth in Figure 16.2(b), the behavior is **risk-seeking**.

The value an agent will accept in lieu of a lottery is called the **certainty equivalent of the lottery**.

Most people will accept about \$400 in lieu of a gamble that gives \$1000 half the time and \$0 the other half - that is, the certainty equivalent of the lottery is \$400.

The difference between the expected monetary value of a lottery and its certainty equivalent is called the **insurance premium**.

Risk aversion is the basis for the insurance industry because it means that **insurance premiums are positive**.

An agent that has a linear curve is said to be **risk-neutral**.

10. What are Multivariate Utility Functions?

Let's give an example of placing a new airport requires consideration of:

- the disruption caused by construction;
- the cost of land;
- the distance from centers of population;
- the noise of flight operations;
- safety issues arising from local topography and weather conditions;
- and so on.

Problems in which outcomes are characterized by two or more attributes, are handled by **multiatribute utility theory**.

We will call the attributes $X = X_1, \dots, X_n$ a complete vector of assignments will be $x = \langle x_1, \dots, x_n \rangle$. Each attribute is generally assumed to have discrete or continuous scalar values.

11. What is a dominance? What are the differences between Strict Dominance and Stochastic Dominance?

Dominance is basically power or influence of one thing over another. In our lectures, we have talked about two kinds of dominance: strict dominance and stochastic dominance. Let's give an example:

Suppose that airport site S_1 costs less, generates less noise pollution, and is safer than site S_2 . We can say that there is strict dominance of S_1 over S_2 .

In general, if an option is of lower value on all attributes than some other option, it need not be considered further. Strict dominance is often very useful in narrowing down the field of choices to the real contenders, but rarely yields a unique choice.

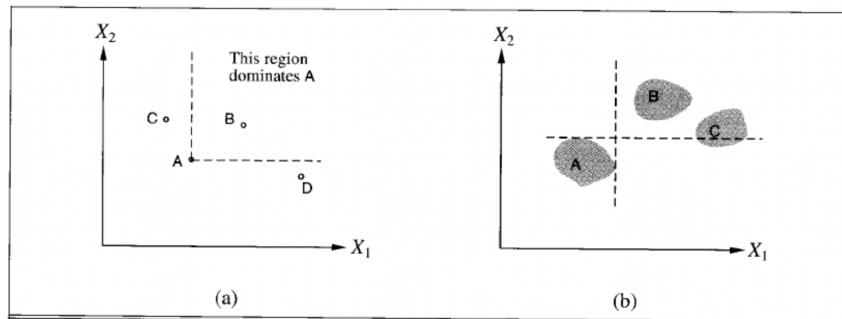


Figure 16.3 Strict dominance. (a) Deterministic: Option A is strictly dominated by B but not by C or D. (b) Uncertain: A is strictly dominated by B but not by C.

A direct analog of strict dominance can be constructed, where, despite the uncertainty, all possible concrete outcomes for S_1 strictly dominate all possible outcomes for S_2 .

Stochastic dominance is the type of dominance where the action outcomes are uncertain. Stochastic dominance condition might seem rather technical and perhaps not so easy to evaluate without extensive probability calculations.

12. Talk about preference without uncertainty.

Basic regularity that arises in deterministic preference structures is called preference independence. Two attributes X_1 and X_2 are preferentially independent of a third attribute X_3 if the preference between outcomes $\langle x_1, x_2, x_3 \rangle$ and $\langle x'_1, x'_2, x'_3 \rangle$ does not depend on the particular value x_3 for attribute X_3 .

Example: airport

- attributes to consider (among others): Noise, Cost, and Deaths
- one may propose that Noise and Cost are preferentially independent of Deaths.
- if we prefer a state with 20.000 people residing in the flight path and a construction cost of \$4 billion to a state with 70.000 people residing in the flight path and a cost of \$3,7 billion when the safety level is 0,06 deaths per million passenger miles in both cases, then we would have the same preference when the safety level is 0,13 or 0,01;
- and the same independence would hold for preferences between any other pair,

- It is also apparent that **Cost** and **Deaths** are preferentially independent of **Noise** and that **Noise** and **Deaths** are preferentially independent of **Cost**.

We say that the **set of attributes** {Noise, Cost, Deaths} exhibits ***mutual preferential independence (MPI)***. MPI says that, whereas each attribute may be important, it does not affect the way in which one trades off the other attributes against each other.

13. What are Decision Networks?

Decision networks combine **Bayesian networks** with additional node types for **actions** and **utilities** and represents information about:

- the agent's **current state**,
- its possible **actions**,
- the **state that will result from the agent's action**,
- the **utility** of that state.

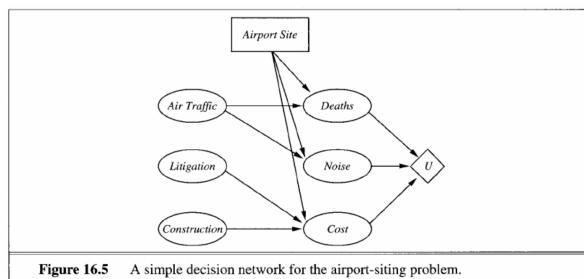


Figure 16.5 A simple decision network for the airport-siting problem.

There are three types of nodes in Decision Networks: chance nodes, decision nodes and utility nodes.

Chance nodes (ovals) represent random variables, just as they do in Bayes nets.

Each chance node has associated with it a **conditional distribution** that is indexed by the state of the parent nodes. In decision networks, the parent nodes can include decision nodes as well as chance nodes.

Decision nodes (rectangles) represent points where the decision-maker has a choice of actions.

In this case, the **AirportSite** action can take on a different value for each site under consideration. The choice influences the cost, safety, and noise that will result.

Utility nodes (diamonds) represent the agent's utility function.

The utility node has as parents all variables describing the outcome that directly affects utility. Associated with the utility node is a description of the agent's utility as a function of the parent attributes.

14. What is the value of information? What is the value of perfect information (VPI)?

Information value theory enables an agent to choose what information to acquire. The acquisition of information is achieved by sensing actions.

Usually, we assume that exact evidence is obtained about the value of some random variable E_j , so the phrase **value of perfect information (VPI)** is used. Let the agent's current knowledge be E . Then the value of the current best action a is defined by:

$$EU(a | E) = \max \sum_i U(Result_i(A))P(Result_i(A) | Do(A), E)$$

Making Complex Decisions

1. In the lecture slides the questions about an agent trying to find its path from start to end why don't we solve it as an AI search problem instead use the Markov decision process?

Because in AI search problems we cannot include the transition model of the environment. We assume the world is deterministic (the random state change can be found with certainty)

2. What is Markov Decision Process?

A sequential decision problem for a fully observable, stochastic environment with a Markovian transition model and additive rewards is called a **Markov decision process**, or **MDP**, and consists of a set of states (with an initial state s_0); a set $ACTIONS(s)$ of actions in each state; a transition model $T(s, a, s')$; and a reward function $R(s)$.

3. What are the three components of a Markov decision process?

There are three components of Markov decision processes which are: **initial state** s_0 , **transition model** $T(s, a, s')$, and **the reward function** $R(s)$. Rewards are additive.

3. Define the following terms :

Transition Model: it is the model that defines the outcome when a particular action is taken given that you are in a specific state. Basically, it describes the outcome of each

action in each state. $T(s, a, s')$ denotes the probability of ending up in state s' if action a is applied in state s . The transitions are Markovian: $T(s, a, s')$ depends only on s , not on the history of earlier states

Utility function: it is a specification about an agent's preferences. The utility function will depend on a sequence of states (this is a sequential decision problem, but still the transitions are Markovian)

Policy: It's one of the solutions for a Markov decision process, it defines what action to take given the state the agent is in. An optimal policy is one that maximizes the expected utility.

Optimal policy: optimal policy is a policy that yields the highest expected utility. We use

Finite horizon: an event that has a fixed time after which the game is over. Thus, it means that there is a fixed time N after which nothing matters—the game is over.

Infinite horizon: an event that has no end to the game.

Stationary preference: If an agent's preference between states sequences does not change with time it is said to have a stationary preference.

Nonstationary preference: we say that the optimal policy for a finite horizon is nonstationary, i.e., it could change over time.

4. What is a discount factor?

Stationarity is a fairly innocuous-looking assumption with very strong consequences: it turns out that under stationarity there are just two coherent ways to assign utilities to sequences:

1. **Additive rewards:** The utility of a state sequence is

$$U_h([s_0, s_1, s_2, \dots]) = R(s_0) + R(s_1) + R(s_2) + \dots,$$

2. **Discounted rewards:** The utility of a state sequence is

$$U_h([s_0, s_1, s_2, \dots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots,$$

where the discount factor γ is a number between 0 and 1. The discount factor describes the preference of an agent for current rewards over future rewards. When γ is close to 0, rewards in the distant future are viewed as insignificant. When γ is 1, discounted rewards are exactly equivalent to additive rewards, so additive rewards are a special case of discounted rewards. Discounting appears to be a good model of both animal and human preferences over time.

5. What are the two algorithms (taught in class) that are used to solve Markov decisions processes and describe them?

1. Value iteration

In the value iteration algorithm, we first need to initialize the value of each state to its immediate reward. Then, we need to iterate to calculate values considering sequential rewards. For each state, we then need to select the action with the maximum expected utility.

That is, the utility of a state is given by:

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' | s, a) U(s')$$

This is called the Bellman equation, after Richard Bellman (1957).

The Bellman equation is the basis of the value iteration algorithm for solving MDPs. If there are n possible states, then there are n Bellman equations, one for each state. The n equations contain n unknowns—the utilities of the states. So we would like to solve these simultaneous equations to find the utilities. There is one problem: the equations are nonlinear because the “max” operator is not a linear operator. Whereas systems of linear equations can be solved quickly using linear algebra techniques, systems of nonlinear equations are more problematic. One thing to try is an iterative approach. We start with arbitrary initial values for the utilities, calculate the right-hand side of the equation, and plug it into the left-hand side—thereby updating the utility of each state from the utilities of its neighbors. We repeat this until we reach an equilibrium. Let $U_i(s)$ be the utility value for state s at the i th iteration. The iteration step, called a Bellman update, looks like this:

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' | s, a) U_i(s')$$

```

function VALUE-ITERATION(mdp,  $\epsilon$ ) returns a utility function
  inputs: mdp, an MDP with states  $S$ , actions  $A(s)$ , transition model  $P(s' | s, a)$ ,
           rewards  $R(s)$ , discount  $\gamma$ 
   $\epsilon$ , the maximum error allowed in the utility of any state
  local variables:  $U$ ,  $U'$ , vectors of utilities for states in  $S$ , initially zero
                      $\delta$ , the maximum change in the utility of any state in an iteration

  repeat
     $U \leftarrow U'$ ;  $\delta \leftarrow 0$ 
    for each state  $s$  in  $S$  do
       $U'[s] \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' | s, a) U[s']$ 
      if  $|U'[s] - U[s]| > \epsilon$  then  $\delta \leftarrow |U'[s] - U[s]|$ 
    until  $\delta < \epsilon(1 - \gamma)/\gamma$ 
  return  $U$ 

```

Figure 17.4 The value iteration algorithm for calculating utilities of states. The termination condition is from Equation (17.8).

Shortly, to find a value iteration:

- initialize the value of each state to its immediate reward,
- iterate to calculate values considering sequential rewards,
- for each state, select the action with the maximum expected utility

2. Policy iteration

The policy iteration algorithm alternates the following two steps, beginning from some initial policy π_0 :

- Policy evaluation: given a policy μ_i , calculate $U_i = U^{\mu_i}$, the utility of each state if π_i were to be executed.
- Policy improvement: calculate a new MEU policy π_{i+1} , using a one-step look-ahead based on U_i .

The algorithm terminates when the policy improvement step yields no change in the utilities. At this point, we know that the utility function U_i is a fixed point of the Bellman update, so it is a solution to the Bellman equations, and μ_i must be an optimal policy. Because there are only finitely many policies for a finite state space, and each iteration can be shown to yield a better policy, policy iteration must terminate.

```

function POLICY-ITERATION(mdp) returns a policy
  inputs: mdp, an MDP with states S, actions A(s), transition model  $P(s' | s, a)$ 
  local variables: U, a vector of utilities for states in S, initially zero
     $\pi$ , a policy vector indexed by state, initially random

  repeat
    U  $\leftarrow$  POLICY-EVALUATION( $\pi$ , U, mdp)
    unchanged?  $\leftarrow$  true
    for each state s in S do
      if  $\max_{a \in A(s)} \sum_{s'} P(s' | s, a) U[s'] > \sum_{s'} P(s' | s, \pi[s]) U[s']$  then do
         $\pi[s] \leftarrow \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s' | s, a) U[s']$ 
      unchanged?  $\leftarrow$  false
    until unchanged?
  return  $\pi$ 

```

Figure 17.7 The policy iteration algorithm for calculating an optimal policy.

To sum up: for a policy iteration:

- get an **initial policy**
- **evaluate the policy** to find the **utility of each state**
- **modify the policy** by **selecting** actions that **increase the utility of a state**. If **changes occurred**, go to the previous step