Uncertainty

Why is uncertainty important?

- Agents (and humans) don’t know everything,
- but need to make decisions anyways!
- Decisions are made in the absence of information,
- or in the presence of noisy information (sensor readings)

The best an agent can do:
know how uncertain it is, and act accordingly

Probability: Frequentist vs. Bayesian

Frequentist view:
probability of heads = # of heads / # of flips

probability of heads this time = probability of heads (history)

Uncertainty is ontological: pertaining to the world

Bayesian view:
probability of heads this time = agent’s belief about this event
belief of agent A : based on previous experience of agent A

Uncertainty is epistemological: pertaining to knowledge

Bayesian probability
all else being equal:
before 2 flips

Bayesian probability
all else being equal:
after 2 flips heads, heads

Bayesian probability
all else being equal:
after 2 flips tails, tails
Should you wear your seatbelt? estimate $P(\text{fatality})$ given you do/don’t wear it.

Frequentist:

<table>
<thead>
<tr>
<th>day</th>
<th>result</th>
<th>$P(\text{fatality})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunday (prior to start)</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>Monday</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>Tuesday</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>Tuesday</td>
<td>0.33333</td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>Friday</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
| N                    | Number of crashes / N |}

Bayesian:

Features

Describe the world in terms of a set of states: $\{s_1, s_2, \ldots, s_N\}$

or, as the product of a set of features (also known as attributes or random variables)

- Number of states $= 2^{\text{number of binary features}}$
- Features describe the state space in a factored form.
- state $\rightarrow$ factorize $\rightarrow$ feature values
- feature values $\rightarrow$ cross product $\rightarrow$ states

Probability Measure

if $X$ is a random variable (feature, attribute), it can take on values $x$, where $x \in \text{Domain}(X)$
Assume $x$ is discrete

$P(x)$ is the probability that $X = x$

joint probability $P(x, y)$ is the probability that $X = x$ and $Y = y$ at the same time

Joint probability distribution:

Axioms of Probability

Axioms are things we have to assume about probability:

- $P(X) \geq 0$
- $\sum_x P(X = x) = 1.0$
- $P(a \lor b) = P(a) + P(b)$ if $a$ and $b$ are contradictory - can’t both be true at the same time e.g. $P(\text{win} \lor \text{lose}) = P(\text{win}) + P(\text{lose}) = 1.0$

Some notes:

- probability between 0-1 is purely convention
- $P(a) = 0$ means $a$ is definitely false
- $P(a) = 1$ means $a$ is definitely true
- $0 < P(a) < 1$ means you have belief about the truth of $a$. It does not mean that $a$ is true to some degree, just that you are ignorant of its truth value.
- Probability = measure of ignorance
Independence

- describe a system with \( n \) features: \( 2^n - 1 \) probabilities
- Use **independence** to reduce number of probabilities
  - e.g. radially symmetric dartboard, \( P(\text{hit a sector}) \)
  - \( P(\text{sector}) = P(r, \theta) \) where \( r = 1, \ldots, 4 \) and \( \theta = 1, \ldots, 8 \).
- 32 sectors in total - need to give 31 numbers

Independence - Example

- 4 independent Boolean variables, \( X_1, X_2, X_3, X_4 \), where \( x_i \) means \( X_i = \text{true} \), and \( \overline{x_i} \) means \( X_i = \text{false} \)
- \( P(x_1) = 0.4, P(x_2) = 0.2, P(x_3) = 0.5, P(x_4) = 0.8 \)
- would usually need 16 numbers to specify the joint probability distribution, but if they are all independent, we just need 4:
  \[
P(x_1, \overline{x_2}, x_3, x_4) = P(x_1)(1 - P(x_2))P(x_3)P(x_4)
  = (0.4)(0.8)(0.5)(0.8)
  = 0.128
\]
  \[
P(x_1, x_2, x_3|x_4) = P(x_1)P(x_2)P(x_3)
  = (0.4)(0.2)(0.5)
  = 0.04
\]

Conditional Probability

if \( X \) and \( Y \) are random variables, then

\( P(x|y) \) is the probability that \( X = x \) given that \( Y = y \).

e.g.

\( P(\text{flies}|\text{is bird}) \) is different than \( P(\text{flies}) \)

\( P(\text{flies}|\text{is a penguin, is bird}) \) is different again

**incorporate independence:**

\( P(\text{flies}|\text{is bird, has feathers}) = P(\text{flies}|\text{is bird}) \)

**Product rule (Chain rule):**

\[
P(\text{flies, is bird}) = P(\text{flies}|\text{is bird})P(\text{is bird})
P(\text{flies, is bird}) = P(\text{is bird}|\text{flies})P(\text{flies})
\]

**leads to:** **Bayes’ rule**

\[
P(\text{is bird}|\text{flies}) = \frac{P(\text{flies}|\text{is bird})P(\text{is bird})}{P(\text{flies})}
\]

Sum Rule

We know (an Axiom):

\[
\sum_x P(X = x) = 1.0 \quad \text{and therefore that} \quad \sum_x P(X = x|Y) = 1.0
\]

This means that

\[
\sum_x P(X = x, Y) = P(Y)
\]

proof:

\[
\sum_x P(X = x, Y) = \sum_x P(X = x|Y)P(Y)
= P(Y) \sum_x P(X = x|Y)
= P(Y)
\]

We call \( P(Y) \) the **marginal** distribution over \( Y \).
**Conditional Probability**

- **X and Y are independent iff**
  
  \[ P(X) = P(X|Y) \]
  
  \[ P(Y) = P(Y|X) \]
  
  \[ P(X,Y) = P(X)P(Y) \]
  
  so learning Y doesn’t influence beliefs about X
- **X and Y are conditionally independent given Z iff**
  
  \[ P(X|Z) = P(X|Y,Z) \]
  
  \[ P(Y|Z) = P(Y|X,Z) \]
  
  \[ P(X,Y|Z) = P(X|Z)P(Y|Z) \]
  
  so learning Y doesn’t influence beliefs about X if you already know Z...does not mean X and Y are independent

**Belief Networks**

- **Bayesian network or belief network**
  - Directed Acyclic graph
  - Encodes independencies in a graphical format
  - Edges give \( P(X_i|\text{parents}(X_i)) \)

**Bayesian networks - example**

If Jesse’s alarm doesn’t go off (A), Jesse probably won’t get coffee (C); if Jesse doesn’t get coffee, he’s likely grumpy (G). If he is grumpy then it’s possible that the lecture won’t go smoothly L. If the lecture does not go smoothly then the students will likely be sad S.

\[ A \rightarrow C \rightarrow G \rightarrow L \rightarrow S \]

A=Jesse’s alarm doesn’t go off
C=Jesse doesn’t get coffee
G=Jesse is grumpy
L=lecture doesn’t go smoothly
S=students are sad

**Expected Values**

- expected value of a function on X, \( V(X) \):
  
  \[ E(V) = \sum_{x\in\text{Dom}(X)} P(x)V(x) \]
  
  where \( P(x) \) is the probability that \( X = x \).

This is useful in decision making, where \( V(X) \) is the utility of situation X.

Bayesian decision making is then

\[ E( V(\text{decision})) = \sum_{\text{outcome}} P(\text{outcome}|\text{decision})V(\text{outcome}) \]

**Value of Independence**

- complete independence reduces both representation and inference from \( O(2^n) \) to \( O(n) \)
- Unfortunately, complete mutual independence is rare
- Fortunately, most domains do exhibit a fair amount of conditional independence
- Bayesian Networks or Belief Networks (BNs) encode this information

**Correlation and Causality**

- Directed links in Bayes’ net \( \approx \) causal
- However, not always the case: chocolate \( \rightarrow \) Nobel or Nobel \( \rightarrow \) chocolate?
- In a Bayes net, it doesn’t matter!
- But, some structures will be easier to specify

In this example, it’s probably chocolate \( \leftarrow \) “Switzerland – ness” \( \rightarrow \) Nobel
Bayesian Networks

A Bayesian Network (Belief Network, Probabilistic Network) or BN over variables \{X_1, X_2, \ldots, X_N\} consists of:

- a DAG whose nodes are the variables
- a set of Conditional Probability tables (CPTs) giving \(P(X_i | \text{Parents}(X_i))\) for each \(X_i\)

**Inference is Easy**

Want to know \(P(g)\)? - Use sum rule to sum over parents of \(g\)

\[
P(g) = \sum_{c_i \in \text{Dom}(C)} P(g, c_i)
\]

\[
= \sum_{c_i \in \text{Dom}(C)} P(g|c_i)P(c_i)
\]

\[
= \sum_{c_i \in \text{Dom}(C)} P(g|c_i) \sum_{a_j \in \text{Dom}(A)} P(c_i|a_j)P(a_j)
\]

\[
= P(g|c[P(c|a)P(a) + P(c|\overline{a})P(\overline{a})]
\]

\[
+ P(g|\overline{c})[P(\overline{c}|a)P(a) + P(\overline{c}|\overline{a})P(\overline{a})]
\]

\[
= 1.0 \times [0.8 \times 0.3 + 0.15 \times 0.7] + 0.2 \times [0.2 \times 0.3 + 0.85 \times 0.7]
\]

\[
= 0.4760
\]
The structure of the BN means that:

\[ P(X_i|S, Parents(X_i)) = P(X_i|Parents(X_i)) \]

for any subset \( S \subseteq NonDescendants(X_i) \)

The BN defines a factorization of the joint probability distribution. The joint distribution is formed by multiplying the conditional probability tables together.

To represent a domain in a belief network, you need to consider:

- What are the relevant variables?
  - What will you observe? - this is the evidence
  - What would you like to find out? - this is the query
  - What other features make the model simpler? - these are the other variables
- What values should these variables take?
- What is the relationship between them? This should be expressed in terms of local influence.
- How does the value of each variable depend on its parents? This is expressed in terms of the conditional probabilities.

Test B depends on Cancer and Malfunction
Test A depends only on Cancer
Report depends only on Test B
Database depends only on Report

Database and Test B independent if Report is observed
Test B and Test A are independent if Cancer is observed
Malfunction and Cancer are independent if Test B is not observed.

Testing Independence

Given a BN, how do we determine if two variables X,Y are independent (given evidence E)?

**D-separation**: A set of variables $E$ d-separates X and Y if it blocks every undirected path in the BN between X and Y.

But what does “block” mean?

Markov Blanket

The **Markov Blanket** of a node $v$ is:

- the parents, children, and the (other) parents of children
- the minimal set of nodes that d-separates $v$ from all other variables

The joint distribution over the **Markov Blanket** allows for the computation of the distribution of $v$. 
D-Separations: Example

- TravelSubway and Thermometer (given no evidence)?
- TravelSubway and Thermometer (given Flu or Fever)?
- TravelSubway and Malaria (given Fever)?
- TravelSubway and Exotic Trip (given Jaundice)?
- TravelSubway and Exotic Trip (given Jaundice and Thermometer)?
- TravelSubway and Exotic Trip (given Malaria and Thermometer)?

Agent has a prior belief in a hypothesis, $h$, $P(h)$,

Agent observes some evidence $e$ that has a likelihood given the hypothesis: $P(e|h)$.

The agent's posterior belief about $h$ after observing $e$, $P(h|e)$, is given by Bayes' Rule:

$$P(h|e) = \frac{P(e|h)P(h)}{P(e)} = \frac{P(e|h)P(h)}{\sum_h P(e|h)P(h)}$$

Why is Bayes' theorem interesting?

- Often you have causal knowledge:
  - $P($ symptom | disease $)$
  - $P($ light is off | status of switches and switch positions $)$
  - $P($ alarm | fire $)$
  - $P($ image looks like 🌲 | a tree is in front of a car $)$

- and want to do evidential reasoning:
  - $P($ disease | symptom $)$
  - $P($ status of switches | light is off and switch positions $)$
  - $P($ fire | alarm $)$.

$P($ a tree is in front of a car | image looks like 🌲 $)$

Before you get any information

- $P($ Cancer $) = 0.32$
- $P($ Malfunction $) = 0.08$

Suppose the doctor reads a positive Test B in the Database so the evidence gives Database=true (not directly Test B= true) we want to know $P($Cancer = true$|$Database = true$)$

$P($Cancer = true$|$Database = true$) = 0.80$

$P($Malfunction = true$|$Database = true$) = 0.14$

(we will see how to get these numbers later)
Suppose Test A is negative, though! we want \( P(\text{Cancer} = \text{true}|\text{Database} = \text{true} \land \text{TestA} = \text{false}) \)

- \( P(\text{Cancer} = \text{true}|\text{Database} = \text{true} \land \text{TestA} = \text{false}) = 0.48 \)
- \( P(\text{M} = \text{true}|\text{Database} = \text{true} \land \text{TestA} = \text{false}) = 0.27 \)

(we will see how to get these numbers later)

When evidence is downstream of query, then we must reason "backwards". This requires Bayes' rule.

\[
\begin{align*}
P(\text{ET} | \text{j}) & \propto P(\text{j}, \text{ET}) \\
& = P(\text{j} | \text{ET}) P(\text{ET}) \quad \text{(Bayes' rule)} \\
& = \sum_{\text{M}} P(\text{j} | \text{M} \land \text{ET}) P(\text{M} | \text{ET}) P(\text{ET}) \\
& = \sum_{\text{M}} P(\text{j} | \text{M} \land \text{ET}) P(\text{M} | \text{ET}) P(\text{ET}) \\
& = \sum_{\text{M}} P(\text{j} | \text{M}) P(\text{M} \land \text{ET}) P(\text{ET}) \\
& = \sum_{\text{M}} P(\text{j} | \text{M}) P(\text{M} | \text{ET}) P(\text{ET}) \\
\end{align*}
\]

normalising constant is \( \frac{1}{P(\text{j})} \), but this can be computed as

\[
P(\text{j}) = \sum_{\text{ET}} P(\text{ET}, \text{j})
\]
Backward Inference

F: Bridge on Fire
C: All friends Crazy
J: All friends Jump
What is \( P(F|J = \text{true}) \)?

| F | C | P(J = \text{true}|F,C) |
|---|---|------------------|
| t | t | 0.95 |
| t | f | 0.99 |
| f | t | 0.99 |
| f | f | 0.01 |

Variable Elimination

- intuitions above: polytree algorithm
- works for simple networks without loops
- more general algorithm: \text{Variable Elimination}
- applies sum-out rule repeatedly
- distributes sums

Multiplying factors example

The product of factor \( f_1(X, Y) \) and \( f_2(Y, Z) \), where \( Y \) are the variables in common, is the factor \((f_1 \times f_2)(X, Y, Z)\) defined by:

\[(f_1 \times f_2)(X, Y, Z) = f_1(X, Y)f_2(Y, Z).\]
Summing out variables

We can sum out a variable, say $X_1$ with domain \{ $v_1, \ldots, v_k$ \}, from factor $f(X_1, \ldots, X_j)$, resulting in a factor on $X_2, \ldots, X_j$ defined by:

$$
\left( \sum_{X_1} f(X_2, \ldots, X_j) \right) = f(X_1 = v_1, \ldots, X_j) + \cdots + f(X_1 = v_k, \ldots, X_j)
$$

Evidence

If we want to compute the posterior probability of $Z$ given evidence $Y_1 = v_1 \land \ldots \land Y_j = v_j$:

$$
P(Z | Y_1 = v_1, \ldots, Y_j = v_j) = \frac{P(Z, Y_1 = v_1, \ldots, Y_j = v_j)}{P(Y_1 = v_1, \ldots, Y_j = v_j)}
$$

Suppose the variables of the belief network are $X_1, \ldots, X_n$. To compute $P(Z, Y_1 = v_1, \ldots, Y_j = v_j)$, we sum out the other variables, $Z_1, \ldots, Z_k = \{X_1, \ldots, X_n\} - \{Z\} - \{Y_1, \ldots, Y_j\}$. We order the $Z_j$ into an elimination ordering.

$$
P(Z, Y_1 = v_1, \ldots, Y_j = v_j) = \sum_{Z_k} \cdots \sum_{Z_1} \prod_{i=1}^{n} P(X_i | \text{parents}(X_i)) Y_i = v_i, \ldots, Y_j = v_j.
$$

Computing sums of products

Computation in belief networks reduces to computing the sums of products.

- How can we compute $ab + ac$ efficiently?

Computing sums of products

Computation in belief networks reduces to computing the sums of products.

- How can we compute $ab + ac$ efficiently?
- Distribute out the $a$ giving $a(b + c)$
Computation in belief networks reduces to computing the sums of products.

- How can we compute $ab + ac$ efficiently?
- Distribute out the $a$ giving $a(b + c)$
- How can we compute $\sum Z_1 \prod_{i=1}^n P(X_i|\text{parents}(X_i))$ efficiently?

Notes on VE

Complexity is linear in number of variables, and exponential in the size of the largest factor

When we create new factors: sometimes this blows up

Depends on the elimination ordering

For polytrees: work outside in

For general BNs this can be hard

simply finding the optimal elimination ordering is NP-hard for general BNs

inference in general is NP-hard
Variable Ordering: Polytrees

- Eliminate "singly-connected" nodes \((D, A, C, X_1, \ldots, X_k)\) first
- Then no factor is ever larger than original CPTs
- If you eliminate \(B\) first, a factor is created that includes \(A, C, X_1, \ldots, X_k\)
- E.g.

\[
P(D|X_1) = \sum_{A, C} P(D|A)P(A)P(C) \sum_{B, X_2, \ldots, X_k} P(B|A, C)P(x_1|B) \prod_{j=2}^{k} P(X_j|B)
\]

Example II

Other Representations for Probability distributions

- Decision Tree:
- Noisy Or: \(P(x|Y_1, \ldots, Y_k)\) where \(X\) is caused by parents \(Y\)
- Logistic Regression where parents \(Y\) caused by \(X\).

\[
P(x|Y_1, \ldots, Y_k) = \text{sigmoid}(\sum_i w_i Y_i)
\]

Next:
- Reasoning under Uncertainty Part II (Poole & Mackworth (2nd ed.) Chapter 8.5-8.9)