Chapter 6: Reasoning under Uncertainty

"The mind is a neural computer, fitted by natural selection with combinatorial algorithms for causal and probabilistic reasoning about plants, animals, objects, and people.

"In a universe with any regularities at all, decisions informed about the past are better than decisions made at random. That has always been true, and we would expect organisms, especially informavores such as humans, to have evolved acute intuitions about probability. The founders of probability, like the founders of logic, assumed they were just formalizing common sense."

Steven Pinker, How the Mind Works, 1997, pp. 524, 343.

Learning Objectives

At the end of the class you should be able to:

- justify the use and semantics of probability
- know how to compute marginals and apply Bayes' theorem
- build a belief network for a domain
- predict the inferences for a belief network
- explain the predictions of a causal model

Using Uncertain Knowledge

- Agents don't have complete knowledge about the world.
- Agents need to make decisions based on their uncertainty.
- It isn't enough to assume what the world is like.
  Example: wearing a seat belt.
- An agent needs to reason about its uncertainty.

Why Probability?

- There is lots of uncertainty about the world, but agents still need to act.
- Predictions are needed to decide what to do:
  - definitive predictions: you will be run over tomorrow
  - point probabilities: probability you will be run over tomorrow is 0.002
  - probability ranges: you will be run over with probability in range [0.001,0.34]
- Acting is gambling: agents who don't use probabilities will lose to those who do — Dutch books.
- Probabilities can be learned from data.
  Bayes' rule specifies how to combine data and prior knowledge.

Probability

- Probability is an agent's measure of belief in some proposition — subjective probability.
- An agent's belief depends on its prior assumptions and what the agent observes.

Numerical Measures of Belief

- Belief in proposition, f, can be measured in terms of a number between 0 and 1 — this is the probability of f.
  - The probability f is 0 means that f is believed to be definitely false.
  - The probability f is 1 means that f is believed to be definitely true.
- Using 0 and 1 is purely a convention.
- f has a probability between 0 and 1, means the agent is ignorant of its truth value.
- Probability is a measure of an agent's ignorance.
- Probability is *not* a measure of degree of truth.

Random Variables

- A random variable is a term in a language that can take one of a number of different values.
- The range of a variable X, written $range(X)$, is the set of values X can take.
- A tuple of random variables $\langle X_1, \ldots, X_n \rangle$ is a complex random variable with range $range(X_1) \times \cdots \times range(X_n)$. Often the tuple is written as $X_1, \ldots, X_n$.
- Assignment $X = x$ means variable X has value x.
- A proposition is a Boolean formula made from assignments of values to variables.

Possible World Semantics

- A possible world specifies an assignment of one value to each random variable.
- A random variable is a function from possible worlds into the range of the random variable.
- $\omega \models X = x$
  means variable X is assigned value x in world ω.
- Logical connectives have their standard meaning:

  $\omega \models \alpha \wedge \beta$ if $\omega \models \alpha$ and $\omega \models \beta$

  $\omega \models \alpha \vee \beta$ if $\omega \models \alpha$ or $\omega \models \beta$

  $\omega \models \neg\alpha$ if $\omega \not\models \alpha$
- Let Ω be the set of all possible worlds.

Semantics of Probability

For a finite number of possible worlds:

- Define a nonnegative measure $\mu(\omega)$ to each world ω so that the measures of the possible worlds sum to 1.
- The **probability** of proposition f is defined by:

$$P(f) = \sum_{\omega \models f} \mu(\omega).$$

Axioms of Probability: finite case

Three axioms define what follows from a set of probabilities:

Axiom 1 $0 \leq P(a)$ for any proposition a.

Axiom 2 $P(\mathit{true}) = 1$

Axiom 3 $P(a \vee b) = P(a) + P(b)$ if a and b cannot both be true.

- These axioms are sound and complete with respect to the semantics.

Semantics of Probability: general case

In the general case, probability defines a measure on sets of possible worlds. We define $\mu(S)$ for some sets $S \subseteq \Omega$ satisfying:

- $\mu(S) \geq 0$
- $\mu(\Omega) = 1$
- $\mu(S_1 \cup S_2) = \mu(S_1) + \mu(S_2)$ if $S_1 \cap S_2 = \{\}$.
  Or sometimes σ-additivity:

$$\mu(\bigcup_i S_i) = \sum_i \mu(S_i) \text{ if } S_i \cap S_j = \{\} \text{ for } i \neq j$$

Then $P(\alpha) = \mu(\{\omega | \omega \models \alpha\})$.

Probability Distributions

- A probability distribution on a random variable X is a function $range(X) \to [0, 1]$ such that

$$x \mapsto P(X = x).$$

  This is written as $P(X)$.
- This also includes the case where we have tuples of variables. E.g., $P(X, Y, Z)$ means $P(\langle X, Y, Z\rangle)$.
- When $range(X)$ is infinite sometimes we need a probability density function...

Conditioning

- Probabilistic conditioning specifies how to revise beliefs based on new information.
- An agent builds a probabilistic model taking all background information into account. This gives the prior probability.
- All other information must be conditioned on.
- If evidence e is all the information obtained subsequently, the conditional probability $P(h|e)$ of h given e is the posterior probability of h.

Semantics of Conditional Probability

- Evidence e rules out possible worlds incompatible with e.
- Evidence e induces a new measure, μ_e, over possible worlds

$$\mu_e(S) = \begin{cases} c \times \mu(S) & \text{if } \omega \models e \text{ for all } \omega \in S \\ 0 & \text{if } \omega \not\models e \text{ for some } \omega \in S \end{cases}$$

We can show $c =$

Semantics of Conditional Probability

- Evidence e rules out possible worlds incompatible with e.
- Evidence e induces a new measure, μ_e, over possible worlds

$$\mu_e(S) = \begin{cases} c \times \mu(S) & \text{if } \omega \models e \text{ for all } \omega \in S \\ 0 & \text{if } \omega \not\models e \text{ for some } \omega \in S \end{cases}$$

  We can show $c = \frac{1}{P(e)}$.
- The conditional probability of formula h given evidence e is

$$\begin{aligned} P(h|e) &= \mu_e(\{\omega : \omega \models h\}) \\ &= \end{aligned}$$

Semantics of Conditional Probability

- Evidence e rules out possible worlds incompatible with e.
- Evidence e induces a new measure, μ_e, over possible worlds

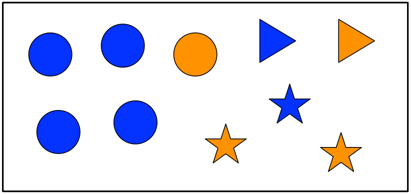
$$\mu_e(S) = \begin{cases} c \times \mu(S) & \text{if } \omega \models e \text{ for all } \omega \in S \\ 0 & \text{if } \omega \not\models e \text{ for some } \omega \in S \end{cases}$$

  We can show $c = \frac{1}{P(e)}$.
- The conditional probability of formula h given evidence e is

$$\begin{aligned} P(h|e) &= \mu_e(\{\omega : \omega \models h\}) \\ &= \frac{P(h \wedge e)}{P(e)} \end{aligned}$$

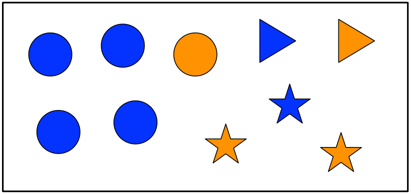
Conditioning

Possible Worlds:

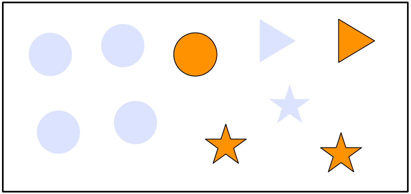


Conditioning

Possible Worlds:



Observe *Color* $=$ *orange*:



Exercise

Flu	*Sneeze*	*Snore*	μ
true	true	true	0.064
true	true	false	0.096
true	false	true	0.016
true	false	false	0.024
false	true	true	0.096
false	true	false	0.144
false	false	true	0.224
false	false	false	0.336

What is:

(a) $P(flu \wedge sneeze)$

(b) $P(flu \wedge \neg sneeze)$

(c) $P(flu)$

(d) $P(sneeze \mid flu)$

(e) $P(\neg flu \wedge sneeze)$

(f) $P(flu \mid sneeze)$

(g) $P(sneeze \mid flu \wedge snore)$

(h) $P(flu \mid sneeze \wedge snore)$

Chain Rule

$$P(f_1 \wedge f_2 \wedge \ldots \wedge f_n)$$

$$=$$

Chain Rule

$$
\begin{aligned}
& P(f_1 \wedge f_2 \wedge \ldots \wedge f_n) \\
& \quad = \quad P(f_n | f_1 \wedge \cdots \wedge f_{n-1}) \times \\
& \qquad\quad\; P(f_1 \wedge \cdots \wedge f_{n-1}) \\
& \quad =
\end{aligned}
$$

Chain Rule

$$
\begin{aligned}
& P(f_1 \wedge f_2 \wedge \ldots \wedge f_n) \\
& \quad = \quad P(f_n | f_1 \wedge \cdots \wedge f_{n-1}) \times \\
& \qquad\quad\; P(f_1 \wedge \cdots \wedge f_{n-1}) \\
& \quad = \quad P(f_n | f_1 \wedge \cdots \wedge f_{n-1}) \times \\
& \qquad\quad\; P(f_{n-1} | f_1 \wedge \cdots \wedge f_{n-2}) \times \\
& \qquad\quad\; P(f_1 \wedge \cdots \wedge f_{n-2}) \\
& \quad = \quad P(f_n | f_1 \wedge \cdots \wedge f_{n-1}) \times \\
& \qquad\quad\; P(f_{n-1} | f_1 \wedge \cdots \wedge f_{n-2}) \\
& \qquad\quad\; \times \cdots \times P(f_3 | f_1 \wedge f_2) \times P(f_2 | f_1) \times P(f_1) \\
& \quad = \quad \prod_{i=1}^{n} P(f_i | f_1 \wedge \cdots \wedge f_{i-1})
\end{aligned}
$$

Bayes' theorem

The chain rule and commutativity of conjunction ($h \wedge e$ is equivalent to $e \wedge h$) gives us:

$$P(h \wedge e) \quad =$$

Bayes' theorem

The chain rule and commutativity of conjunction ($h \wedge e$ is equivalent to $e \wedge h$) gives us:

$$P(h \wedge e) = P(h|e) \times P(e)$$

Bayes' theorem

The chain rule and commutativity of conjunction ($h \wedge e$ is equivalent to $e \wedge h$) gives us:

$$\begin{aligned} P(h \wedge e) &= P(h|e) \times P(e) \\ &= P(e|h) \times P(h). \end{aligned}$$

Bayes' theorem

The chain rule and commutativity of conjunction ($h \wedge e$ is equivalent to $e \wedge h$) gives us:

$$
\begin{aligned}
P(h \wedge e) &= P(h|e) \times P(e) \\
&= P(e|h) \times P(h).
\end{aligned}
$$

If $P(e) \neq 0$, divide the right hand sides by $P(e)$:

$$P(h|e) =$$

Bayes' theorem

The chain rule and commutativity of conjunction ($h \wedge e$ is equivalent to $e \wedge h$) gives us:

$$\begin{aligned} P(h \wedge e) &= P(h|e) \times P(e) \\ &= P(e|h) \times P(h). \end{aligned}$$

If $P(e) \neq 0$, divide the right hand sides by $P(e)$:

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

This is Bayes' theorem.

Why is Bayes' theorem interesting?

- Often you have causal knowledge:
  $P(\mathit{symptom} \mid \mathit{disease})$
  $P(\mathit{light\ is\ off} \mid \mathit{status\ of\ switches\ and\ switch\ positions})$
  $P(\mathit{alarm} \mid \mathit{fire})$
  $P($*image looks like*  $\mid$ *a tree is in front of a car*$)$
- and want to do evidential reasoning:
  $P(\mathit{disease} \mid \mathit{symptom})$
  $P(\mathit{status\ of\ switches} \mid \mathit{light\ is\ off\ and\ switch\ positions})$
  $P(\mathit{fire} \mid \mathit{alarm})$.
  $P($*a tree is in front of a car* $\mid$ *image looks like* $)$

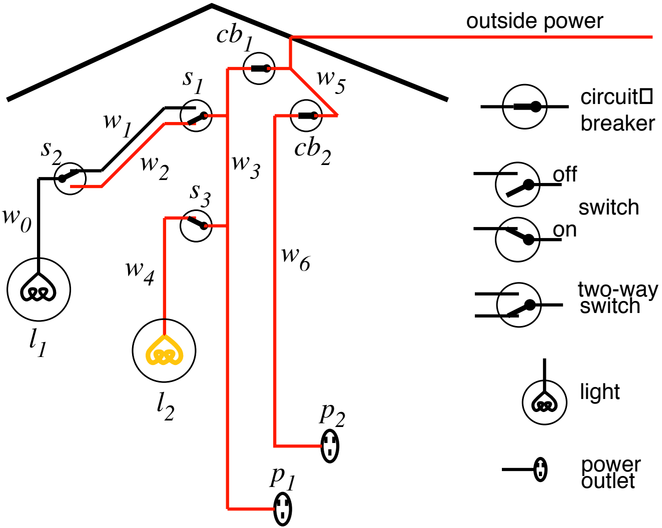
Conditional independence

Random variable X is independent of random variable Y given random variable Z if, for all $x_i \in dom(X)$, $y_j \in dom(Y)$, $y_k \in dom(Y)$ and $z_m \in dom(Z)$,

$$\begin{aligned} & P(X = x_i | Y = y_j \wedge Z = z_m) \\ & \quad = \quad P(X = x_i | Y = y_k \wedge Z = z_m) \\ & \quad = \quad P(X = x_i | Z = z_m). \end{aligned}$$

That is, knowledge of Y's value doesn't affect your belief in the value of X, given a value of Z.

Example domain (diagnostic assistant)

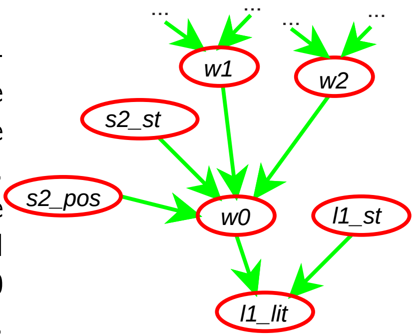


Examples of conditional independence

- The identity of the queen of Canada is independent of whether light $l1$ is lit given whether there is outside power.
- Whether there is someone in a room is independent of whether a light $l2$ is lit given the position of switch $s3$.
- Whether light $l1$ is lit is independent of the position of light switch $s2$ given whether there is power in wire w_0.
- Every other variable may be independent of whether light $l1$ is lit given whether there is power in wire w_0 and the status of light $l1$ (if it's *ok*, or if not, how it's broken).

Idea of belief networks

Whether $l1$ is lit ($L1\_lit$) depends only on the status of the light ($L1\_st$) and whether there is power in wire $w0$. Thus, $L1\_lit$ is independent of the other variables given $L1\_st$ and $W0$. In a belief network, $W0$ and $L1\_st$ are parents of $L1\_lit$.



Similarly, $W0$ depends only on whether there is power in $w1$, whether there is power in $w2$, the position of switch $s2$ ($S2\_pos$), and the status of switch $s2$ ($S2\_st$).

Belief networks

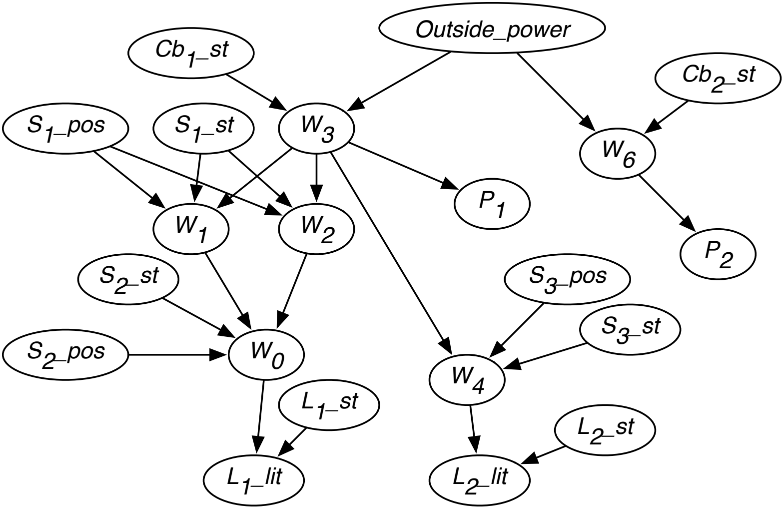
- Totally order the variables of interest: $X_1, \ldots, X_n$
- Theorem of probability theory (chain rule): $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | X_1, \ldots, X_{i-1})$
- The parents $parents(X_i)$ of X_i are those predecessors of X_i that render X_i independent of the other predecessors. That is, $parents(X_i) \subseteq X_1, \ldots, X_{i-1}$ and $P(X_i | parents(X_i)) = P(X_i | X_1, \ldots, X_{i-1})$
- So $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | parents(X_i))$
- A belief network is a graph: the nodes are random variables; there is an arc from the parents of each node into that node.

Components of a belief network

A belief network consists of:

- a directed acyclic graph with nodes labeled with random variables
- a domain for each random variable
- a set of conditional probability tables for each variable given its parents (including prior probabilities for nodes with no parents).

Example belief network



Example belief network (continued)

The belief network also specifies:

- The domain of the variables:
  $W_0, \ldots, W_6$ have domain $\{live, dead\}$
  S_1_pos, S_2_pos, and S_3_pos have domain $\{up, down\}$
  S_1_st has $\{ok, upside\_down, short, intermittent, broken\}$.
- Conditional probabilities, including:
  $P(W_1 = live | s_1\_pos = up \wedge S_1\_st = ok \wedge W_3 = live)$
  $P(W_1 = live | s_1\_pos = up \wedge S_1\_st = ok \wedge W_3 = dead)$
  $P(S_1\_pos = up)$
  $P(S_1\_st = upside\_down)$

Belief network summary

- A belief network is automatically acyclic by construction.
- A belief network is a directed acyclic graph (DAG) where nodes are random variables.
- The parents of a node n are those variables on which n directly depends.
- A belief network is a graphical representation of dependence and independence:
  - A variable is independent of its non-descendants given its parents.

Constructing belief networks

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

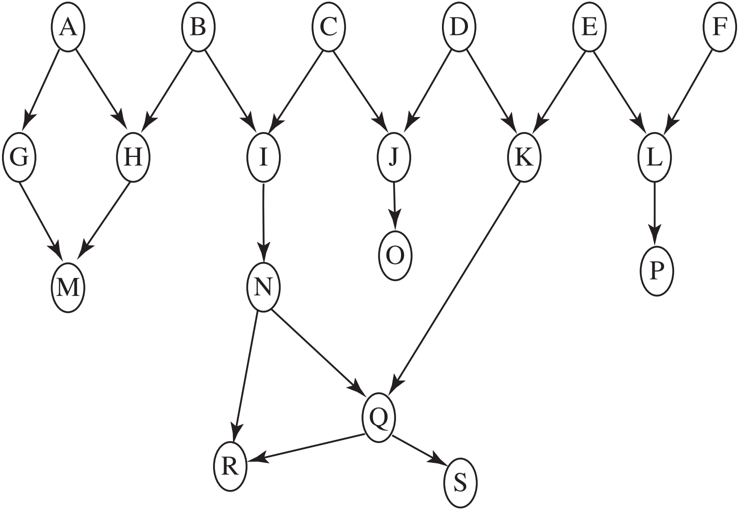
- What are the relevant variables?
  - What will you observe?
  - What would you like to find out (query)?
  - What other features make the model simpler?
- What values should these variables take?
- What is the relationship between them? This should be expressed in terms of local influence.
- How does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

Using belief networks

The power network can be used in a number of ways:

- Conditioning on the status of the switches and circuit breakers, whether there is outside power and the position of the switches, you can simulate the lighting.
- Given values for the switches, the outside power, and whether the lights are lit, you can determine the posterior probability that each switch or circuit breaker is *ok* or not.
- Given some switch positions and some outputs and some intermediate values, you can determine the probability of any other variable in the network.

Understanding independence: example



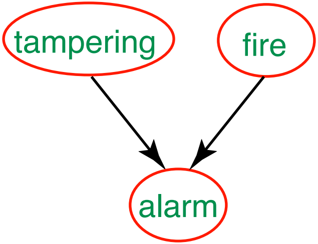
Understanding independence: questions

- On which given probabilities does $P(N)$ depend?
- If you were to observe a value for B, which variables' probabilities will change?
- If you were to observe a value for N, which variables' probabilities will change?
- Suppose you had observed a value for M; if you were to then observe a value for N, which variables' probabilities will change?
- Suppose you had observed B and Q; which variables' probabilities will change when you observe N?

What variables are affected by observing?

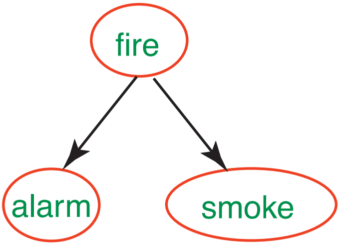
- If you observe variable $\overline{Y}$, the variables whose posterior probability is different from their prior are:
  - The ancestors of $\overline{Y}$ and
  - their descendants.
- Intuitively (if you have a causal belief network):
  - You do abduction to possible causes and
  - prediction from the causes.

Common descendants



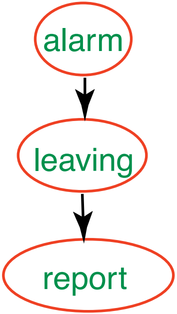
- *tampering* and *fire* are independent
- *tampering* and *fire* are dependent given *alarm*
- Intuitively, *tampering* can explain away *fire*

Common ancestors



- *alarm* and *smoke* are dependent
- *alarm* and *smoke* are independent given *fire*
- Intuitively, *fire* can explain *alarm* and *smoke*; learning one can affect the other by changing your belief in *fire*.

Chain



- *alarm* and *report* are dependent
- *alarm* and *report* are independent given *leaving*
- Intuitively, the only way that the *alarm* affects *report* is by affecting *leaving*.

Pruning Irrelevant Variables

Suppose you want to compute $P(X|e_1 \ldots e_k)$:

- Prune any variables that have no observed or queried descendents.
- Connect the parents of any observed variable.
- Remove arc directions.
- Remove observed variables.
- Remove any variables not connected to X in the resulting (undirected) graph.

Belief network inference

Four main approaches to determine posterior distributions in belief networks:

- Variable Elimination: exploit the structure of the network to eliminate (sum out) the non-observed, non-query variables one at a time.
- Search-based approaches: enumerate some of the possible worlds, and estimate posterior probabilities from the worlds generated.
- Stochastic simulation: random cases are generated according to the probability distributions.
- Variational methods: find the closest tractable distribution to the (posterior) distribution we are interested in.

Factors

A factor is a representation of a function from a tuple of random variables into a number.

We will write factor f on variables $X_1, \ldots, X_j$ as $f(X_1, \ldots, X_j)$.

We can assign some or all of the variables of a factor:

- $f(X_1 = v_1, X_2, \ldots, X_j)$, where $v_1 \in dom(X_1)$, is a factor on $X_2, \ldots, X_j$.
- $f(X_1 = v_1, X_2 = v_2, \ldots, X_j = v_j)$ is a number that is the value of f when each X_i has value v_i.

The former is also written as $f(X_1, X_2, \ldots, X_j)_{X_1 = v_1}$, etc.

Example factors

$r(X, Y, Z)$:

X	Y	Z	val
t	t	t	0.1
t	t	f	0.9
t	f	t	0.2
t	f	f	0.8
f	t	t	0.4
f	t	f	0.6
f	f	t	0.3
f	f	f	0.7

$r(X{=}t, Y, Z)$:

Y	Z	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

$r(X{=}t, Y, Z{=}f)$:

Y	val
t	0.9
f	0.8

$$r(X{=}t, Y{=}f, Z{=}f) = 0.8$$

Multiplying factors

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

$$(f_1 \times f_2)(\overline{X}, \overline{Y}, \overline{Z}) \quad = \quad f_1(\overline{X}, \overline{Y}) f_2(\overline{Y}, \overline{Z}).$$

Multiplying factors example

f_1:

A	B	val
t	t	0.1
t	f	0.9
f	t	0.2
f	f	0.8

f_2:

B	C	val
t	t	0.3
t	f	0.7
f	t	0.6
f	f	0.4

$f_1 \times f_2$:

A	B	C	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

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:

$$
\begin{aligned}
& (\sum_{X_1} f)(X_2, \ldots, X_j) \\
& = \quad f(X_1 = v_1, \ldots, X_j) + \cdots + f(X_1 = v_k, \ldots, X_j)
\end{aligned}
$$

Summing out a variable example

f_3:

A	B	C	val
t	t	t	0.03
t	t	f	0.07
t	f	t	0.54
t	f	f	0.36
f	t	t	0.06
f	t	f	0.14
f	f	t	0.48
f	f	f	0.32

$\sum_B f_3$:

A	C	val
t	t	0.57
t	f	0.43
f	t	0.54
f	f	0.46

Evidence

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

$$P(Z \mid Y_1 = v_1, \ldots, Y_j = v_j)$$

$$=$$

Evidence

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

$$
\begin{aligned}
& P(Z|Y_1 = v_1, \ldots, Y_j = v_j) \\
= \quad & \frac{P(Z, Y_1 = v_1, \ldots, Y_j = v_j)}{P(Y_1 = v_1, \ldots, Y_j = v_j)} \\
= \quad &
\end{aligned}
$$

Evidence

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

$$
\begin{aligned}
& P(Z \mid Y_1 = v_1, \ldots, Y_j = v_j) \\
= \quad & \frac{P(Z, Y_1 = v_1, \ldots, Y_j = v_j)}{P(Y_1 = v_1, \ldots, Y_j = v_j)} \\
= \quad & \frac{P(Z, Y_1 = v_1, \ldots, Y_j = v_j)}{\sum_Z P(Z, Y_1 = v_1, \ldots, Y_j = v_j)}.
\end{aligned}
$$

So the computation reduces to the probability of $P(Z, Y_1 = v_1, \ldots, Y_j = v_j)$.
We normalize at the end.

Probability of a conjunction

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_i into an elimination ordering.

$$P(Z, Y_1 = v_1, \ldots, Y_j = v_j)$$

$$=$$

Probability of a conjunction

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_i into an elimination ordering.

$$P(Z, Y_1 = v_1, \ldots, Y_j = v_j)$$
$$= \sum_{Z_k} \cdots \sum_{Z_1} P(X_1, \ldots, X_n)_{Y_1 = v_1, \ldots, Y_j = v_j}.$$

$$=$$

Probability of a conjunction

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_i into an elimination ordering.

$$
\begin{aligned}
& P(Z, Y_1 = v_1, \ldots, Y_j = v_j) \\
& \quad = \sum_{Z_k} \cdots \sum_{Z_1} P(X_1, \ldots, X_n)_{Y_1 = v_1, \ldots, Y_j = v_j}. \\
& \quad = \sum_{Z_k} \cdots \sum_{Z_1} \prod_{i=1}^{n} P(X_i | parents(X_i))_{Y_1 = v_1, \ldots, Y_j = v_j}.
\end{aligned}
$$

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 $a(b + c)$

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 $a(b + c)$
- How can we compute $\sum_{Z_1} \prod_{i=1}^{n} P(X_i|parents(X_i))$ 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 $a(b + c)$
- How can we compute $\sum_{Z_1} \prod_{i=1}^{n} P(X_i | parents(X_i))$ efficiently?
- Distribute out those factors that don't involve Z_1.

Variable elimination algorithm

To compute $P(Z|Y_1 = v_1 \wedge \ldots \wedge Y_j = v_j)$:

- Construct a factor for each conditional probability.
- Set the observed variables to their observed values.
- Sum out each of the other variables (the $\{Z_1, \ldots, Z_k\}$) according to some elimination ordering.
- Multiply the remaining factors. Normalize by dividing the resulting factor $f(Z)$ by $\sum_Z f(Z)$.

Summing out a variable

To sum out a variable Z_j from a product $f_1, \dots, f_k$ of factors:

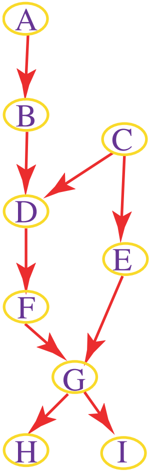
- Partition the factors into
  - those that don't contain Z_j, say $f_1, \dots, f_i$,
  - those that contain Z_j, say $f_{i+1}, \dots, f_k$

We know:

$$\sum_{Z_j} f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \times \left(\sum_{Z_j} f_{i+1} \times \cdots \times f_k \right) .$$

- Explicitly construct a representation of the rightmost factor. Replace the factors $f_{i+1}, \dots, f_k$ by the new factor.

Variable elimination example



$$\left.\begin{array}{l} P(A) \\ P(B|A) \end{array}\right\} \xrightarrow{\text{elim } A} f_1(B)$$

$$\left.\begin{array}{l} P(C) \\ P(D|BC) \\ P(E|C) \end{array}\right\} \xrightarrow{\text{elim } C} f_2(BDE)$$

$$P(F|D)$$

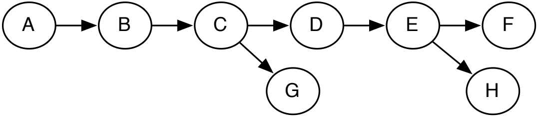
$$P(G|FE)$$

$$P(H|G)\ \} \xrightarrow{\text{obs } H} f_3(G)$$

$$P(I|G)\ \} \xrightarrow{\text{elim } I} f_4(G)$$

$$P(D,h) = \ldots(\textstyle\sum_A P(A)P(B|A))(\sum_I P(I|G))$$

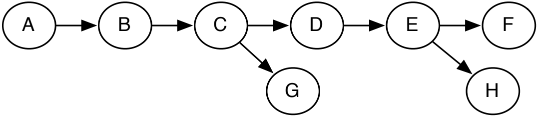
Variable Elimination example



Query: $P(G|f)$; elimination ordering: A, H, E, D, B, C

$$P(G|f) \quad \propto$$

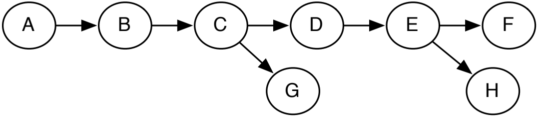
Variable Elimination example



Query: $P(G|f)$; elimination ordering: A, H, E, D, B, C

$$P(G|f) \propto \sum_C \sum_B \sum_D \sum_E \sum_H \sum_A P(A)P(B|A)P(C|B)$$
$$P(D|C)P(E|D)P(f|E)P(G|C)P(H|E)$$

Variable Elimination example



Query: $P(G|f)$; elimination ordering: A, H, E, D, B, C

$$P(G|f) \propto \sum_C \sum_B \sum_D \sum_E \sum_H \sum_A P(A)P(B|A)P(C|B)$$
$$P(D|C)P(E|D)P(f|E)P(G|C)P(H|E)$$

$$= \sum_C \left(\sum_B \left(\sum_A P(A)P(B|A) \right) P(C|B) \right) P(G|C)$$
$$\left(\sum_D P(D|C) \left(\sum_E P(E|D)P(f|E) \sum_H P(H|E) \right) \right)$$

Stochastic Simulation

- Idea: probabilities $\leftrightarrow$ samples
- Get probabilities from samples:

X	$count$
x_1	n_1
$\vdots$	$\vdots$
x_k	n_k
$total$	m

$\leftrightarrow$

X	$probability$
x_1	n_1/m
$\vdots$	$\vdots$
x_k	n_k/m

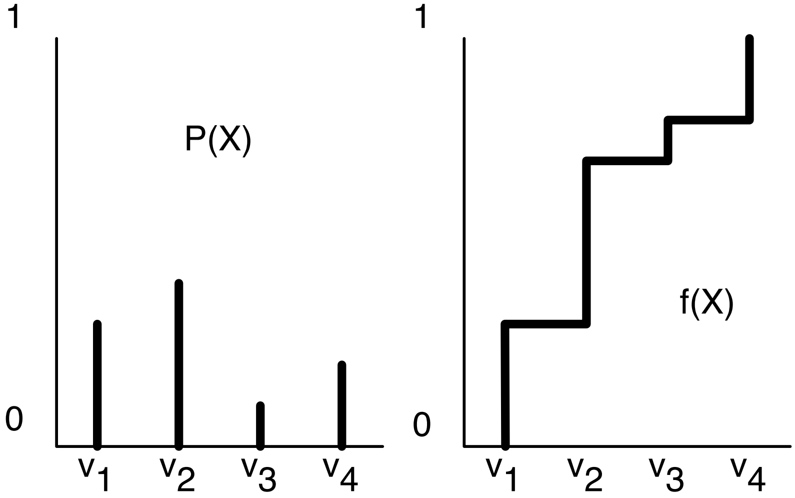
- If we could sample from a variable's (posterior) probability, we could estimate its (posterior) probability.

Generating samples from a distribution

For a variable X with a discrete domain or a (one-dimensional) real domain:

- Totally order the values of the domain of X.
- Generate the cumulative probability distribution: $f(x) = P(X \leq x)$.
- Select a value y uniformly in the range $[0, 1]$.
- Select the x such that $f(x) = y$.

Cumulative Distribution



Forward sampling in a belief network

- Sample the variables one at a time; sample parents of X before sampling X.
- Given values for the parents of X, sample from the probability of X given its parents.

Rejection Sampling

- To estimate a posterior probability given evidence $Y_1 = v_1 \wedge \ldots \wedge Y_j = v_j$:
- Reject any sample that assigns Y_i to a value other than v_i.
- The non-rejected samples are distributed according to the posterior probability:

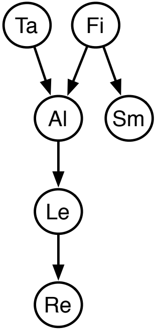
$$P(\alpha|evidence) \approx \frac{\sum_{sample \models \alpha} 1}{\sum_{sample} 1}$$

where we consider only samples consistent with evidence.

Rejection Sampling Example: $P(ta|sm, re)$

Observe $Sm = true, Re = true$

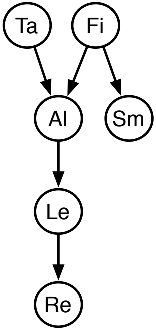
	Ta	Fi	Al	Sm	Le	Re
s_1	false	true	false	true	false	false



Rejection Sampling Example: $P(ta|sm, re)$

Observe $Sm = true, Re = true$

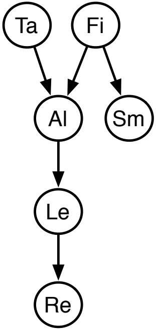
	Ta	Fi	Al	Sm	Le	Re	
s_1	false	true	false	true	false	false	✘
s_2	false	true	true	true	true	true	



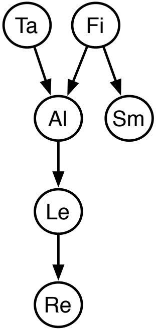
Rejection Sampling Example: $P(ta|sm, re)$

Observe $Sm = true, Re = true$

	Ta	Fi	Al	Sm	Le	Re	
s_1	false	true	false	true	false	false	✘
s_2	false	true	true	true	true	true	✔
s_3	true	false	true	false			



Rejection Sampling Example: $P(ta|sm, re)$

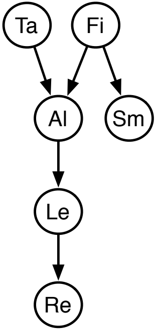


Observe $Sm = true, Re = true$

	Ta	Fi	Al	Sm	Le	Re	
s_1	false	true	false	true	false	false	✘
s_2	false	true	true	true	true	true	✔
s_3	true	false	true	false	—	—	✘
s_4	true	true	true	true	true	true	

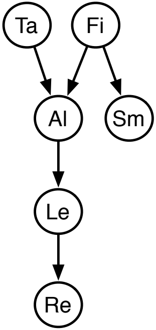
Rejection Sampling Example: $P(ta|sm, re)$

Observe $Sm = true, Re = true$



	Ta	Fi	Al	Sm	Le	Re	
s_1	false	true	false	true	false	false	✘
s_2	false	true	true	true	true	true	✔
s_3	true	false	true	false	—	—	✘
s_4	true	true	true	true	true	true	✔
...							
s_{1000}	false	false	false	false			

Rejection Sampling Example: $P(ta|sm, re)$



Observe $Sm = true, Re = true$

	Ta	Fi	Al	Sm	Le	Re	
s_1	false	true	false	true	false	false	✘
s_2	false	true	true	true	true	true	✔
s_3	true	false	true	false	—	—	✘
s_4	true	true	true	true	true	true	✔
...							
s_{1000}	false	false	false	false	—	—	✘

$P(sm) = 0.02$
$P(re|sm) = 0.32$
How many samples are rejected?
How many samples are used?

Importance Sampling

- Samples have weights: a real number associated with each sample that takes the evidence into account.
- Probability of a proposition is weighted average of samples:

$$P(\alpha|evidence) \approx \frac{\sum_{sample \models \alpha} weight(sample)}{\sum_{sample} weight(sample)}$$

- Mix exact inference with sampling: don't sample all of the variables, but weight each sample according to $P(evidence|sample)$.

Markov chain

- A Markov chain is a special sort of belief network for sequential observations:

- Thus, $P(S_{t+1}|S_0, \ldots, S_t) = P(S_{t+1}|S_t)$.
- Often S_t represents the state at time t. Intuitively S_t conveys all of the information about the history that can affect the future states.
- "The past is independent of the future given the present."

Stationary Markov chain

- A stationary Markov chain is when for all $t > 0$, $t' > 0$, $P(S_{t+1}|S_t) = P(S_{t'+1}|S_{t'})$.
- We specify $P(S_0)$ and $P(S_{t+1}|S_t)$.
  - Simple model, easy to specify
  - Often the natural model
  - The network can extend indefinitely

Markov Models

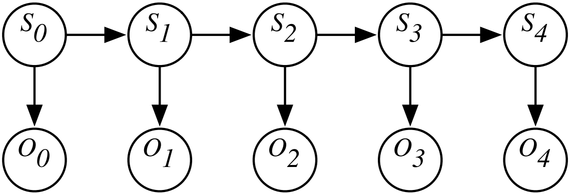
- modelling dependencies of various lengths
  - bigrams: $P(S_i|S_{i-1})$
  - trigrams: $P(S_i|S_{i-2}S_{i-1})$
  - quadrograms: $P(S_i|S_{i-3}S_{i-2}S_{i-1})$
  - ...
- e.g. to predict the probability of the next event
- speech and language processing, genome analysis, time series predictions (stock market, natural desasters, ...)

Markov Models

- examples of Markov chains for German letter sequences
- unigrams:
  aiobnin*tarsfneonlpiitdregedcoa*ds*e*dbieastnreleeucdkeaitb*
  dnurlarsls*omn*keu**svdleeoieei* ...
- bigrams:
  er*agepteprteiningeit*gerelen*re*unk*ves*mterone*hin*d*an*
  nzerurbom* ...
- trigrams:
  billunten*zugen*die*hin*se*sch*wel*war*gen*man*
  nicheleblant*diertunderstim* ...
- quadrograms:
  eist*des*nich*in*den*plassen*kann*tragen*was*wiese*
  zufahr* ...

Hidden Markov Model

- A Hidden Markov Model (HMM) is a belief network:



- $P(S_0)$ specifies initial conditions
- $P(S_{t+1}|S_t)$ specifies the dynamics
- $P(O_t|S_t)$ specifies the sensor model

Filtering

Filtering:

$$P(S_i | o_1, \ldots, o_i)$$

What is the current belief state based on the observation history?

Filtering

Filtering:

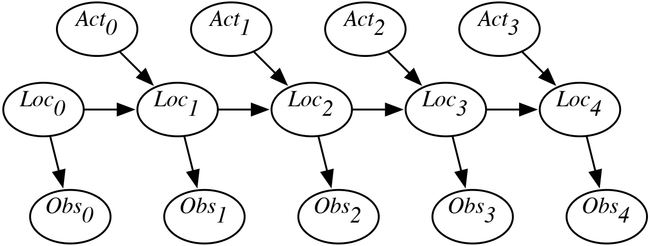
$$P(S_i | o_1, \ldots, o_i)$$

What is the current belief state based on the observation history?

$$\begin{aligned} P(S_i | o_1, \ldots, o_i) &\propto P(o_i | S_i o_1, \ldots, o_{i-1}) P(S_i | o_1, \ldots, o_{i-1}) \\ &= ??? \sum_{S_{i-1}} P(S_i S_{i-1} | o_1, \ldots, o_{i-1}) \\ &= ??? \end{aligned}$$

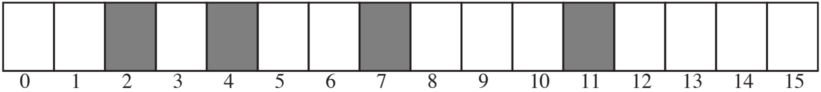
Example: localization

- Suppose a robot wants to determine its location based on its actions and its sensor readings: Localization
- This can be represented by the augmented HMM:



Example localization domain

- Circular corridor, with 16 locations:



- Doors at positions: 2, 4, 7, 11.
- Noisy Sensors
- Stochastic Dynamics
- Robot starts at an unknown location and must determine where it is.

Example Sensor Model

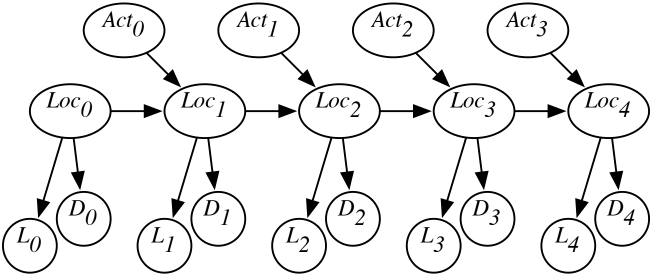
- $P(Observe\ Door \mid At\ Door) = 0.8$
- $P(Observe\ Door \mid Not\ At\ Door) = 0.1$

Example Dynamics Model

- $P(loc_{t+1} = L | action_t = goRight \wedge loc_t = L) = 0.1$
- $P(loc_{t+1} = L+1 | action_t = goRight \wedge loc_t = L) = 0.8$
- $P(loc_{t+1} = L+2 | action_t = goRight \wedge loc_t = L) = 0.074$
- $P(loc_{t+1} = L' | action_t = goRight \wedge loc_t = L) = 0.002$ for any other location L'.
  - All location arithmetic is modulo 16.
  - The action *goLeft* works the same but to the left.

Combining sensor information

- Example: we can combine information from a light sensor and the door sensor Sensor Fusion



S_t robot location at time t
D_t door sensor value at time t
L_t light sensor value at time t