Artificial Intelligence 2
Summer Semester 2025

– Lecture Notes –
Part V: Reasoning with Uncertain Knowledge

Prof. Dr. Michael Kohlhase

Professur für Wissensrepräsentation und -verarbeitung
Informatik, FAU Erlangen-Nürnberg
`Michael.Kohlhase@FAU.de`

2025-05-01

This document contains Part V of the course notes for the course "Artificial Intelligence 2" held at FAU Erlangen-Nürnberg in the Summer Semesters 2017 ff. This part of the lecture notes addresses inference and agent decision making in partially observable environments, i.e. where we only know probabilities instead of certainties whether propositions are true/false. We cover basic probability theory and – based on that – Bayesian Networks and simple decision making in such environments. Finally we extend this to probabilistic temporal models and their decision theory. Other parts of the lecture notes can be found at `http://kwarc.info/teaching/AI/notes-*.pdf`.

Contents

Chapter 22

Quantifying Uncertainty

In this chapter we develop a machinery for dealing with uncertainty: Instead of thinking about what we know to be true, we must think about what is likely to be true.

22.1 Probability Theory

22.1.1 Prior and Posterior Probabilities

Probabilistic Models

- **Definition 22.1.1 (Mathematically (slightly simplified)).** A probability space or (probability model) is a pair $\langle\Omega, P\rangle$ such that:
  - Ω is a set of outcomes (called the sample space),
  - P is a function $\mathcal{P}(\Omega) \to [0,1]$, such that:
    - $P(\Omega) = 1$ and
    - $P(\bigcup_i A_i) = \sum_i P(A_i)$ for all pairwise disjoint $A_i \in \mathcal{P}(\Omega)$.

    P is called a probability measure.

  These properties are called the Kolmogorov axioms.
- **Intuition:** We run some experiment, the outcome of which is any $\omega \in \Omega$.
  - For $X \subseteq \Omega$, $P(X)$ is the probability that the result of the experiment is *any one* of the outcomes in X.
  - Naturally, the probability that *any* outcome occurs is 1 (hence $P(\Omega) = 1$).
  - The probability of pairwise disjoint sets of outcomes should just be the sum of their probabilities.
- **Example 22.1.2 (Dice throws).** Assume we throw a (fair) die two times. Then the sample space Ω is $\{(i,j) \mid 1 \leq i,j \leq 6\}$. We define P by letting $P(\{A\}) = \frac{1}{36}$ for every $A \in \Omega$.

  Since the probability of any outcome is the same, we say P is uniformly distributed.

FAU Michael Kohlhase: Artificial Intelligence 2 734 2025-05-01

The definition is simplified in two places: Firstly, we assume that P is defined on the full power set. This is not always possible, especially if Ω is uncountable. In that case we need an additional set of "events" instead, and lots of mathematical machinery to make sure that we can safely take unions, intersections, complements etc. of these events.

Secondly, we would technically only demand that P is additive on countably many disjoint sets.

In this course we will assume that our sample space is at most countable anyway; usually even finite.

Random Variables

- ▷ In practice, we are rarely interested in the *specific* outcome of an experiment, but rather in some *property* of the outcome. This is especially true in the very common situation where we don't even *know* the precise probabilities of the individual outcomes.
- ▷ **Example 22.1.3.** The probability that the *sum* of our two dice throws is 7 is $P(\{(i,j) \in \Omega \mid i+j=7\}) = P(\{(6,1),(1,6),(5,2),(2,5),(4,3),(3,4)\}) = \frac{6}{36} = \frac{1}{6}$.
- ▷ **Definition 22.1.4 (Again, slightly simplified).** Let D be a set. A random variable is a function $X\colon \Omega \to D$. We call D (somewhat confusingly) the domain of X, denoted $\mathrm{dom}(X)$.

  For $x \in D$, we define the probability of x as $P(X=x) := P(\{\omega \in \Omega \mid X(\omega)=x\})$.
- ▷ **Definition 22.1.5.** We say that a random variable X is finite domain, iff its domain $\mathrm{dom}(X)$ is finite and Boolean, iff $\mathrm{dom}(X) = \{\mathsf{T},\mathsf{F}\}$.

  For a Boolean random variable, we will simply write $P(X)$ for $P(X=\mathsf{T})$ and $P(\neg X)$ for $P(X=\mathsf{F})$.

Note that a random variable, according to the formal definition, is *neither* random *nor* a variable: It is a function with clearly defined domain and codomain – and what we call the domain of the "variable" is actually its codomain... are you confused yet? ☺

This confusion is a side-effect of the *mathematical* formalism. In practice, a random variable is some indeterminate value that results from some statistical experiment – i.e. it is *random*, because the result is not predetermined, and it is a variable, because it can take on different values.

It just so happens that if we want to model this scenario *mathematically*, a function is the most natural way to do so.

Some Examples

- ▷ **Example 22.1.6.** Summing up our two dice throws is a random variable $S\colon \Omega \to [2{,}12]$ with $S((i,j)) = i+j$. The probability that they sum up to 7 is written as $P(S=7) = \frac{1}{6}$.
- ▷ **Example 22.1.7.** The first and second of our two dice throws are random variables $\mathrm{First}, \mathrm{Second}\colon \Omega \to [1{,}6]$ with $\mathrm{First}((i,j)) = i$ and $\mathrm{Second}((i,j)) = j$.
- ▷ *Remark 22.1.8.* Note, that the *identity* $\Omega \to \Omega$ is a random variable as well.
- ▷ **Example 22.1.9.** We can model toothache, cavity and gingivitis as Boolean random variables, with the underlying probability space being...?? ¯_(ツ)_/¯
- ▷ **Example 22.1.10.** We can model tomorrow's weather as a random variable with domain $\{\texttt{sunny}, \texttt{rainy}, \texttt{foggy}, \texttt{warm}, \texttt{cloudy}, \texttt{humid}, \ldots\}$, with the underlying probability space being...?? ¯_(ツ)_/¯

⇒ This is why *probabilistic reasoning* is necessary: We can rarely reduce probabilistic scenarios down to clearly defined, fully known probability spaces and derive all the interesting things from there.

But: The definitions here allow us to *reason* about probabilities and random variables in a *mathematically* rigorous way, e.g. to make our intuitions and assumptions precise, and prove our methods to be *sound*.

Propositions

▷ This is nice and all, but in practice we are interested in "compound" probabilities like:

"What is the probability that the sum of our two dice throws is 7, but neither of the two dice is a 3?"

▷ **Idea:** Reuse the syntax of propositional logic and define the logical connectives for random variables!

▷ **Example 22.1.11.** We can express the above as: $P(\neg(\mathrm{First}=3)\wedge\neg(\mathrm{Second}=3)\wedge(S=7))$

▷ **Definition 22.1.12.** Let X_1, X_2 be random variables, $x_1 \in \mathrm{dom}(X_1)$ and $x_2 \in \mathrm{dom}(X_2)$. We define:

1. $P(X_1 \neq x_1){:=}P(\neg(X_1 = x_1)) := P(\{\omega \in \Omega \,|\, X_1(\omega) \neq x_1\}){=}1 - P(X_1 = x_1)$.
2. $P((X_1 = x_1) \wedge (X_2 = x_2)) := P(\{\omega \in \Omega \,|\, (X_1(\omega) = x_1) \wedge (X_2(\omega) = x_2)\}) {=}P(\{\omega \in \Omega \,|\, X_1(\omega) = x_1\} \cap \{\omega \in \Omega \,|\, X_2(\omega) = x_2\})$.
3. $P((X_1 = x_1) \vee (X_2 = x_2)) := P(\{\omega \in \Omega \,|\, (X_1(\omega) = x_1) \vee (X_2(\omega) = x_2)\}) {=}P(\{\omega \in \Omega \,|\, X_1(\omega) = x_1\} \cup \{\omega \in \Omega \,|\, X_2(\omega) = x_2\})$.

It is also common to write $P(A, B)$ for $P(A \wedge B)$

▷ **Example 22.1.13.** $P((\mathrm{First} \neq 3)\wedge(\mathrm{Second} \neq 3)\wedge(S=7)) = P(\{(1,6),(6,1),(2,5),(5,2)\}) = \frac{1}{9}$

Events

▷ **Definition 22.1.14 (Again slightly simplified).** Let $\langle\Omega, P\rangle$ be a probability space. An event is a subset of Ω.

▷ **Definition 22.1.15 (Convention).** We call an event (by extension) anything that *represents* a subset of Ω: any statement formed from the logical connectives and values of random variables, on which $P(\cdot)$ is defined.

▷ **Problem 1.1**

Remember: We can define $A \vee B := \neg(\neg A \wedge \neg B)$, $\mathsf{T} := A \vee \neg A$ and $\mathsf{F} := \neg\mathsf{T}$ – is this compatible with the definition of probabilities on propositional formulae? And why is $P(X_1 \neq x_1) = 1 - P(X_1 = x_1)$?

▷ **Problem 1.2 (Inclusion-Exclusion-Principle)**
Show that $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$.

▷ **Problem 1.3**
Show that $P(A) = P(A \wedge B) + P(A \wedge \neg B)$

Conditional Probabilities

▷ **Observation:** As we gather new information, our beliefs (*should*) change, and thus our probabilities!

▷ **Example 22.1.16.** Your "probability of missing the connection train" increases when you are informed that your current train has 30 minutes delay.

▷ **Example 22.1.17.** The "probability of cavity" increases when the doctor is informed that the patient has a toothache.

▷ **Example 22.1.18.** The probability that $S = 3$ is clearly higher if I know that $\mathrm{First} = 1$ than otherwise – or if I know that $\mathrm{First} = 6$!

▷ **Definition 22.1.19.** Let A and B be events where $P(B) \neq 0$. The conditional probability of A given B is defined as:

$$P(A|B) := \frac{P(A \wedge B)}{P(B)}$$

We also call $P(A)$ the prior probability of A, and $P(A|B)$ the posterior probability.

▷ **Intuition:** If we *assume* B to hold, then we are only interested in the "part" of Ω where A is true *relative to* B.

▷ **Alternatively:** We restrict our sample space Ω to the subset of outcomes where B holds. We then define a new probability space on this subset by scaling the probability measure so that it sums to 1 – which we do by dividing by $P(B)$. (We "*update our beliefs* based on new evidence")

Examples

▷ **Example 22.1.20.** If we assume $\mathrm{First} = 1$, then $P(S = 3|\mathrm{First} = 1)$ should be precisely $P(\mathrm{Second} = 2) = \frac{1}{6}$. We check:

$$P(S = 3|\mathrm{First} = 1) = \frac{P((S = 3) \wedge (\mathrm{First} = 1))}{P(\mathrm{First} = 1)} = \frac{1/36}{1/6} = \frac{1}{6}$$

▷ **Example 22.1.21.** Assume the prior probability $P(\mathrm{cavity})$ is 0.122. The probability that a patient has both a cavity and a toothache is $P(\mathrm{cavity} \wedge \mathrm{toothache}) = 0.067$. The probability that a patient has a toothache is $P(\text{toothache}) = 0.15$.

If the patient complains about a toothache, we can update our estimation by computing the posterior probability:

$$P(\text{cavity}|\text{toothache}) = \frac{P(\text{cavity} \wedge \text{toothache})}{P(\text{toothache})} = \frac{0.067}{0.15} = 0.45.$$

▷ **Note:** We just computed the probability of some underlying *disease* based on the presence of a *symptom*!

▷ **More Generally:** We computed the probability of a *cause* from observing its *effect*.

Some Rules

▷ Equations on unconditional probabilities have direct analogues for conditional probabilities.

▷ **Problem 1.4**

Convince yourself of the following:

- ▷ $P(A|C) = 1 - P(\neg A|C)$.
- ▷ $P(A|C) = P(A \wedge B|C) + P(A \wedge \neg B|C)$.
- ▷ $P(A \vee B|C) = P(A|C) + P(B|C) - P(A \wedge B|C)$.

▷ But **not on the right hand side!**

▷ **Problem 1.5**

Find *counterexamples* for the following (**false**) claims:

- ▷ $P(A|C) = 1 - P(A|\neg C)$
- ▷ $P(A|C) = P(A|B \wedge C) + P(A|B \wedge \neg C)$.
- ▷ $P(A|B \vee C) = P(A|B) + P(A|C) - P(A|B \wedge C)$.

Bayes' Rule

▷ **Note:** By definition, $P(A|B) = \frac{P(A \wedge B)}{P(B)}$. In practice, we often know the conditional probability already, and use it to compute the probability of the conjunction instead: $P(A \wedge B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A)$.

▷ **Theorem 22.1.22 (Bayes' Theorem).** *Given propositions A and B where $P(A) \neq 0$ and*

$P(B) \neq 0$, *we have:*

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

▷ *Proof:*

1. $P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{P(B|A) \cdot P(A)}{P(B)}$

□

...okay, that was straightforward... what's the big deal?

▷ **(Somewhat Dubious) Claim:** Bayes' Rule is the entire scientific method condensed into a single equation!

▷ This is an extreme overstatement, but there is a grain of truth in it.

Bayes' Theorem - Why the Hype?

▷ Say we have a *hypothesis* H about the world. (e.g. "The universe had a beginning")

▷ We have *some prior belief* $P(H)$.

▷ We gather *evidence* E. (e.g. "We observe a cosmic microwave background at 2.7K everywhere")

▷ Bayes' Rule tells us how to *update our belief* in H based on H's ability to *predict* E (the *likelihood* $P(E|H)$) – *and*, importantly, the ability of *competing hypotheses* to predict the *same* evidence. (This is actually how scientific hypotheses should be evaluated)

$$\underbrace{P(H|E)}_{\text{posterior}} = \frac{P(E|H) \cdot P(H)}{P(E)} = \frac{\overbrace{P(E|H)}^{\text{likelihood}} \cdot \overbrace{P(H)}^{\text{prior}}}{\underbrace{P(E|H)}_{\text{likelihood}} \underbrace{P(H)}_{\text{prior}} + \underbrace{P(E|\neg H) P(\neg H)}_{\text{competition}}}$$

...if I keep gathering evidence and update, ultimately the impact of the prior belief will diminish.

"You're entitled to your own priors, but not your own likelihoods"

22.1.2 Independence

Independence

▷ **Question:** What is the probability that $S = 7$ *and* the patient has a toothache?

Or less contrived: What is the probability that the patient has a gingivitis *and* a cavity?

▷ **Definition 22.1.23.** Two events A and B are called independent, iff $P(A \wedge B) = P(A) \cdot P(B)$.

Two random variables X_1, X_2 are called independent, iff for all $x_1 \in \mathrm{dom}(X_1)$ and $x_2 \in \mathrm{dom}(X_2)$, the events $X_1 = x_1$ and $X_2 = x_2$ are independent. We write $A \perp B$ or $X_1 \perp X_2$, respectively.

▷ **Theorem 22.1.24.** *Equivalently: Given events A and B with $P(B) \neq 0$, then A and B are independent iff $P(A|B) = P(A)$ (equivalently: $P(B|A) = P(B)$).*

▷ *Proof:*

1. $\Rightarrow$
   By definition, $P(A|B) = \frac{P(A \wedge B)}{P(B)} = \frac{P(A) \cdot P(B)}{P(B)} = P(A)$,

3. $\Leftarrow$
   Assume $P(A|B) = P(A)$.
   Then $P(A \wedge B) = P(A|B) \cdot P(B) = P(A) \cdot P(B)$.

□

▷ **Note:** Independence asserts that two events are *"not related"* – the probability of one does not depend on the other.

Mathematically, we can *determine* independence by checking whether $P(A \wedge B) = P(A) \cdot P(B)$.

In practice, this is impossible to check. Instead, we *assume* independence based on *domain knowledge*, and then *exploit* this to compute $P(A \wedge B)$.

Independence (Examples)

▷ **Example 22.1.25.**

- $\mathrm{First} = 2$ and $\mathrm{Second} = 3$ are independent – more generally, First and Second are independent (The outcome of the first die does not affect the outcome of the second die)
  Quick check: $P((\mathrm{First} = a) \wedge (\mathrm{Second} = b)) = \frac{1}{36} = P(\mathrm{First} = a) \cdot P(\mathrm{Second} = b)$ ✓
- First and S are **not** independent. (The outcome of the first die affects the sum of the two dice.) Counterexample: $P((\mathrm{First} = 1) \wedge (S = 4)) = \frac{1}{36} \neq P(\mathrm{First} = 1) \cdot P(S = 4) = \frac{1}{6} \cdot \frac{1}{2} = \frac{1}{72}$
- **But:** $P((\mathrm{First} = a) \wedge (S = 7)) = \frac{1}{36} = \frac{1}{6} \cdot \frac{1}{6} = P(\mathrm{First} = a) \cdot P(S = 7)$ – so the events $\mathrm{First} = a$ and $S = 7$ *are* independent. (Why?)

▷ **Example 22.1.26.**

- Are cavity and toothache independent?
  ...since cavities can cause a toothache, that would probably be a bad design decision ...
- Are cavity and gingivitis independent? Cavities do not cause gingivitis, and gingivitis does not cause cavities, so... yes... right? (...as far as I know. I'm not a dentist.)
- **Probably not!** A patient who has cavities has probably worse dental hygiene than those who don't, and is thus more likely to have gingivitis as well.

▷ $\rightsquigarrow$ cavity may be *evidence* that raises the probabilty of gingivitis, even if they are not directly causally related.

Conditional Independence – Motivation

▷ A dentist can diagnose a cavity by using a *probe*, which may (or may not) *catch* in a cavity.

▷ Say we know from clinical studies that $P(\text{cavity}) = 0.2$, $P(\text{toothache}|\text{cavity}) = 0.6$, $P(\text{toothache}|\neg\text{cavity}) = 0.1$, $P(\text{catch}|\text{cavity}) = 0.9$, and $P(\text{catch}|\neg\text{cavity}) = 0.2$.

▷ Assume the patient complains about a toothache, and our probe indeed catches in the aching tooth. What is the likelihood of having a cavity $P(\text{cavity}|\text{toothache} \wedge \text{catch})$?

▷ **Idea:** Use Bayes' rule:

$$P(\text{cavity}|\text{toothache} \wedge \text{catch}) = \frac{P(\text{toothache} \wedge \text{catch}|\text{cavity}) \cdot P(\text{cavity})}{P(\text{toothache} \wedge \text{catch})}$$

▷ **Note:** $P(\text{toothache} \wedge \text{catch}) = P(\text{toothache} \wedge \text{catch}|\text{cavity}) \cdot P(\text{cavity}) + P(\text{toothache} \wedge \text{catch}|\neg\text{cavity}) \cdot P(\neg\text{cavity})$

▷ **Problem:** Now we're only missing $P(\text{toothache} \wedge \text{catch}|\text{cavity} = b)$ for $b \in \{\mathsf{T}, \mathsf{F}\}$. ... Now what?

▷ Are toothache and catch independent, maybe? **No**: Both have a common (possible) cause, cavity.

Also, there's this pesky $P(\cdot|\text{cavity})$ in the way. . . ...wait a minute...

Conditional Independence – Definition

▷ *Assuming* the patient has (or does not have) a cavity, the events toothache and catch are independent: Both are caused by a cavity, but they don't influence each other otherwise.

i.e. cavity "contains all the information" that links toothache and catch in the first place.

▷ **Definition 22.1.27.** Given events A, B, C with $P(C) \neq 0$, then A and B are called conditionally independent given C, iff $P(A \wedge B|C) = P(A|C) \cdot P(B|C)$.

Equivalently: iff $P(A|B \wedge C) = P(A|C)$, or $P(B|A \wedge C) = P(B|C)$.

Let Y be a random variable. We call two random variables X_1, X_2 conditionally independent given Y, iff for all $x_1 \in \mathbf{dom}(X_1)$, $x_2 \in \mathbf{dom}(X_2)$ and $y \in \mathbf{dom}(Y)$, the events $X_1 = x_1$ and $X_2 = x_2$ are conditionally independent given $Y = y$.

▷ **Example 22.1.28.** Let's assume toothache and catch are conditionally independent given cavity/$\neg$cavity. Then we can finally compute:

$P(\text{cavity}|\text{toothache} \wedge \text{catch}) = \frac{P(\text{toothache}\wedge\text{catch}|\text{cavity})\cdot P(\text{cavity})}{P(\text{toothache}\wedge\text{catch})}$

$= \frac{P(\text{toothache}|\text{cavity})\cdot P(\text{catch}|\text{cavity})\cdot P(\text{cavity})}{P(\text{toothache}|\text{cavity})\cdot P(\text{catch}|\text{cavity})\cdot P(\text{cavity})+P(\text{toothache}|\neg\text{cavity})\cdot P(\text{catch}|\neg\text{cavity})\cdot P(\neg\text{cavity})} = \frac{0.6\cdot 0.9\cdot 0.2}{0.6\cdot 0.9\cdot 0.2+0.1\cdot 0.2\cdot 0.8} = 0.87$

Conditional Independence

▷ **Lemma 22.1.29.** *If A and B are conditionally independent given C, then $P(A|B \wedge C) = P(A|C)$*

Proof:

$P(A|B \wedge C) = \frac{P(A\wedge B\wedge C)}{P(B\wedge C)} = \frac{P(A\wedge B|C)\cdot P(C)}{P(B\wedge C)} = \frac{P(A|C)\cdot P(B|C)\cdot P(C)}{P(B\wedge C)} = \frac{P(A|C)\cdot P(B\wedge C)}{P(B\wedge C)} = P(A|C)$

□

▷ **Question:** If A and B are conditionally independent given C, does this imply that A and B are independent? **No.** See previous slides for a counterexample.

▷ **Question:** If A and B are independent, does this imply that A and B are also conditionally independent given C? **No.** For example: First and Second are independent, but not conditionally independent given $S = 4$.

▷ **Question:** Okay, so what if A, B and C are *all* pairwise independent? Are A and B conditionally independent given C *now*? **Still no.** Remember: $\text{First} = a$, $\text{Second} = b$ and $S = 7$ are all independent, but First and Second are not conditionally independent given $S = 7$.

▷ **Question:** When can we infer conditional independence from a "more general" notion of independence?

We need *mutual independence*. Roughly: A set of events is called *mutually* independent, if every event is independent from *any conjunction of the others*. (Not really relevant for this course though)

22.1.3 Conclusion

Summary

▷ Probability spaces serve as a mathematical model (and hence justification) for everything related to probabilities.

▷ The "atoms" of any statement of probability are the random variables. (Important special cases: Boolean and finite domain)

▷ We can define probabilities on compund (propositional logical) statements, with (outcomes of) random variables as "propositional variables".

▷ Conditional probabilities represent *posterior probabilities* given some observed outcomes.

- independence and conditional independence are strong assumptions that allow us to simplify computations of probabilities
- Bayes' Theorem

So much about the math...

- We now have a mathematical setup for probabilities.
- **But:** The math does not tell us what probabilities *are*:
- Assume we can mathematically derive this to be the case: *the probability of rain tomorrow is* 0.3. What does this even *mean*?
- **Frequentist Answer:** The probability of an event is the limit of its relative frequency in a large number of trials.

  In other words: "In 30% of the cases where we have similar weather conditions, it rained the next day."
- **Objection:** Okay, but what about *unique* events? "The probability of me passing the exam is 80%" – does this mean anything, if I only take the exam once? Am I comparable to "similar students"? What counts as sufficiently "similar"?
- **Bayesian Answer:** Probabilities are *degrees of belief*. It means you **should** be 30% confident that it will rain tomorrow.
- **Objection:** And why *should* I? Is this not purely *subjective* then?

Pragmatics

- **Pragmatically** both interpretations amount to the same thing: I should *act as if* I'm 30% confident that it will rain tomorrow. (Whether by fiat, or because in 30% of comparable cases, it rained.)
- **Objection:** Still: why should I? And why should my beliefs follow the seemingly arbitrary Kolmogorov axioms?
- [DF31]: If an agent has a belief that violates the Kolmogorov axioms, then there exists a combination of "bets" on propositions so that the agent *always* loses money.
- **In other words:** If your beliefs are not consistent with the mathematics, and you *act in accordance with your beliefs*, there is a way to exploit this inconsistency to your disadvantage.
- ...and, more importantly, the AI agents you design! ☺

22.2 Probabilistic Reasoning Techniques

Okay, now how do I implement this?

- ▷ This is a CS course. We need to implement this stuff.
- ▷ Do we... implement random variables as functions? Is a probability space a... class maybe?
- ▷ **No:** As mentioned, we rarely know the probability space entirely. Instead we will use probability distributions, which are just arrays (of arrays of...) of probabilities.
- ▷ And then we represent *those* as sparsely as possible, by exploiting independence, conditional independence, ...

22.2.1 Probability Distributions

Probability Distributions

- ▷ **Definition 22.2.1.** The probability distribution for a random variable X, written $\mathbb{P}(X)$, is the vector of probabilities for the (ordered) domain of X.
- ▷ **Note:** The values in a probability distribution are all positive and sum to 1. (Why?)
- ▷ **Example 22.2.2.** $\mathbb{P}(\text{First}) = \mathbb{P}(\text{Second}) = \langle\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\rangle$. (Both First and Second are uniformly distributed)
- ▷ **Example 22.2.3.** The probability distribution $\mathbb{P}(S)$ is $\langle\frac{1}{36}, \frac{1}{18}, \frac{1}{12}, \frac{1}{9}, \frac{5}{36}, \frac{1}{6}, \frac{5}{36}, \frac{1}{9}, \frac{1}{12}, \frac{1}{18}, \frac{1}{36}\rangle$. Note the symmetry, with a "peak" at 7 – the random variable is (*approximately*, because our domain is discrete rather than continuous) normally distributed (or gaussian distributed, or follows a bell-curve,...).
- ▷ **Example 22.2.4.** Probability distributions for Boolean random variables are naturally *pairs* (probabilities for T and F), e.g.:

$$\mathbb{P}(\text{toothache}) = \langle 0.15, 0.85\rangle$$
$$\mathbb{P}(\text{cavity}) = \langle 0.122, 0.878\rangle$$

- ▷ More generally:

  Definition 22.2.5. A probability distribution is a vector $\mathbf{v}$ of values $\mathbf{v}_i \in [0{,}1]$ such that $\sum_i \mathbf{v}_i = 1$.

The Full Joint Probability Distribution

- ▷ **Definition 22.2.6.** Given random variables $X_1, \ldots, X_n$, the full joint probability distribution, denoted $\mathbb{P}(X_1, \ldots, X_n)$, is the n-dimensional array of size $|D_1 \times \ldots \times D_n|$ that lists the probabilities of all conjunctions of values of the random variables.
- ▷ **Example 22.2.7.** $\mathbb{P}(\text{cavity}, \text{toothache}, \text{gingivitis})$ could look something like this:

	toothache		¬toothache	
	gingivitis	¬gingivitis	gingivitis	¬gingivitis
cavity	0.007	0.06	0.005	0.05
¬cavity	0.08	0.003	0.045	0.75

▷ **Example 22.2.8.** $\mathbb{P}(\mathrm{First}, S)$

First \ S	2	3	4	5	6	7	8	9	10	11	12
1	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	0	0	0	0	0
2	0	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	0	0	0	0
3	0	0	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	0	0	0
4	0	0	0	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	0	0
5	0	0	0	0	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	0
6	0	0	0	0	0	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$	$\frac{1}{36}$

Note that if we know the value of First, the value of S is completely determined by the value of Second.

Conditional Probability Distributions

▷ **Definition 22.2.9.** Given random variables X and Y, the conditional probability distribution of X given Y, written $\mathbb{P}(X|Y)$ is the table of all conditional probabilities of values of X given values of Y.

▷ For sets of variables analogously: $\mathbb{P}(X_1,\ldots,X_n|Y_1,\ldots,Y_m)$.

▷ **Example 22.2.10.** $\mathbb{P}(\mathrm{cavity}|\mathrm{toothache})$:

	toothache	¬toothache
cavity	$P(\mathrm{cavity}|\mathrm{toothache}) = 0.45$	$P(\mathrm{cavity}|\neg\mathrm{toothache}) = 0.065$
¬cavity	$P(\neg\mathrm{cavity}|\mathrm{toothache}) = 0.55$	$P(\neg\mathrm{cavity}|\neg\mathrm{toothache}) = 0.935$

▷ **Example 22.2.11.** $\mathbb{P}(\mathrm{First}|S)$

First \ S	2	3	4	5	6	7	8	9	10	11	12
1	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	0	0	0	0	0
2	0	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{5}$	0	0	0	0
3	0	0	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{5}$	$\frac{1}{4}$	0	0	0
4	0	0	0	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{3}$	0	0
5	0	0	0	0	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	0
6	0	0	0	0	0	$\frac{1}{6}$	$\frac{1}{5}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{2}$	1

▷ **Note:** Every "column" of a conditional probability distribution is itself a probability distribution.
(Why?)

Convention

▷ We now "lift" multiplication and division to the level of whole probability distributions:

▷ **Definition 22.2.12.** Whenever we use $\mathbb{P}$ in an equation, we take this to mean a *system of equations*, for each value in the domains of the random variables involved.

Example 22.2.13.

▷ $\mathbb{P}(X,Y) = \mathbb{P}(X|Y) \cdot \mathbb{P}(Y)$ represents the system of equations $P(X = x \wedge Y = y) = P(X = x|Y = y) \cdot P(Y = y)$ for all x, y in the respective domains.

▷ $\mathbb{P}(X|Y) := \frac{\mathbb{P}(X,Y)}{\mathbb{P}(Y)}$ represents the system of equations $P(X = x|Y = y) := \frac{P((X=x) \wedge (Y=y))}{P(Y=y)}$

▷ Bayes' Theorem: $\mathbb{P}(X|Y) = \frac{\mathbb{P}(Y|X) \cdot \mathbb{P}(X)}{\mathbb{P}(Y)}$ represents the system of equations $P(X = x|Y = y) = \frac{P(Y=y|X=x) \cdot P(X=x)}{P(Y=y)}$

So, what's the point?

▷ Obviously, the probability distribution contains all the information about a specific random variable we need.

▷ **Observation:** The full joint probability distribution of variables $X_1, \ldots, X_n$ contains *all* the information about the random variables *and their conjunctions* we need.

▷ **Example 22.2.14.** We can read off the probability $P(\text{toothache})$ from the full joint probability distribution as $0.007+0.06+0.08+0.003=0.15$, and the probability $P(\text{toothache} \wedge \text{cavity})$ as $0.007 + 0.06 = 0.067$

▷ We can actually implement this! (They're just (nested) arrays)

But just as we often don't have a fully specified probability space to work in, we often don't have a full joint probability distribution for our random variables either.

▷ Also: Given random variables $X_1, \ldots, X_n$, the full joint probability distribution has $\prod_{i=1}^{n} |\text{dom}(X_i)|$ entries! ($\mathbb{P}(\text{First}, S)$ already has 60 entries!)

⇒ The rest of this section deals with keeping things small, by *computing* probabilities instead of *storing* them all.

Probabilistic Reasoning

▷ **Probabilistic reasoning** refers to inferring probabilities of events from the probabilities of other events

as opposed to determining the probabilities e.g. *empirically*, by gathering (sufficient amounts of *representative*) data and counting.

▷ **Note:** In practice, we are *primarily* interested in, and have access to, conditional probabilities rather than the unconditional probabilities of conjunctions of events:

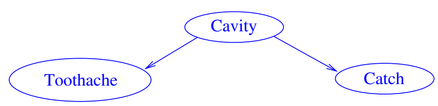
- ▷ We don't reason in a vacuum: Usually, we have some evidence and want to infer the posterior probability of some related event. (e.g. infer a plausible *cause* given some *symptom*)

  ⇒ we are interested in the conditional probability $P(\text{hypothesis}|\text{observation})$.
- ▷ *"80% of patients with a cavity complain about a toothache"* (i.e. $P(\text{toothache}|\text{cavity})$) is more the kind of data people actually collect and publish than *"1.2% of the general population have both a cavity and a toothache"* (i.e. $P(\text{cavity} \wedge \text{toothache})$).
- ▷ Consider the probe catching in a cavity. The probe is a diagnostic tool, which is usually evaluated in terms of its *sensitivity* $P(\text{catch}|\text{cavity})$ and *specificity* $P(\neg\text{catch}|\neg\text{cavity})$. (You have probably heard these words a lot since 2020...)

22.2.2 Naive Bayes

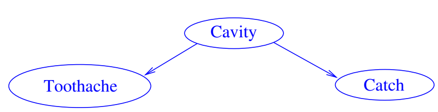
Naive Bayes Models

- ▷ Consider again the dentistry example with random variables cavity, toothache, and catch. We assume cavity **causes** both toothache and catch, and that toothache and catch are conditionally independent given cavity:



- ▷ We likely know the *sensitivity* $P(\text{catch}|\text{cavity})$ and *specificity* $P(\neg\text{catch}|\neg\text{cavity})$, which jointly give us $\mathbb{P}(\text{catch}|\text{cavity})$, and from medical studies, we should be able to determine $P(\text{cavity})$ (the *prevalence* of cavities in the population) and $\mathbb{P}(\text{toothache}|\text{cavity})$.
- ▷ This kind of situation is surprisingly common, and therefore deserves a name.

Naive Bayes Models



- ▷ **Definition 22.2.15.** A naive Bayes model (or, less accurately, Bayesian classifier, or, derogatorily, idiot Bayes model) consists of:

  1. random variables $C, E_1, \ldots, E_n$ such that all the $E_1, \ldots, E_n$ are conditionally independent given C,
  2. the probability distribution $\mathbb{P}(C)$, and
  3. the conditional probability distributions $\mathbb{P}(E_i|C)$.

We call C the cause and the $E_1, \ldots, E_n$ the effects of the model.

- ▷ **Convention:** Whenever we draw a graph of random variables, we take the arrows to connect *causes* to their direct *effects*, and assert that unconnected nodes are conditionally independent given all their ancestors. We will make this more precise later.
- ▷ Can we compute the full joint probability distribution $\mathbb{P}(\mathrm{cavity}, \mathrm{toothache}, \mathrm{catch})$ from this information?

Recovering the Full Joint Probability Distribution

- ▷ **Lemma 22.2.16 (Product rule).** $\mathbb{P}(X,Y) = \mathbb{P}(X|Y) \cdot \mathbb{P}(Y)$.
- ▷ We can generalize this to more than two variables, by repeatedly applying the product rule:
- ▷ **Lemma 22.2.17 (Chain rule).** *For any sequence of random variables $X_1, \ldots, X_n$:*

$$\mathbb{P}(X_1, \ldots, X_n) = \mathbb{P}(X_1|X_2, \ldots, X_n) \cdot \mathbb{P}(X_2|X_3, \ldots X_n) \cdot \ldots \cdot \mathbb{P}(X_{n-1}|X_n) \cdot P(X_n)$$

.

Hence:

- ▷ **Theorem 22.2.18.** *Given a naive Bayes model with effects $E_1, \ldots, E_n$ and cause C, we have*

$$\mathbb{P}(C, E_1, \ldots, E_n) = \mathbb{P}(C) \cdot (\prod_{i=1}^{n} \mathbb{P}(E_i|C)).$$

- ▷ *Proof:* Using the chain rule:

1. $\mathbb{P}(E_1, \ldots, E_n, C) = \mathbb{P}(E_1|E_2, \ldots, E_n, C) \cdot \ldots \cdot \mathbb{P}(E_n|C) \cdot \mathbb{P}(C)$
2. Since all the E_i are conditionally independent, we can drop them on the right hand sides of the $\mathbb{P}(E_j|..., C)$

□

Marginalization

- ▷ Great, so now we can compute $\mathbb{P}(C|E_1, \ldots, E_n) = \frac{\mathbb{P}(C, E_1, \ldots, E_n)}{\mathbb{P}(E_1, \ldots, E_n)}$...

  ...except that we don't know $\mathbb{P}(E_1, \ldots, E_n)$:-/

  ...except that we can compute the full joint probability distribution, so we can recover it:
- ▷ **Lemma 22.2.19 (Marginalization).** *Given random variables $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_m$, we have* $\mathbb{P}(X_1, \ldots, X_n) = \sum_{y_1 \in \mathrm{dom}(Y_1), \ldots, y_m \in \mathrm{dom}(Y_m)} \mathbb{P}(X_1, \ldots, X_n, Y_1 = y_1, \ldots, Y_m = y_m)$.

  (This is just a fancy way of saying "we can add the relevant entries of the full joint probability distribution")

▷ **Example 22.2.20.** Say we observed toothache = T and catch = T. Using marginalization, we can compute

$$\begin{aligned}P(\text{cavity}|\text{toothache} \wedge \text{catch}) &= \frac{P(\text{cavity} \wedge \text{toothache} \wedge \text{catch})}{P(\text{toothache} \wedge \text{catch})} \\ &= \frac{P(\text{cavity} \wedge \text{toothache} \wedge \text{catch})}{\sum_{c \in \{\text{cavity}, \neg\text{cavity}\}} P(c \wedge \text{toothache} \wedge \text{catch})} \\ &= \frac{P(\text{cavity}) \cdot P(\text{toothache}|\text{cavity}) \cdot P(\text{catch}|\text{cavity})}{\sum_{c \in \{\text{cavity}, \neg\text{cavity}\}} P(c) \cdot P(\text{toothache}|c) \cdot P(\text{catch}|c)}\end{aligned}$$

Unknowns

▷ What if we don't know catch? (I'm not a dentist, I don't have a probe...)

▷ We split our effects into $\{E_1, \ldots, E_n\} = \{O_1, \ldots, O_{n_O}\} \cup \{U_1, \ldots, U_{n_U}\}$ – the *observed* and *unknown* random variables.

▷ Let $D_U := \mathrm{dom}(U_1) \times \ldots \times \mathrm{dom}(U_{n_u})$. Then

$$\begin{aligned}\mathbb{P}(C|O_1, \ldots, O_{n_O}) &= \frac{\mathbb{P}(C, O_1, \ldots, O_{n_O})}{\mathbb{P}(O_1, \ldots, O_{n_O})} \\ &= \frac{\sum_{u \in D_U} \mathbb{P}(C, O_1, \ldots, O_{n_O}, U_1 = u_1, \ldots, U_{n_u} = u_{n_u})}{\sum_{c \in \mathrm{dom}(C)} \sum_{u \in D_U} \mathbb{P}(O_1, \ldots, O_{n_O}, C = c, U_1 = u_1, \ldots, U_{n_u} = u_{n_u})} \\ &= \frac{\sum_{u \in D_U} \mathbb{P}(C) \cdot (\prod_{i=1}^{n_O} \mathbb{P}(O_i|C)) \cdot (\prod_{j=1}^{n_U} \mathbb{P}(U_j = u_j|C))}{\sum_{c \in \mathrm{dom}(C)} \sum_{u \in D_U} P(C = c) \cdot (\prod_{i=1}^{n_O} \mathbb{P}(O_i|C = c)) \cdot (\prod_{j=1}^{n_U} P(U_j = u_j|C = c))} \\ &= \frac{\mathbb{P}(C) \cdot (\prod_{i=1}^{n_O} \mathbb{P}(O_i|C)) \cdot (\sum_{u \in D_U} \prod_{j=1}^{n_U} \mathbb{P}(U_j = u_j|C))}{\sum_{c \in \mathrm{dom}(C)} P(C = c) \cdot (\prod_{i=1}^{n_O} \mathbb{P}(O_i|C = c)) \cdot (\sum_{u \in D_U} \prod_{j=1}^{n_U} P(U_j = u_j|C = c))}\end{aligned}$$

...oof...

Unknowns

▷ Continuing from above:

$$\mathbb{P}(C|O_1, \ldots, O_{n_O}) = \frac{\mathbb{P}(C) \cdot (\prod_{i=1}^{n_O} \mathbb{P}(O_i|C)) \cdot (\sum_{u \in D_U} \prod_{j=1}^{n_U} \mathbb{P}(U_j = u_j|C))}{\sum_{c \in \mathrm{dom}(C)} P(C = c) \cdot (\prod_{i=1}^{n_O} \mathbb{P}(O_i|C = c)) \cdot (\sum_{u \in D_U} \prod_{j=1}^{n_U} P(U_j = u_j|C = c))}$$

▷ First, note that $\sum_{u \in D_U} \prod_{j=1}^{n_U} P(U_j = u_j|C = c) = 1$ (We're summing over all possible events on the (conditionally independent) $U_1, \ldots, U_{n_U}$ given $C = c$)

▷

$$\mathbb{P}(C|O_1, \ldots, O_{n_O}) = \frac{\mathbb{P}(C) \cdot (\prod_{i=1}^{n_O} \mathbb{P}(O_i|C))}{\sum_{c \in \mathrm{dom}(C)} P(C = c) \cdot (\prod_{i=1}^{n_O} \mathbb{P}(O_i|C = c))}$$

▷ Secondly, note that the *denominator* is

1. the same for any given observations $O_1,\ldots,O_{n_O}$, independent of the value of C, and
2. the *sum* over all the *numerators* in the full distribution.

That is: The denominator only serves to *scale* what is *almost* already the distribution $\mathbb{P}(C|O_1,\ldots,O_{n_O})$ to sum up to 1.

Normalization

▷ **Definition 22.2.21 (Normalization).** Given a vector $w := \langle w_1,\ldots,w_k\rangle$ of numbers in $[0,1]$ where $\sum_{i=1}^{k} w_i \leq 1$.

Then the normalized vector $\alpha(w)$ is defined (component-wise) as

$$(\alpha(w))_i := \frac{w_i}{\sum_{j=1}^{k} w_j}.$$

Note that $\sum_{i=1}^{k} \alpha(w)_i = 1$, i.e. $\alpha(w)$ is a probability distribution.

▷ This finally gives us:

Theorem 22.2.22 (Inference in a Naive Bayes model). *Let $C, E_1,\ldots,E_n$ a naive Bayes model and $E_1,\ldots,E_n = O_1,\ldots,O_{n_O}, U_1,\ldots,U_{n_U}$.*

Then

$$\mathbb{P}(C|O_1 = o_1,\ldots,O_{n_O} = o_{n_O}) = \alpha(\mathbb{P}(C) \cdot (\prod_{i=1}^{n_O} \mathbb{P}(O_i = o_i|C)))$$

▷ Note, that this is entirely independent of the *unknown* random variables $U_1,\ldots,U_{n_U}$!

▷ Also, note that this is just a fancy way of saying "first, compute all the numerators, then divide all of them by their sums".

Dentistry Example

▷ Putting things together, we get:

$$\begin{aligned}\mathbb{P}(\text{cavity}|\text{toothache} = \mathsf{T}) &= \alpha(\mathbb{P}(\text{cavity}) \cdot \mathbb{P}(\text{toothache} = \mathsf{T}|\text{cavity})) \\ &= \alpha(\langle P(\text{cavity}) \cdot P(\text{toothache}|\text{cavity}), P(\neg\text{cavity}) \cdot P(\text{toothache}|\neg\text{cavity})\rangle)\end{aligned}$$

▷ Say we have $P(\text{cavity}) = 0.1$, $P(\text{toothache}|\text{cavity}) = 0.8$, and $P(\text{toothache}|\neg\text{cavity}) = 0.05$. Then

$$\mathbb{P}(\text{cavity}|\text{toothache} = \mathsf{T}) = \alpha(\langle 0.1 \cdot 0.8, 0.9 \cdot 0.05\rangle) = \alpha(\langle 0.08, 0.045\rangle)$$

$0.08 + 0.045 = 0.125$, hence

$$\mathbb{P}(\text{cavity}|\text{toothache} = \mathsf{T}) = \langle \frac{0.08}{0.125}, \frac{0.045}{0.125} \rangle = \langle 0.64, 0.36 \rangle$$

Naive Bayes Classification

We can use a naive Bayes model as a very simple *classifier*:

▷ Assume we want to classify newspaper articles as one of the categories *politics*, *sports*, *business*, *fluff*, etc. based on the words they contain.

▷ Given a large set of articles, we can determine the relevant probabilities by counting the occurrences of the categories $\mathbb{P}(\text{category})$, and of words per category – i.e. $\mathbb{P}(\text{word}_i|\text{category})$ for some (huge) list of words $(\text{word}_i)_{i=1}^n$.

▷ We assume that the occurrence of each word is conditionally independent of the occurrence of any other word given the category of the document. (This assumption is clearly wrong, but it makes the model simple and often works well in practice.) (⇒ "Idiot Bayes model")

▷ Given a new article, we just count the occurrences k_i of the words in it and compute

$$\mathbb{P}(\text{category}|\text{word}_1 = k_1, \ldots, \text{word}_n = k_n) = \alpha(\mathbb{P}(\text{category}) \cdot (\prod_{i=1}^{n} \mathbb{P}(\text{word}_i = k_i|\text{category})))$$

▷ We then choose the category with the highest probability.

22.2.3 Inference by Enumeration

Inference by Enumeration

▷ The rules we established for naive Bayes models, i.e. Bayes's theorem, the product rule and chain rule, marginalization and normalization, are *general* techniques for probabilistic reasoning, and their usefulness is not limited to the naive Bayes models.

▷ More generally:

▷ **Theorem 22.2.23.** *Let $Q, E_1, \ldots, E_{n_E}, U_1, \ldots, U_{n_U}$ be random variables and $D := \text{dom}(U_1) \times \ldots \times \text{dom}(U_{n_U})$. Then*

$$\mathbb{P}(Q|E_1 = e_1, \ldots, E_{n_E} = e_{n_e}) = \alpha(\sum_{u} D\mathbb{P}(Q, E_1 = e_1, \ldots, E_{n_E} = e_{n_e}, U_1 = u_1, \ldots, U_{n_U} = u_{n_U}))$$

.

*We call Q the **query variable**, $E_1, \ldots, E_{n_E}$ the **evidence**, and $U_1, \ldots, U_{n_U}$ the **unknown** (or **hidden**) **variables**, and computing a conditional probability this way **enumeration**.*

▷ Note that this is just a "mathy" way of saying we

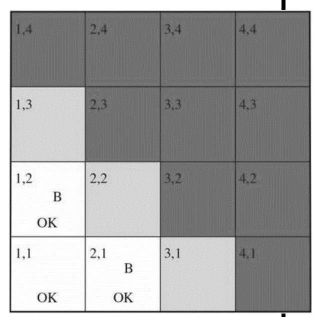
1. sum over all relevant entries of the full joint probability distribution of the variables, and
2. normalize the result to yield a probability distribution.

22.2.4 Example – The Wumpus is Back

We will fortify our intuition about naive Bayes models with a variant of the Wumpus world we looked at ??? to understand whether logic was up to the job of guiding an agent in the Wumpus cave.

Example: The Wumpus is Back

- ▷ We have a maze where
  - ▷ Every cell except $[1,1]$ possibly contains a *pit*, with 20% probability.
  - ▷ pits cause a *breeze* in neighboring cells (we forget the wumpus and the gold for now)



- ▷ Where should the agent go, if there is a breeze at $[1,2]$ and $[2,1]$?
- ▷ Pure logical inference can conclude nothing about which square is *most likely* to be safe!

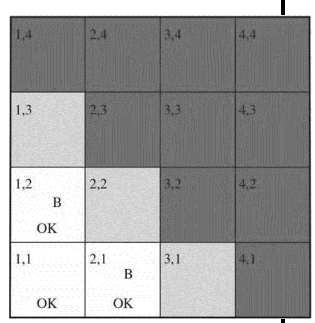
We can model this using the Boolean random variables:

- ▷ $P_{i,j}$ for $i,j \in \{1,2,3,4\}$, stating there is a pit at square $[i,j]$, and
- ▷ $B_{i,j}$ for $(i,j) \in \{(1,1),(1,2),(2,1)\}$, stating there is a breeze at square $[i,j]$

⇒ let's apply our machinery!

Wumpus: Probabilistic Model

- ▷ **First:** Let's try to compute the full joint probability distribution $\mathbb{P}(P_{1,1},\ldots,P_{4,4},B_{1,1},B_{1,2},B_{2,1})$.
  1. By the product rule, this is equal to $\mathbb{P}(B_{1,1},B_{1,2},B_{2,1}|P_{1,1},\ldots,P_{4,4}) \cdot \mathbb{P}(P_{1,1},\ldots,P_{4,4})$.
  2. Note that $\mathbb{P}(B_{1,1},B_{1,2},B_{2,1}|P_{1,1},\ldots,P_{4,4})$ is either 1 (if all the $B_{i,j}$ are consistent with the positions of the pits $P_{k,l}$) or 0 (otherwise).
  3. Since the pits are spread independently, we have $\mathbb{P}(P_{1,1},\ldots,P_{4,4}) = \prod_{i,j=1,1}^{4,4} \mathbb{P}(P_{i,j})$

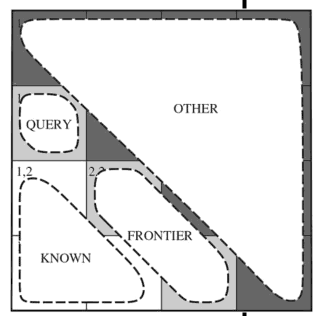


- ▷ ⇝ We know all of these probabilities.
- ▷ ⇝ We can now use enumeration to compute $\mathbb{P}(P_{i,j}|<known>) = \alpha(\sum_{<unknowns>} \mathbb{P}(P_{i,j},<known>,<unknowns>))$

Wumpus Continued

- **Problem:** We only know $P_{i,j}$ for three fields. If we want to compute e.g. $P_{1,3}$ via enumeration, that leaves $2^{4^2-4} = 4096$ terms to sum over!
- **Let's do better.**
  - Let $b := \neg B_{1,1} \wedge B_{1,2} \wedge B_{2,1}$ (All the breezes we know about)
  - Let $p := \neg P_{1,1} \wedge \neg P_{1,2} \wedge \neg P_{2,1}$. (All the pits we know about)
  - Let $F := \{P_{3,1} \wedge P_{2,2}, \neg P_{3,1} \wedge P_{2,2}, P_{3,1} \wedge \neg P_{2,2}, \neg P_{3,1} \wedge P_{2,2}\}$ (the current *"frontier"*)
  - Let O be (the set of assignments for) all the other variables $P_{i,j}$. (i.e. except p, F and our query $P_{1,3}$)

  Then the observed breezes b are conditionally independent of O given p and F. (Whether there is a pit anywhere else does not influence the breezes we observe.)



- $\Rightarrow P(b|P_{1,3}, p, O, F) = P(b|P_{1,3}, p, F)$. Let's exploit this!

Optimized Wumpus

- In particular:

$$\mathbb{P}(P_{1,3}|p,b) = \alpha(\sum_{o \in O, f \in F} \mathbb{P}(P_{1,3}, b, p, f, o)) = \alpha(\sum_{o \in O, f \in F} P(b|P_{1,3}, p, o, f) \cdot \mathbb{P}(P_{1,3}, p, f, o))$$

$$= \alpha(\sum_{f \in F}\sum_{o \in O} P(b|P_{1,3}, p, f) \cdot \mathbb{P}(P_{1,3}, p, f, o)) = \alpha(\sum_{f \in F} P(b|P_{1,3}, p, f) \cdot (\sum_{o \in O} \mathbb{P}(P_{1,3}, p, f, o)))$$

$$= \alpha(\sum_{f \in F} P(b|P_{1,3}, p, f) \cdot (\sum_{o \in O} \mathbb{P}(P_{1,3}) \cdot P(p) \cdot P(f) \cdot P(o)))$$

$$= \alpha(\mathbb{P}(P_{1,3}) \cdot P(p) \cdot (\sum_{f \in F} \underbrace{P(b|P_{1,3}, p, f)}_{\in \{0,1\}} \cdot P(f) \cdot (\underbrace{\sum_{o \in O} P(o)}_{=1})))$$

$\rightsquigarrow$ this is just a sum over the frontier, i.e. 4 terms ☺

- So: $\mathbb{P}(P_{1,3}|p,b) = \alpha(\langle 0.2 \cdot (0.8)^3 \cdot (1 \cdot 0.04 + 1 \cdot 0.16 + 1 \cdot 0.16 + 0), 0.8 \cdot (0.8)^3 \cdot (1 \cdot 0.04 + 1 \cdot 0.16 + 0 + 0)\rangle) \approx \langle 0.31, 0.69\rangle$
- Analogously: $\mathbb{P}(P_{3,1}|p,b) = \langle 0.31, 0.69\rangle$ and $\mathbb{P}(P_{2,2}|p,b) = \langle 0.86, 0.14\rangle$ ($\Rightarrow$ avoid $[2,2]$!)

Cooking Recipe

▷ In general, when you want to reason probabilistically, a good heuristic is:

1. Try to frame the full joint probability distribution in terms of the probabilities you know. Exploit product rule/chain rule, independence, conditional independence, marginalization **and domain knowledge** (as e.g. $\mathbb{P}(b|p,f) \in \{0,1\}$)

⇒ the problem can be solved at all!

2. **Simplify**: Start with the equation for enumeration:

$$\mathbb{P}(Q|E_1, ...) = \alpha(\sum_{u \in U} \mathbb{P}(Q, E_1, ..., U_1 = u_1, ...))$$

3. Substitute by the result of 1., and again, exploit all of our machinery
4. Implement the resulting (system of) equation(s)
5. ???
6. Profit

Summary

▷ Probability distributions and conditional probability distributions allow us to represent random variables as convenient datastructures in an implementation (Assuming they are finite domain...)

▷ The full joint probability distribution allows us to compute all probabilities of statements about the random variables contained (But possibly inefficient)

▷ Marginalization and normalization are the specific techniques for extracting the *specific* probabilities we are interested in from the full joint probability distribution.

▷ The product and chain rule, exploiting (conditional) independence, Bayes' Theorem, and of course *domain specific* knowledge allow us to do so much more efficiently.

▷ Naive Bayes models are one example where all these techniques come together.

Chapter 23

Probabilistic Reasoning: Bayesian Networks

23.1 Introduction

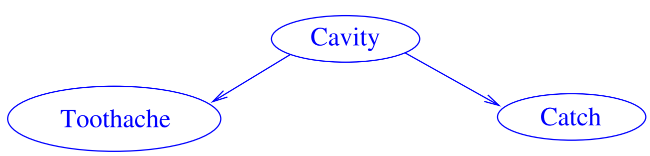
Reminder: Our Agenda for This Topic

▷ Our treatment of the topic "probabilistic reasoning" consists of this and last section.

- ▷ ???: All the basic machinery at use in Bayesian networks.
- ▷ **This section**: Bayesian networks: What they are, how to build them, how to use them.
- ▷ The most wide-spread and successful practical framework for probabilistic reasoning.

Reminder: Our Machinery

1. **Graph captures variable dependencies:** (Variables $X_1, \ldots, X_n$)



   ▷ Given evidence $\mathbf{e}$, want to know $\mathbf{P}(X|e)$. Remaining vars: $\mathbf{Y}$.

2. **Normalization+Marginalization:**

$$\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}\in\mathbf{Y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$$

   ▷ A sum over atomic events!

3. **Chain rule:** $X_1, \ldots, X_n$ ordered consistently with dependency graph.

$$\mathbf{P}(X_1, \ldots, X_n) = \mathbf{P}(X_n|X_{n-1}, \ldots, X_1) \cdot \mathbf{P}(X_{n-1}|X_{n-2}, \ldots, X_1) \cdot \ldots \cdot \mathbf{P}(X_1)$$

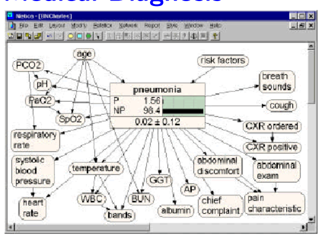
4. **Exploit conditional independence:** Instead of $\mathbf{P}(X_i|X_{i-1},\ldots,X_1)$, we can use $\mathbf{P}(X_i|\mathrm{Parents}(X_i))$.

   ▷ Bayesian networks!

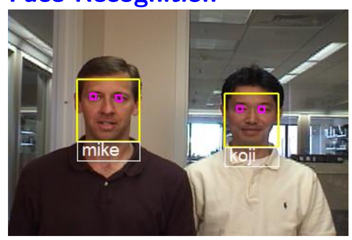
Some Applications

▷ A ubiquitous problem: Observe "symptoms", need to infer "causes".

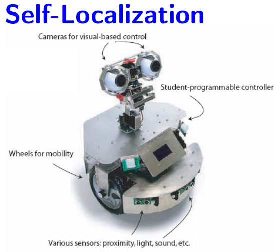
Medical Diagnosis



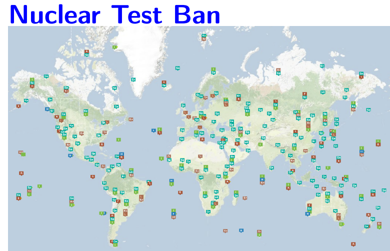
Face Recognition



Self-Localization



Nuclear Test Ban



Our Agenda for This Chapter

▷ **What is a Bayesian Network?:** i.e. What is the syntax?

  ▷ Tells you what Bayesian networks look like.

▷ **What is the Meaning of a Bayesian Network?:** What is the semantics?

  ▷ Makes the intuitive meaning precise.

▷ **Constructing Bayesian Networks:** How do we design these networks? What effect do our choices have on their size?

  ▷ Before you can start doing inference, you need to model your domain.

▷ **Inference in Bayesian Networks:** How do we use these networks? What is the associated complexity?

  ▷ Inference is our primary purpose. It is important to understand its complexities and how it can be improved.

23.2 What is a Bayesian Network?

What is a Bayesian Network? (Short: BN)

- What do the others say?
  - "*A Bayesian network is a methodology for representing the full joint probability distribution. In some cases, that representation is compact.*"
  - "*A Bayesian network is a graph whose nodes are random variables X_i and whose edges $\langle X_j, X_i \rangle$ denote a direct influence of X_j on X_i. Each node X_i is associated with a conditional probability table (CPT), specifying $\mathbf{P}(X_i|\mathrm{Parents}(X_i))$.*"
  - "*A Bayesian network is a graphical way to depict conditional independence relations within a set of random variables.*"
- A Bayesian network (BN) represents the structure of a given domain. Probabilistic inference exploits that structure for improved efficiency.
- BN inference: Determine the distribution of a query variable X given observed evidence $\mathbf{e}$: $\mathbf{P}(X|\mathbf{e})$.

John, Mary, and My Brand-New Alarm

- **Example 23.2.1 (From Russell/Norvig).**
  - I got very valuable stuff at home. So I bought an alarm. Unfortunately, the alarm just rings at home, doesn't call me on my mobile.
  - I've got two neighbors, Mary and John, who'll call me if they hear the alarm.
  - The problem is that, sometimes, the alarm is caused by an earthquake.
  - Also, John might confuse the alarm with his telephone, and Mary might miss the alarm altogether because she typically listens to loud music.
- **Question:** Given that both John and Mary call me, what is the probability of a burglary?

John, Mary, and My Alarm: Designing the Network

- **Cooking Recipe:**

  (1) Design the random variables $X_1, \ldots, X_n$;

  (2) Identify their dependencies;

  (3) Insert the conditional probability tables $\mathbf{P}(X_i|\mathrm{Parents}(X_i))$.

- **Example 23.2.2 (Let's cook!).** Using this recipe on Example 23.2.1, ...

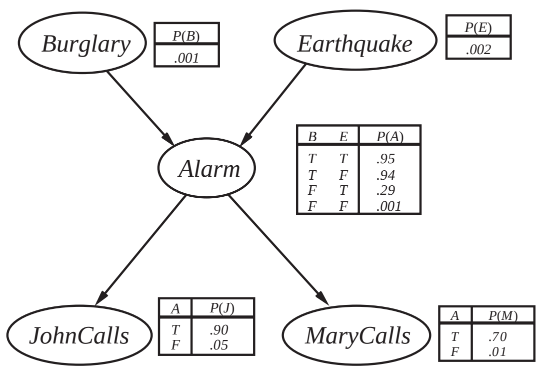
  (1) Random variables: $\mathrm{Burglary}$, $\mathrm{Earthquake}$, Alarm, $\mathrm{JohnCalls}$, $\mathrm{MaryCalls}$.

(2) **Dependencies**: Burglaries and earthquakes are independent. (this is actually debatable $\leadsto$ design decision!)
The alarm might be activated by either. John and Mary call if and only if they hear the alarm. (they don't care about earthquakes)

(3) **Conditional probability tables**: Assess the probabilities, see next slide.

John, Mary, and My Alarm: The Bayesian network

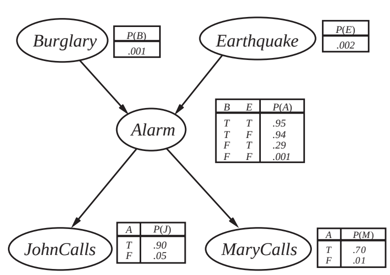
▷ **Example 23.2.3.** Continuing Example 23.2.2 we obtain



▷ **Note:** In each $\mathbf{P}(X_i|\text{Parents}(X_i))$, we show only $\mathbf{P}(X_i = \mathsf{T}|\text{Parents}(X_i))$. We don't show $\mathbf{P}(X_i = \mathsf{F}|\text{Parents}(X_i))$ which is $1 - \mathbf{P}(X_i = \mathsf{T}|\text{Parents}(X_i))$.

The Syntax of Bayesian Networks

▷ To fix the exact definition of Bayesian networks recall the ???:



▷ **Definition 23.2.4 (Bayesian Network).** Given random variables $X_1, \ldots, X_n$ with finite domains $D_1, \ldots, D_n$, a Bayesian network (also belief network or probabilistic network) is a node labeled DAG $\mathcal{B} := \langle\{X_1, \ldots, X_n\}, E, \text{CPT}\rangle$.

Each X_i is labeled with a function

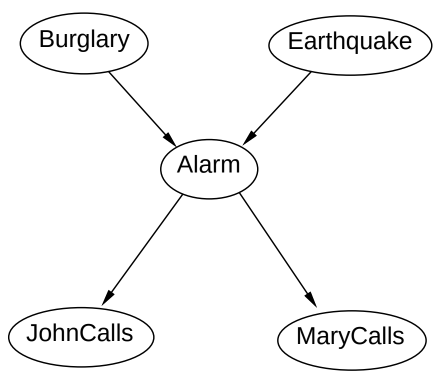
$$\mathrm{CPT}(X_i)\colon D_i \times \prod_{X_j \in \mathrm{Parents}(X_i)} D_j \to [0,1]$$

where $\mathrm{Parents}(X_i) := \{X_j \mid (X_j, X_i) \in E\}$ it is called the conditional probability table at X_i.

▷ **Definition 23.2.5.** Bayesian networks and related formalisms summed up under the term graphical models.

23.3 What is the Meaning of a Bayesian Network?

The Semantics of Bayesian Networks: Illustration



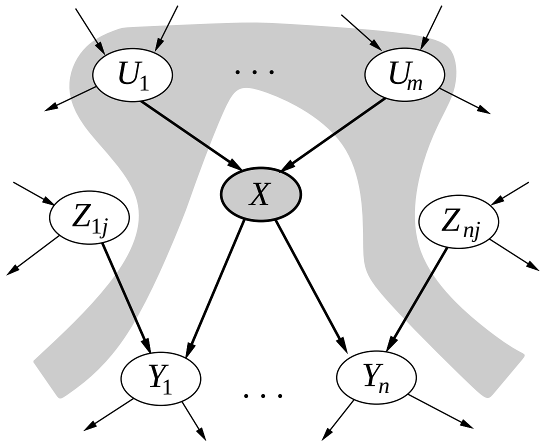
▷ Alarm depends on Burglary and Earthquake.

▷ MaryCalls only depends on Alarm. $\mathbf{P}(\mathrm{MaryCalls}|\mathrm{Alarm}, \mathrm{Burglary}) = \mathbf{P}(\mathrm{MaryCalls}|\mathrm{Alarm})$

▷ Bayesian networks represent sets of conditional independence assumptions.

The Semantics of Bayesian Networks: Illustration, ctd.

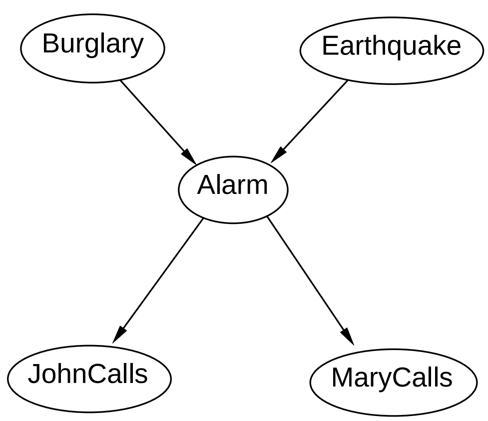
▷ **Observation 23.3.1.** *Each node X in a BN is conditionally independent of its non-descendants given its parents $\mathrm{Parents}(X)$.*



▷ **Question:** Why non-descendants of X?

▷ **Intuition:** Given that BNs are acyclic, these are exactly those nodes that *could* have an edge into X.

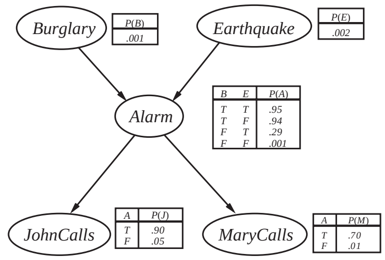
The Semantics of BNs



▷ **Question:** Given the value of Alarm, MaryCalls is independent of?

▷ **Answer:** reserved for the plenary sessions ⇝ be there!

The Semantics of Bayesian Networks: Formal



- **Definition 23.3.2.** Let $\langle\mathcal{X}, E\rangle$ be a Bayesian network, $X \in \mathcal{X}$, and E^* the reflexive transitive closure of E, then $\mathrm{NonDesc}(X) := \{Y \mid (X,Y) \notin E^*\} \backslash \mathrm{Parents}(X)$ is the set of non-descendents of X.

- **Definition 23.3.3.** Given a Bayesian network $\mathcal{B} := \langle\mathcal{X}, E\rangle$, we identify $\mathcal{B}$ with the following two assumptions:

  (A) $X \in \mathcal{X}$ is conditionally independent of $\mathrm{NonDesc}(X)$ given $\mathrm{Parents}(X)$.

  (B) For all values x of $X \in \mathcal{X}$, and all value combinations of $\mathrm{Parents}(X)$, we have $P(x|\mathrm{Parents}(X)) = \mathrm{CPT}(x, \mathrm{Parents}(X))$.

Recovering the Full Joint Probability Distribution

- **Intuition:** A Bayesian network is a methodology for representing the full joint probability distribution.

- **Problem:** How to recover the full joint probability distribution $\mathbf{P}(X_1, \ldots, X_n)$ from $\mathcal{B} := \langle\{X_1, \ldots, X_n\}, E\rangle$?

- **Chain Rule:** For *any* variable ordering $X_1, \ldots, X_n$, we have:

$$\mathbf{P}(X_1, \ldots, X_n) = \mathbf{P}(X_n|X_{n-1}, \ldots, X_1) \cdot \mathbf{P}(X_{n-1}|X_{n-2}, \ldots, X_1) \cdot \ldots \cdot \mathbf{P}(X_1)$$

  Choose $X_1, \ldots, X_n$ consistent with $\mathcal{B}$: $X_j \in \mathrm{Parents}(X_i) \rightsquigarrow j < i$.

- **Observation 23.3.4 (Exploiting Conditional Independence).** *With ??? (A), we can use* $\mathbf{P}(X_i|\mathrm{Parents}(X_i))$ *instead of* $\mathbf{P}(X_i|X_{i-1}, \ldots, X_1)$*:*

$$\mathbf{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i|\mathrm{Parents}(X_i))$$

  The distributions $\mathbf{P}(X_i|\mathrm{Parents}(X_i))$ *are given by ??? (B).*

- Same for atomic events $P(X_1, \ldots, X_n)$.

- **Observation 23.3.5 (Why "acyclic"?).** *For cyclic $\mathcal{B}$, this does NOT hold, indeed cyclic BNs may be self contradictory.* (*need a consistent ordering*)

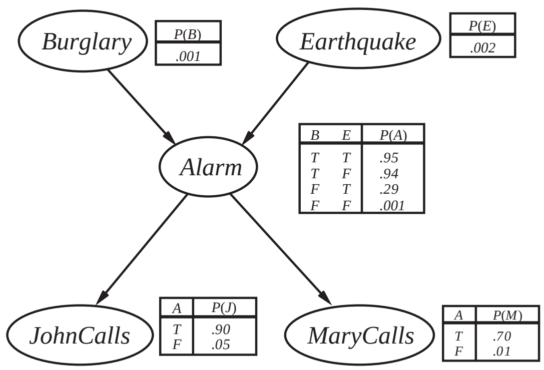
Note: If there is a cycle, then any variable ordering $X_1, \ldots, X_n$ will not be consistent with the BN; so in the chain rule on $X_1, \ldots, X_n$ there comes a point where we have $\mathbf{P}(X_i|X_{i-1}, \ldots, X_1)$ in the chain but $\mathbf{P}(X_i|\mathrm{Parents}(X_i))$ in the definition of distribution, and $\mathrm{Parents}(X_i) \not\subseteq \{X_{i-1}, \ldots, X_1\}$

but then the products are different. So the chain rule can no longer be used to prove that we can reconstruct the full joint probability distribution. In fact, cyclic Bayesian network contain ambiguities (several interpretations possible) and may be self-contradictory (no probability distribution matches the Bayesian network).

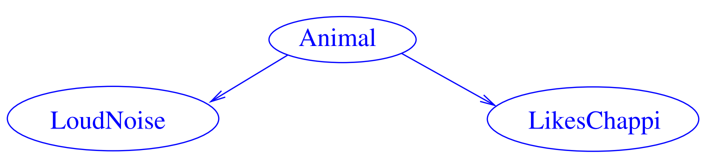
Recovering a Probability for John, Mary, and the Alarm

▷ **Example 23.3.6.** John and Mary called because there was an alarm, but no earthquake or burglary

$$\begin{aligned} P(j,m,a,\neg b,\neg e) &= P(j|a)\cdot P(m|a)\cdot P(a|\neg b,\neg e)\cdot P(\neg b)\cdot P(\neg e) \\ &= 0.9*0.7*0.001*0.999*0.998 \\ &= 0.00062 \end{aligned}$$



Meaning of Bayesian Networks

▷ 

Say $\mathcal{B}$ is the Bayesian network above. Which statements are correct?

(A) Animal is independent of LikesChappi.

(B) LoudNoise is independent of LikesChappi.

(C) Animal is conditionally independent of LikesChappi given LoudNoise.

(D) LikesChappi is conditionally independent of LoudNoise given Animal.

Think about this intuitively: Given both values for variable X, is the chances of Y being true higher for one of these (fixing value of third var where specified)?

▷ **Answers:** reserved for the plenary sessions $\rightsquigarrow$ be there!

23.4 Constructing Bayesian Networks

Reducing Edges: Variable Order Matters

Given a set of random variables $X_1, \ldots, X_n$, consider the following (impractical, but illustrative) pseudo-algorithm for constructing a Bayesian network:

▷ **Definition 23.4.1 (BN construction algorithm).**

1. Initialize $BN := \langle\{X_1, \ldots, X_n\}, E\rangle$ where $E = \emptyset$.
2. Fix any variable ordering, $X_1, \ldots, X_n$.
3. **for** $i := 1, \ldots, n$ **do**
   a. Choose a minimal set $\mathrm{Parents}(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$ such that

   $$\mathbb{P}(X_i|X_{i-1}, \ldots, X_1) = \mathbb{P}(X_i|\mathrm{Parents}(X_i))$$

   b. For each $X_j \in \mathrm{Parents}(X_i)$, insert (X_j, X_i) into E.
   c. Associate X_i with $\mathbb{P}(X_i|\mathrm{Parents}(X_i))$.

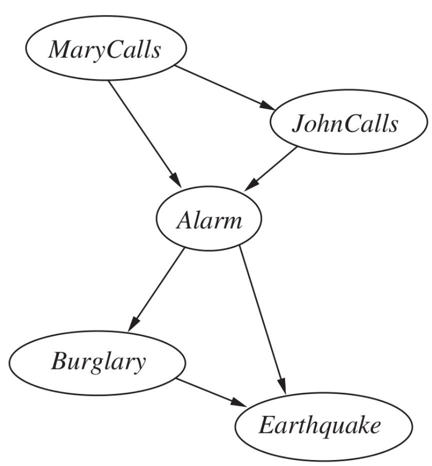
▷ **Attention:** *Which* variables we need to include into $\mathrm{Parents}(X_i)$ depends on what "$\{X_1, \ldots, X_{i-1}\}$" is . . . !

▷ **Thus:** The size of the resulting BN depends on the chosen variable ordering $X_1, \ldots, X_n$.

▷ **In Particular:** The size of a Bayesian network is *not* a fixed property of the domain. It depends on the skill of the designer.

John and Mary Depend on the Variable Order!

▷ **Example 23.4.2.** `Mary`, `John`, `Alarm`, `Burglary`, `Earthquake`.

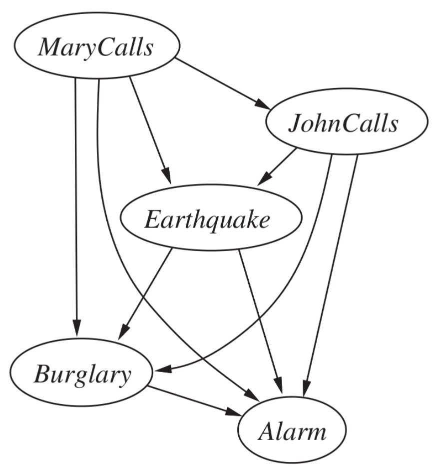


Note: For ??? we try to determine whether – given different value assignments to potential parents – the probability of X_i being true differs? If yes, we include these parents. In the particular case:

1. M to J yes because the common cause may be the alarm.
2. M, J to A yes because they may have heard alarm.
3. A to B yes because if A then higher chance of B.
4. However, M/J to B no because M/J only react to the alarm so if we have the value of A then values of M/J don't provide more information about B.
5. A to E yes because if A then higher chance of E.
6. B to E yes because, if A and not B then chances of E are higher than if A and B.

John and Mary Depend on the Variable Order! Ctd.

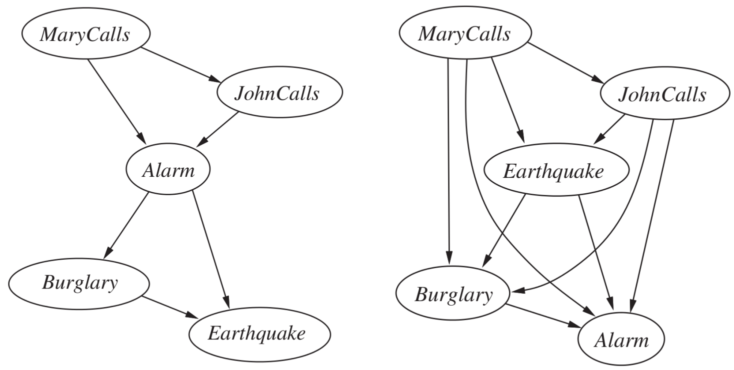
▷ **Example 23.4.3.** `Mary`, `John`, `Earthquake`, `Burglary`, `Alarm`.



Again: Given different value assignments to potential parents, does the probability of X_i being true differ? If yes, include these parents.

1. M to J as before.
2. M, J to E as probability of E is higher if M/J is true.
3. Same for B; E to B because, given M and J are true, if E is true as well then prob of B is lower than if E is false.
4. $M/J/B/E$ to A because if $M/J/B/E$ is true (even when changing the value of just one of these) then probability of A is higher.

John and Mary, What Went Wrong?



- **Intuition:** These BNs link from *effects* to their *causes*!

  ⇒ Even though `Mary` and `John` are conditionally independent given `Alarm`, this is not exploited, since `Alarm` is not ordered before `Mary` and `John`!

⇒ **Rule of Thumb:** We should order causes before symptoms.

Compactness of Bayesian Networks

- **Definition 23.4.4.** Given random variables $X_1, \ldots, X_n$ with finite domains $D_1, \ldots, D_n$, the size of $\mathcal{B} := \langle \{X_1, \ldots, X_n\}, E \rangle$ is defined as

$$\mathrm{size}(\mathcal{B}) := \sum_{i=1}^{n} |D_i| \cdot \Big(\prod_{X_j \in \mathrm{Parents}(X_i)} |D_j| \Big)$$

- **Note:** $\mathrm{size}(\mathcal{B}) \hat{=}$ The total number of entries in the conditional probability distributions.
- **Note:** Smaller BN $\rightsquigarrow$ need to assess less probabilities, more efficient inference.
- **Observation 23.4.5.** *Explicit full joint probability distribution has size* $\prod_{i=1}^{n} |D_i|$.
- **Observation 23.4.6.** *If* $|\mathrm{Parents}(X_i)| \leq k$ *for every* X_i, *and* $D_{\max}$ *is the largest random variable domain, then* $\mathrm{size}(\mathcal{B}) \leq n |D_{\max}|^{k+1}$.
- **Example 23.4.7.** For $|D_{\max}| = 2$, $n = 20$, $k = 4$ we have $2^{20} = 1048576$ probabilities, but a Bayesian network of size $\leq 20 \cdot 2^5 = 640 \ldots$!
- In the *worst case*, $\mathrm{size}(\mathcal{B}) = n \cdot (\prod_{=i}^{1} n)|D_i|$, namely if every variable depends on all its predecessors in the chosen variable ordering.
- **Intuition:** BNs are compact – i.e. of small size – if each variable is directly influenced only by few of its predecessor variables.

Keeping Networks Small

To keep our Bayesian networks small, we can:

1. **Reduce the number of edges:** $\Rightarrow$ Order the variables to allow for exploiting conditional independence (causes before effects), or
2. **represent the conditional probability distributions efficiently:**
   (a) For Boolean random variables X, we only need to store $\mathbf{P}(X = \mathsf{T}|\mathrm{Parents}(X))$ ($\mathbf{P}(X = \mathsf{F}|\mathrm{Parents}(X)) = 1 - \mathbf{P}(X = \mathsf{T}|\mathrm{Parents}(X))$) (Cuts the number of entries in half!)
   (b) Introduce different **kinds** of nodes exploiting domain knowledge; e.g. deterministic and noisy disjunction nodes.

Constructing Bayesian Networks

▷ **Question:** What is the Bayesian network we get by constructing according to the ordering

1. $X_1 = \mathrm{LoudNoise}, X_2 = \mathrm{Animal}, X_3 = \mathrm{LikesChappi}$?
2. $X_1 = \mathrm{LoudNoise}, X_2 = \mathrm{LikesChappi}, X_3 = \mathrm{Animal}$?

▷ **Answer:** reserved for the plenary sessions $\rightsquigarrow$ be there!

23.5 Modeling Simple Dependencies

Representing Conditional Distributions: Deterministic Nodes

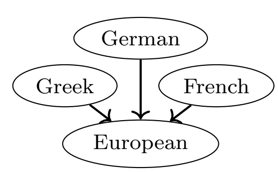
▷ **Problem:** Even if $max(\mathrm{Parents})$ is small, the CPT has 2^k entries. (worst-case)

▷ **Idea:** Usually CPTs follow standard patterns called canonical distributions.

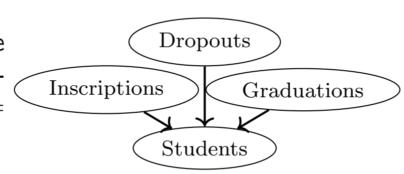
▷ only need to determine pattern and some values.

▷ **Definition 23.5.1.** A node X in a Bayesian network is called deterministic, if its value is completely determined by the values of $\mathrm{Parents}(X)$.

▷ **Example 23.5.2 (Logical Dependencies).** In the network on the right, the node $European$ is deterministic, the CPT corresponds to a logical disjunction, i.e. $P(\mathrm{european}) = P(\mathrm{greek} \vee \mathrm{german} \vee \mathrm{french})$.



▷ **Example 23.5.3 (Numerical Dependencies).** In the network on the right, the node $Students$ is deterministic, the CPT corresponds to a sum, i.e. $P(S = i - d - g) = P(I = i) + P(D = d) + P(G = g)$.

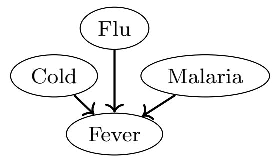


▷ **Intuition:** Deterministic nodes model direct, causal relationships.

Representing Conditional Distributions: Noisy Nodes

- **Problem:** Sometimes, values of nodes are only "almost deterministic". (uncertain, but mostly logical)
- **Idea:** Use "noisy" logical relationships. (generalize logical ones softly to $[0,1]$)
- **Example 23.5.4 (Inhibited Causal Dependencies).**

  In the network on the right, deterministic disjunction for the node Fever is incorrect, since the diseases sometimes fail to develop fever. The causal relation between parent and child is inhibited.



- **Assumptions:** We make the following assumptions for modeling Example 23.5.4:

  1. Cold, Flu, and $\mathrm{Malaria}$ is a complete list of fever causes (add a leak node for the others otherwise).
  2. Inhibitions of the parents are independent.

  Thus we can model the inhibitions by individual inhibition factors q_d.
- **Definition 23.5.5.** The CPT of a noisy disjunction node X in a Bayesian network is given by $P(X_i|\mathrm{Parents}(X_i)) = 1 - (\prod_{\{j \mid X_j = \mathsf{T}\}} q_j)$, where the q_i are the inhibition factors of $X_i \in \mathrm{Parents}(X)$.

Representing Conditional Distributions: Noisy Nodes

- **Example 23.5.6.** We have the following inhibition factors for Example 23.5.4:

$$\begin{aligned}
q_{\mathrm{cold}} &= P(\neg\mathrm{fever}|\mathrm{cold}, \neg\mathrm{flu}, \neg\mathrm{malaria}) = 0.6\\
q_{\mathrm{flu}} &= P(\neg\mathrm{fever}|\neg\mathrm{cold}, \mathrm{flu}, \neg\mathrm{malaria}) = 0.2\\
q_{\mathrm{malaria}} &= P(\neg\mathrm{fever}|\neg\mathrm{cold}, \neg\mathrm{flu}, \mathrm{malaria}) = 0.1
\end{aligned}$$

If we model Fever as a noisy disjunction node, then the general rule $P(X_i|\mathrm{Parents}(X_i)) = \prod_{\{j \mid X_j = \mathsf{T}\}} q_j$ for the CPT gives the following table:

Cold	Flu	Malaria	$P(\mathrm{Fever})$	$P(\neg\mathrm{Fever})$
F	F	F	0.0	1.0
F	F	T	0.9	**0.1**
F	T	F	0.8	**0.2**
F	T	T	0.98	$0.02 = 0.2 \cdot 0.1$
T	F	F	0.4	**0.6**
T	F	T	0.94	$0.06 = 0.6 \cdot 0.1$
T	T	F	0.88	$0.12 = 0.6 \cdot 0.2$
T	T	T	0.988	$0.012 = 0.6 \cdot 0.2 \cdot 0.1$

Representing Conditional Distributions: Noisy Nodes

▷ **Observation 23.5.7.** *In general, noisy logical relationships in which a variable depends on k parents can be described by $\mathcal{O}(k)$ parameters instead of $\mathcal{O}(2^k)$ for the full conditional probability table. This can make assessment (and learning) tractable.*

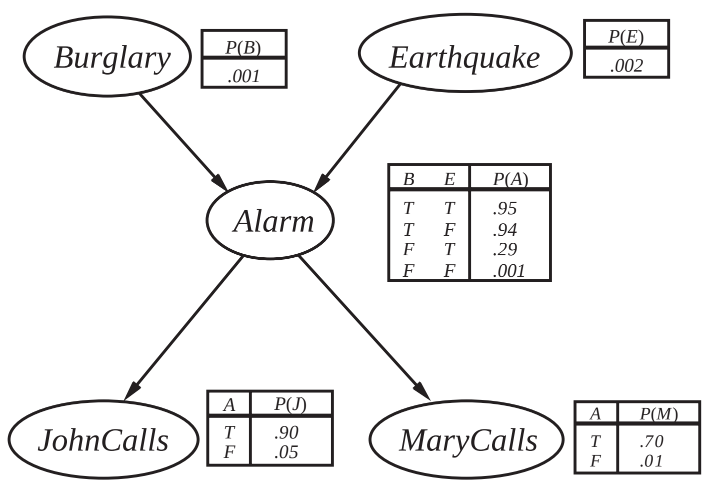
▷ **Example 23.5.8.** The CPCS network [Pra+94] uses noisy-OR and noisy-MAX distributions to model relationships among diseases and symptoms in internal medicine. With 448 nodes and 906 links, it requires only 8,254 values instead of 133,931,430 for a network with full CPTs.

23.6 Inference in Bayesian Networks

Inference for Mary and John

▷ **Intuition:** Observe evidence variables and draw conclusions on query variables.

▷ **Example 23.6.1.**



- ▷ What is $\mathbf{P}(\text{Burglary}|\text{johncalls})$?
- ▷ What is $\mathbf{P}(\text{Burglary}|\text{johncalls}, \text{marycalls})$?

Probabilistic Inference Tasks in Bayesian Networks

▷ **Definition 23.6.2 (Probabilistic Inference Task).** Let $X_1, \ldots, X_n$ be a set of random variables, a probabilistic inference task consists of

- a set $\mathbf{X} \subseteq \{X_1, \ldots, X_n\}$ of query variables,
- a set $\mathbf{E} \subseteq \{X_1, \ldots, X_n\}$ of evidence variables, and
- an event $\mathbf{e}$ that assigns values to $\mathbf{E}$.

We wish to compute the conditional probability distribution $\mathbf{P}(\mathbf{X}|\mathbf{e})$.
Variables in $\mathbf{Y} := \{X_1, \ldots, X_n\} \backslash (\mathbf{X} \cup \mathbf{E})$ are called hidden variables.

▷ **Notes:**

- We assume that a Bayesian network $\mathcal{B}$ for $X_1, \ldots, X_n$ is given.
- In the remainder, for simplicity, $\mathbf{X} = \{X\}$ is a singleton.

▷ **Example 23.6.3.** In $\mathbf{P}(\mathrm{Burglary}|\mathrm{johncalls}, \mathrm{marycalls})$, $X = \mathrm{Burglary}$, $\mathbf{e} = \mathrm{johncalls}, \mathrm{marycalls}$, and $\mathbf{Y} = \{\mathrm{Alarm}, \mathrm{EarthQuake}\}$.

Inference by Enumeration: The Principle (A Reminder!)

▷ **Problem:** Given evidence $\mathbf{e}$, want to know $\mathbb{P}(X|\mathbf{e})$.
Hidden variables: $\mathbf{Y}$.

▷ **1. Bayesian network:** Construct a Bayesian network $\mathcal{B}$ that captures variable dependencies.

▷ **2. Normalization+Marginalization:**

$$\mathbb{P}(X|\mathbf{e}) = \alpha\mathbb{P}(X, \mathbf{e});\text{ if } \mathbf{Y} \neq \emptyset \text{ then } \mathbb{P}(X|\mathbf{e}) = \alpha(\sum_{\mathbf{y}\in\mathbf{Y}} \mathbb{P}(X, \mathbf{e}, \mathbf{y}))$$

▷ Recover the summed-up probabilities $\mathbb{P}(X, \mathbf{e}, \mathbf{y})$ from $\mathcal{B}$!

▷ **3. Chain Rule:** Order $X_1, \ldots, X_n$ consistent with $\mathcal{B}$.

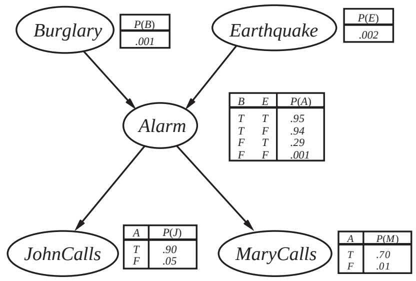
$$\mathbb{P}(X_1, \ldots, X_n) = \mathbb{P}(X_n|X_{n-1}, \ldots, X_1) \cdot \mathbb{P}(X_{n-1}|X_{n-2}, \ldots, X_1) \cdot \ldots \cdot \mathbb{P}(X_1)$$

▷ **4. Exploit conditional independence:** Instead of $\mathbb{P}(X_i|X_{i-1}, \ldots, X_1)$, use $\mathbb{P}(X_i|\mathrm{Parents}(X_i))$.

▷ Given a Bayesian network $\mathcal{B}$, probabilistic inference tasks can be solved as sums of products of conditional probabilities from $\mathcal{B}$.

▷ Sum over all value combinations of hidden variables.

Inference by Enumeration: John and Mary



- **Want:** $\mathbb{P}(\mathrm{Burglary}|\mathrm{johncalls},\mathrm{marycalls})$.
  Hidden variables: $\mathbf{Y}=\{\mathrm{Earthquake},\mathrm{Alarm}\}$.

- **Normalization+Marginalization:**

$$\mathbb{P}(B|j,m)=\alpha\mathbb{P}(B,j,m)=\alpha(\sum_{v_E}\sum_{v_A}\mathbb{P}(B,j,m,v_E,v_A))$$

- **Order:** $X_1=B$, $X_2=E$, $X_3=A$, $X_4=J$, $X_5=M$.

- **Chain rule and conditional independence:**

$$\mathbb{P}(B|j,m)=\alpha(\sum_{v_E}\sum_{v_A}\mathbb{P}(B)\cdot P(v_E)\cdot\mathbb{P}(v_A|B,v_E)\cdot P(j|v_A)\cdot P(m|v_A))$$

Inference by Enumeration: John and Mary, ctd.

- **Move variables outwards** until we hit the first parent:

$$\mathbb{P}(B|j,m)=\alpha\cdot\mathbb{P}(B)\cdot(\sum_{v_E}P(v_E)\cdot(\sum_{v_A}\mathbb{P}(v_A|B,v_E)\cdot P(j|v_A)\cdot P(m|v_A)))$$

  Note: This step *is* actually done by the pseudo-code, implicitly in the sense that in the recursive calls to enumerate-all we multiply our own prob with all the rest. That is valid because, the variable ordering being consistent, all our parents are already here which is just another way of saying "my own prob does not depend on the variables in the rest of the order".

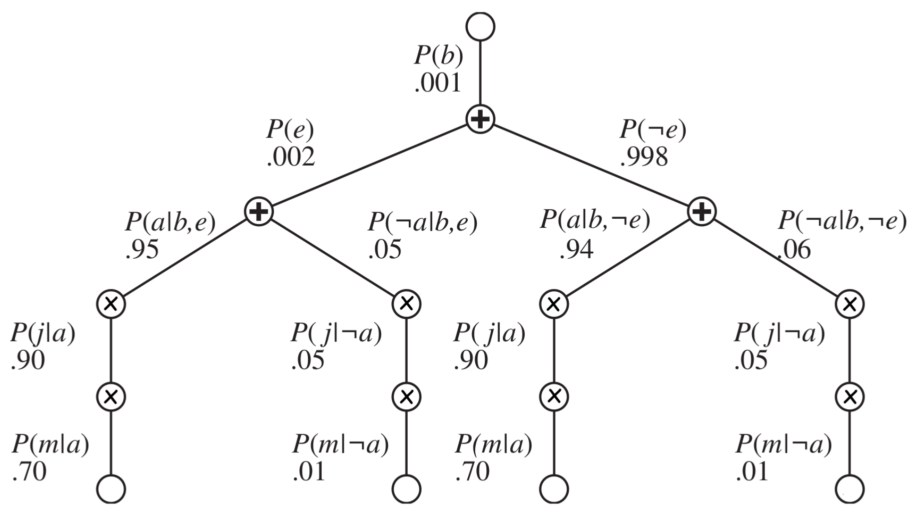
- The probabilities of the outside-variables multiply the entire "rest of the sum"

▷ **Chain rule and conditional independence, ctd.:**

$$
\begin{aligned}
& \mathbb{P}(B|j,m) \\
= \; & \alpha \mathbb{P}(B)(\sum_{v_E} P(v_E)(\sum_{v_A} \mathbb{P}(v_A|B,v_E)P(j|v_A)P(m|v_A))) \\
= \; & \alpha \cdot P(b) \cdot \left(\begin{array}{l}
P(e) \cdot \left(\begin{array}{l} \overbrace{P(a|b,e)P(j|a)P(m|a)}^{a} \\ + \; \underbrace{P(\neg a|b,e)P(j|\neg a)P(m|\neg a)}_{\neg a} \end{array} \right) \Big\} e \\
+ \; P(\neg e) \cdot \left(\begin{array}{l} \overbrace{P(a|b,\neg e)P(j|a)P(m|a)}^{a} \\ + \; \underbrace{P(\neg a|b,\neg e)P(j|\neg a)P(m|\neg a)}_{\neg a} \end{array} \right) \Big\} \neg e
\end{array} \right) \\
= \; & \alpha\langle 0.00059224, 0.0014919\rangle \approx \langle 0.284, 0.716\rangle
\end{aligned}
$$

This computation can be viewed as a "search tree"! (see next slide)

The Evaluation of $P(b|j,m)$, as a "Search Tree"



▷ Inference by enumeration = a tree with "sum nodes" branching over values of hidden variables, and with non-branching "multiplication nodes".

Inference by Enumeration: Variable Elimination

▷ **Inference by Enumeration:**

- ▹ Evaluates the tree in a depth-first manner.
- ▹ space complexity: linear in the number of variables.

- ▷ time complexity: exponential in the number of hidden variables, e.g. $\mathcal{O}(2^{\#(\mathbf{Y})})$ in case these variables are Boolean.

▷ Can we do better than this?

▷ **Definition 23.6.4.** Variable elimination is a BNI algorithm that avoids

- ▷ repeated computation, and (see below)
- ▷ irrelevant computation. (see below)

▷ In some special cases, variable elimination runs in polynomial time.

Variable Elimination: Sketch of Ideas

▷ **Avoiding repeated computation:** Evaluate expressions from right to left, storing all intermediate results.

▷ For query $P(B|j,m)$:

1. CPTs of BN yield *factors* (probability tables):

$$\mathbf{P}(B|j,m) = \alpha \cdot \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \cdot (\sum_{v_E} \underbrace{P(v_E)}_{\mathbf{f}_2(E)} \sum_{v_A} \underbrace{\mathbf{P}(v_A|B,v_E)}_{\mathbf{f}_3(A,B,E)} \cdot \underbrace{P(j|v_A)}_{\mathbf{f}_4(A)} \cdot \underbrace{P(m|v_A)}_{\mathbf{f}_5(A)})$$

2. Then the computation is performed in terms of *factor product* and *summing out variables* from factors:

$$\mathbf{P}(B|j,m) = \alpha \cdot \mathbf{f}_1(B) \cdot (\sum_{v_E} \mathbf{f}_2(E) \cdot (\sum_{v_A} \mathbf{f}_3(A,B,E) \cdot \mathbf{f}_4(A) \cdot \mathbf{f}_5(A)))$$

▷ **Avoiding irrelevant computation:** Repeatedly remove hidden variables that are leaf nodes.

▷ For query $P(\mathrm{JohnCalls}|\mathrm{burglary})$:

$$\mathbf{P}(J|b) = \alpha \cdot P(b) \cdot (\sum_{v_E} P(v_E) \cdot (\sum_{v_A} P(v_A|b,v_E) \cdot \mathbf{P}(J|v_A) \cdot (\sum_{v_M} P(v_M|v_A))))$$

- ▷ The rightmost sum equals 1 and can be dropped.

The Complexity of Exact Inference

▷ **Definition 23.6.5.** A graph G is called singly connected, or a polytree (otherwise multiply connected), if there is at most one undirected path between any two nodes in G.

▷ **Theorem 23.6.6 (Good News).** *On singly connected Bayesian networks, variable elimination runs in polynomial time.*

▷ Is our BN for Mary & John a polytree? (Yes.)

▷ **Theorem 23.6.7 (Bad News).** *For multiply connected Bayesian networks, probabilistic inference is $\#\mathbf{P}$-hard.* *($\#\mathbf{P}$ is harder than $\mathbf{NP}$, i.e. $\mathbf{NP} \subseteq \#\mathbf{P}$)*

▷ **So?:** Life goes on ... In the hard cases, if need be we can throw exactitude to the winds and approximate.

▷ **Example 23.6.8.** Sampling techniques as in MCTS.

23.7 Conclusion

Summary

▷ Bayesian networks (BN) are a wide-spread tool to model uncertainty, and to reason about it. A BN represents conditional independence relations between random variables. It consists of a graph encoding the variable dependencies, and of conditional probability tables (CPTs).

▷ Given a variable ordering, the BN is small if every variable depends on only a few of its predecessors.

▷ Probabilistic inference requires to compute the probability distribution of a set of query variables, given a set of evidence variables whose values we know. The remaining variables are hidden.

▷ Inference by enumeration takes a BN as input, then applies Normalization+Marginalization, the chain rule, and exploits conditional independence. This can be viewed as a tree search that branches over all values of the hidden variables.

▷ Variable elimination avoids unnecessary computation. It runs in polynomial time for poly-tree BNs. In general, exact probabilistic inference is $\#\mathbf{P}$-hard. Approximate probabilistic inference methods exist.

Topics We Didn't Cover Here

▷ **Inference by sampling**: A whole zoo of methods for doing this exists.

▷ **Clustering**: Pre-combining subsets of variables to reduce the running time of inference.

▷ **Compilation to SAT**: More precisely, to "weighted model counting" in CNF formulas. Model counting extends DPLL with the ability to determine the number of satisfying interpretations. Weighted model counting allows to define a mass for each such interpretation (= the probability of an atomic event).

▷ **Dynamic BN**: BN with one slice of variables at each "time step", encoding probabilistic behavior over time.

▷ **Relational BN**: BN with predicates and object variables.

▷ **First-order BN**: Relational BN with quantification, i.e. probabilistic logic. E.g., the BLOG language developed by Stuart Russel and co-workers.

Reading:

- *Chapter 14: Probabilistic Reasoning* of [RN03].
  - Section 14.1 roughly corresponds to my "What is a Bayesian Network?".
  - Section 14.2 roughly corresponds to my "What is the Meaning of a Bayesian Network?" and "Constructing Bayesian Networks".The main change I made here is to *define* the semantics of the BN in terms of the conditional independence relations, which I find clearer than RN's definition that uses the reconstructed full joint probability distribution instead.
  - Section 14.4 roughly corresponds to my "Inference in Bayesian Networks". RN give full details on variable elimination, which makes for nice ongoing reading.
  - Section 14.3 discusses how CPTs are specified in practice.
  - Section 14.5 covers approximate sampling-based inference.
  - Section 14.6 briefly discusses relational and first-order BNs.
  - Section 14.7 briefly discusses other approaches to reasoning about uncertainty.

  All of this is nice as additional background reading.

Chapter 24

Making Simple Decisions Rationally

24.1 Introduction

Overview

We now know how to update our world model, represented as (a set of) random variables, given observations. Now we need to *act*.

For that we need to answer two questions:

Questions:

- ▷ Given a world model and a set of *actions*, what will the likely consequences of each action be?
- ▷ How "good" are these consequences?

Idea:

- ▷ Represent actions as "special random variables":

  Given disjoint actions $a_1, \ldots, a_n$, introduce a random variable A with domain $\{a_1, \ldots, a_n\}$. Then we can model/query $\mathbb{P}(X|A = a_i)$.
- ▷ Assign *numerical values* to the possible outcomes of actions (i.e. a function $u\colon \mathbf{dom}(X) \to \mathbb{R}$) indicating their desirability.
- ▷ Choose the action that maximizes the *expected value* of u

Definition 24.1.1. Decision theory investigates decision problems, i.e. how a utility-based agent a deals with choosing among actions based on the desirability of their outcomes given by a real-valued utility function U on states $s \in S$: i.e. $U\colon S \to \mathbb{R}$.

Decision Theory

If our states are random variables, then we obtain a random variable for the utility function:
Observation: Let $X_i\colon \Omega \to D_i$ random variables on a probability model $\langle \Omega, P \rangle$ and $f\colon D_1 \times \ldots \times D_n \to E$. Then $F(x) := f(X_0(x), \ldots, X_n(x))$ is a random variable $\Omega \to E$.

Definition 24.1.2. Given a probability model $\langle \Omega, P \rangle$ and a random variable $X\colon \Omega \to D$ with $D \subseteq \mathbb{R}$, then $E(X) := \sum_{x \in D} P(X = x) \cdot x$ is called the expected value (or expectation) of X.

(Assuming the sum/series is actually defined!)

Analogously, let $e_1, \ldots, e_n$ a sequence of events. Then the expected value of X given $e_1, \ldots, e_n$ is defined as $E(X|e_1, \ldots, e_n) := \sum_{x \in D} P(X = x|e_1, \ldots, e_n) \cdot x$.

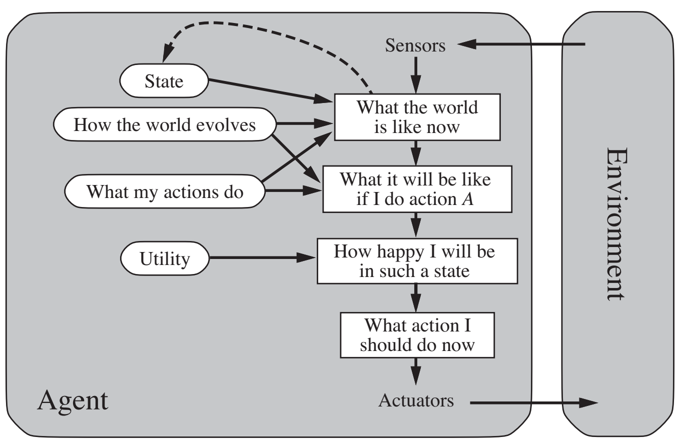
Putting things together:

Definition 24.1.3. Let $A\colon \Omega \to D$ a random variable (where D is a set of actions) $X_i\colon \Omega \to D_i$ random variables (the state), and $U\colon D_1 \times \ldots \times D_n \to \mathbb{R}$ a utility function. Then the expected utility of the action $a \in D$ is the expected value of U (interpreted as a random variable) given $A = a$; i.e.

$$\mathrm{EU}(a) := \sum_{\langle x_1, \ldots, x_n \rangle \in D_1 \times \ldots \times D_n} P(X_1 = x_1, \ldots, X_n = x_n | A = a) \cdot U(x_1, \ldots, x_n)$$

Utility-based Agents

- **Definition 24.1.4.** A utility-based agent uses a world model along with a utility function that models its preferences among the states of that world. It chooses the action that leads to the best expected utility.
- **Agent Schema:**



Maximizing Expected Utility (Ideas)

Definition 24.1.5 (MEU principle for Rationality). We call an action rational if it maximizes expected utility (MEU). An utility-based agent is called rational, iff it always chooses a rational action.

Hooray: This solves all of AI. (in principle)

Problem: There is a long, long way towards an operationalization ;)

Note: An agent can be entirely rational (consistent with MEU) without ever representing or manipulating utilities and probabilities.

Example 24.1.6. A reflex agent for tic tac toe based on a perfect lookup table is rational if we take (the negative of) "winning/drawing in n steps" as the utility function.

Example 24.1.7 (AI1). Heuristics in tree search (greedy search, A^*) and game-play (minimax, alpha-beta pruning) maximize "expected" utility.

⇒ In fully observable, deterministic environments, "expected utility" reduces to a specific determined utility value:

$\mathrm{EU}(a) = U(T(S(s,e),a))$, where e the most recent percept, s the current state, S the sensor function and T the transition function.

Now let's figure out how to actually assign utilities!

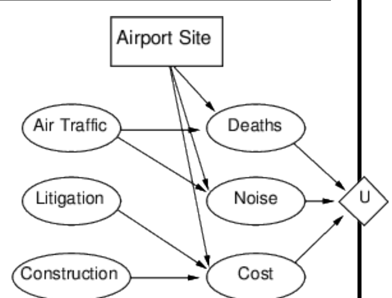
24.2 Decision Networks

Now that we understand multi-attribute utility functions, we can complete our design of a utility-based agent, which we now recapitulate as a refresher. As we already use Bayesian networks for the belief state of an utility-based agent, integrating utilities and possible actions into the network suggests itself naturally. This leads to the notion of a decision network.

Decision networks

Definition 24.2.1. A decision network is a Bayesian network with two additional kinds of nodes:

- ▷ action nodes, representing a set of possible actions, and (square nodes)
- ▷ A single utility node (also called value node). (diamond node)



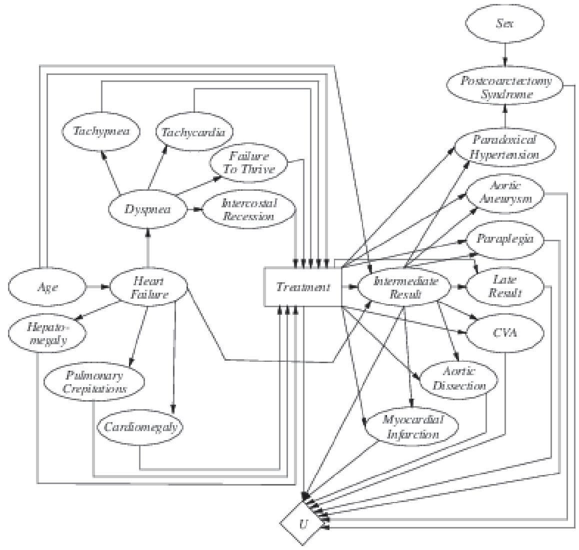
General Algorithm: Given evidence $E_j = e_j$, and action nodes $A_1,\ldots,A_k$, compute the expected utility of each action, given the evidence, i.e. return the sequence of actions

$$\operatorname{argmax} a_1,\ldots,a_k \overbrace{\sum_{\langle x_1,\ldots,x_n\rangle} \underbrace{P(X_i = x_i | A_1 = a_1,\ldots,A_k = a_k, E_j = e_j)}_{\text{usual Bayesian Network inference}} \cdot U(X_i = x_i)}^{=\text{expected utility of } a_1,\ldots,a_k}$$

Note the sheer amount of summands in the sum above in the general case! (⇒ We will simplify where possible later)

Decision Networks: Example

- ▷ **Example 24.2.2 (A Decision-Network for Aortic Coarctation).** from [Luc96]



24.3 Preferences and Utilities

Preferences in Deterministic Environments

Problem: How do we determine the utility of a state? (We cannot directly measure our satisfaction/happiness in a possibly future state...) (What unit would we even use?)

Example 24.3.1. I have to decide whether to go to class today (or sleep in). What is the utility of this lecture? (obviously 42)

Idea: We can let people/agents choose between two states (subjective preference) and derive a utility from these choices.

Example 24.3.2. *Give me your cell-phone or I will give you a bloody nose.* $\leadsto$
To make a decision in a deterministic environment, the agent must determine whether it prefers a state without phone to one with a bloody nose?

Definition 24.3.3. Given states A and B (we call them prizes) an agent can express preferences of the form

- ▷ $A \succ B$ A prefered over B
- ▷ $A \sim B$ indifference between A and B
- ▷ $A \succeq B$ B not prefered over A

i.e. Given a set $\mathcal{S}$ (of states), we define binary relations $\succ$ and $\sim$ on $\mathcal{S}$.

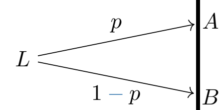
Preferences in Non-Deterministic Environments

Problem: In nondeterministic environments we do not have full information about the states we choose between.

Example 24.3.4 (Airline Food). *Do you want chicken or pasta* (but we cannot see through the tin foil)

Definition 24.3.5.

Let $\mathcal{S}$ a set of states. We call a random variable X with domain $\{A_1, \ldots, A_n\} \subseteq \mathcal{S}$ a lottery and write $[p_1,A_1\,;\ldots;\,p_n,A_n]$, where $p_i = P(X = A_i)$.



Idea: A lottery represents the result of a nondeterministic action that can have outcomes A_i with prior probability p_i. For the binary case, we use $[p,A;1-p,B]$. We can then extend preferences to include lotteries, as a measure of how *strongly* we prefer one prize over another.

Convention: We assume $\mathcal{S}$ to be *closed under lotteries*, i.e. lotteries themselves are also states. That allows us to consider lotteries such as $[p,A;1-p,[q,B;1-q,C]]$.

Rational Preferences

Note: Preferences of a rational agent must obey certain constraints – An agent with *rational* preferences can be described as an MEU-agent.

Definition 24.3.6. We call a set $\succ$ of preferences rational, iff the following constraints hold:

Orderability	$A \succ B \vee B \succ A \vee A \sim B$
Transitivity	$A \succ B \wedge B \succ C \Rightarrow A \succ C$
Continuity	$A \succ B \succ C \Rightarrow (\exists p.[p,A;1-p,C] \sim B)$
Substitutability	$A \sim B \Rightarrow [p,A;1-p,C] \sim [p,B;1-p,C]$
Monotonicity	$A \succ B \Rightarrow ((p > q) \Leftrightarrow [p,A;1-p,B] \succ [q,A;1-q,B])$
Decomposability	$[p,A;1-p,[q,B;1-q,C]] \sim [p,A\,;\,((1-p)q),B\,;\,((1-p)(1-q)),C]$

From a set of rational preferences, we can obtain a meaningful utility function.

The rationality constraints can be understood as follows:

Orderability: $A \succ B \vee B \succ A \vee A \sim B$ Given any two prizes or lotteries, 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. Refusing to bet is like refusing to allow time to pass.

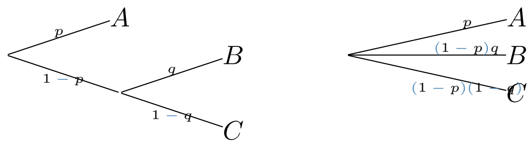
Transitivity: $A \succ B \wedge B \succ C \Rightarrow A \succ C$

Continuity: $A \succ B \succ C \Rightarrow (\exists p.[p,A;1-p,C] \sim B)$ If some lottery B is between A and C in preference, then there is some probability p for which the rational agent will be indifferent between getting B for sure and the lottery that yields A with probability p and C with probability $1-p$.

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

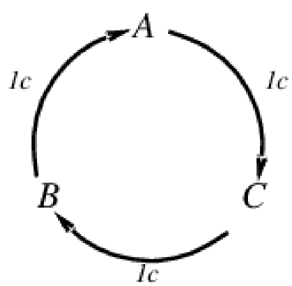
Monotonicity: $A \succ B \Rightarrow ((p > q) \Leftrightarrow [p,A;1-p,B] \succ [q,A;1-q,B])$ Suppose two lotteries have the same two possible outcomes, A and B. If an agent prefers A to B, then the agent must prefer the lottery that has a higher probability for A (and vice versa).

Decomposability: $[p,A;1-p,[q,B;1-q,C]] \sim [p,A;((1-p)q),B;((1-p)(1-q)),C]$ 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: the following two are equivalent:



Rational preferences contd.

- Violating the rationality constraints from ??? leads to self-evident irrationality.
- **Example 24.3.7.** An agent with intransitive preferences can be induced to give away all its money:
  - If $B \succ C$, then an agent who has C would pay (say) 1 cent to get B
  - If $A \succ B$, then an agent who has B would pay (say) 1 cent to get A
  - If $C \succ A$, then an agent who has A would pay (say) 1 cent to get C



24.4 Utilities

Ramseys Theorem and Value Functions

- **Theorem 24.4.1.** *(Ramsey, 1931; von Neumann and Morgenstern, 1944)*
  Given a rational set of preferences there exists a real valued function U such that $U(A) \geq U(B)$, iff $A \succeq B$ and $U([p_1,S_1 ; \ldots ; p_n,S_n]) = \sum_i p_i U(S_i)$
- This is an existence theorem, uniqueness not guaranteed.
- **Note:** Agent behavior is *invariant* w.r.t. positive linear transformations, i.e. an agent with utility function $U'(x) = k_1 U(x) + k_2$ where $k_1 > 0$ behaves exactly like one with U.
- **Observation:** With deterministic prizes only (no lottery choices), only a total ordering on prizes can be determined.
- **Definition 24.4.2.** We call a total ordering on states a value function or ordinal utility function. (If we don't need to care about *relative* utilities of states, e.g. to compute non-trivial expected utilities, that's all we need anyway!)

Utilities

- **Intuition:** Utilities map states to real numbers.
- **Question:** Which numbers exactly?
- **Definition 24.4.3 (Standard approach to assessment of human utilities).** Compare a given state A to a standard lottery L_p that has
  - "best possible prize" $u_\top$ with probability p
  - "worst possible catastrophe" $u_\bot$ with probability $1 - p$

  adjust lottery probability p until $A{\sim}L_p$. Then $U(A) = p$.
- **Example 24.4.4.** Choose $u_\top \mathrel{\widehat{=}}$ current state, $u_\bot \mathrel{\widehat{=}}$ instant death

pay \$30$\sim L$ $\xrightarrow{0.999999}$ continue as before

pay \$30$\sim L$ $\xrightarrow{0.000001}$ instant death

Popular Utility Functions

- **Definition 24.4.5.** Normalized utilities: $u_\top = 1$, $u_\bot = 0$.

  (Not very meaningful, but at least it's independent of the specific problem...)
- **Obviously**: *Money* (Very intuitive, often easy to determine, but actually not well-suited as a utility function (see later))
- **Definition 24.4.6.** Micromorts: one millionth chance of instant death.

  (useful for Russian roulette, paying to reduce product risks, etc.)

  But: Not necessarily a good measure of risk, if the risk is "merely" severe injury or illness. . .

  Better:
- **Definition 24.4.7.** QALYs: quality adjusted life years

  QALYs are useful for medical decisions involving substantial risk.

Comparing Utilities

Problem: What is the monetary value of a micromort?
Just ask people: What would you pay to avoid playing Russian roulette with a million-barrelled revolver? (Usually: quite a lot!)

But their behavior suggests a lower price:

- Driving in a car for $370\mathrm{km}$ incurs a risk of one micromort;
- Over the life of your car – say, $150,000\mathrm{km}$ that's 400 micromorts.

▷ People appear to be willing to pay about $10,000$€ more for a safer car that halves the risk of death. ($\rightsquigarrow$ 25€ per micromort)

This figure has been confirmed across many individuals and risk types.

Of course, this argument holds only for small risks. Most people won't agree to kill themselves for 25M€. (Also: People are pretty bad at estimating and comparing risks, especially if they are small.) (Various cognitive biases and heuristics are at work here!)

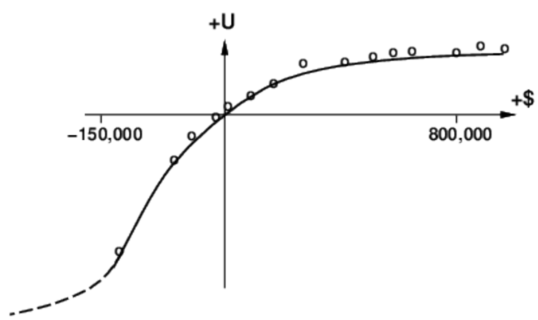
Money vs. Utility

▷ Money does *not* behave as a utility function should.

▷ Given a lottery L with expected monetary value $\mathrm{EMV}(L)$, usually $U(L) < U(\mathrm{EMV}(L))$, i.e., people are risk averse.

▷ **Utility curve:** For what probability p am I indifferent between a prize x and a lottery $[p,M\$;1-p,0\$]$ for large numbers M?

▷ Typical empirical data, extrapolated with risk prone behavior for debitors:



▷ **Empirically:** Comes close to the logarithm on the natural numbers.

24.5 Multi-Attribute Utility

In this section we will make the ideas introduced above more practical. The discussion above conceived utility functions as functions on atomic states, which were good enough for introducing the theory. But when we build decision models for utility-based agent we want to characterize states by attributes that are already random variables in the Bayesian network we use to represent the belief state. For factored states, the utility function can be expressed as a multivariate function on attribute values.

Utility Functions on Attributes

Recap: So far we understand how to obtain utility functions $u\colon S \to \mathbb{R}$ on states $s \in S$ from (rational) preferences.

But in practice, our actions often impact *multiple* distinct "attributes" that need to be weighed against each other.

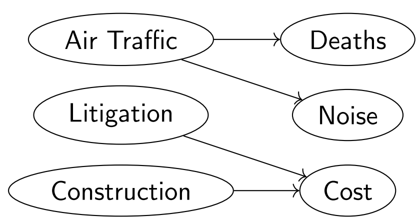
⇒ Lotteries become complex very quickly

Definition 24.5.1. Let $X_1, \ldots, X_n$ be random variables with domains $D_1, \ldots, D_n$. Then we call a function $u: D_1 \times \ldots \times D_n \to \mathbb{R}$ a (multi-attribute) utility function on attributes $X_1, \ldots, X_n$.

Note: In the general (worst) case, a multi-attribute utility function on n random variables with domain sizes k each requires k^n parameters to represent.
But: A utility function on multiple attributes often has "internal structure" that we can exploit to simplify things.
For example, the distinct attributes are often "independent" with respect to their utility (a higher-quality product is better than a lower-quality one that costs the same, and a cheaper product is better than an expensive one of the same quality)

Multi-Attribute Utility: Example

▷ **Example 24.5.2 (Assessing an Airport Site).**



▷ Attributes: Deaths, Noise, Cost.

▷ **Question**: What is $U(\text{Deaths}, \text{Noise}, \text{Cost})$ for a projected airport?

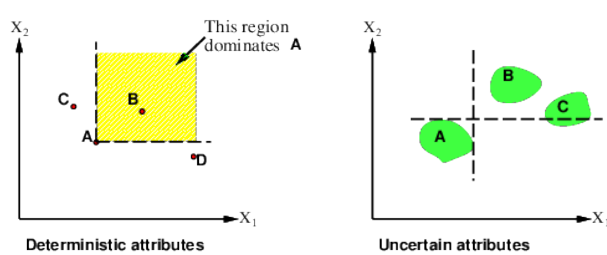
▷ How can complex utility function be assessed from preference behaviour?

▷ **Idea 1:** Identify conditions under which decisions can be made without complete identification of $U(X_1, \ldots, X_n)$.

▷ **Idea 2:** Identify various types of *independence* in preferences and derive consequent canonical forms for $U(X_1, \ldots, X_n)$.

Strict Dominance

First Assumption: U is often *monotone* in each argument. (wlog. growing)
Definition 24.5.3. *(Informally)* An action B strictly dominates an action A, iff every possible outcome of B is at least as good as every possible outcome of A,



If A strictly dominates B, we can just ignore B entirely.

Observation: Strict dominance seldom holds in practice (life is difficult) but is useful for narrowing down the field of contenders.

Stochastic Dominance

Definition 24.5.4. Let X_1, X_2 distributions with domains $\subseteq \mathbb{R}$.

X_1 stochastically dominates X_2 iff for all $t \in \mathbb{R}$, we have $P(X_1 \geq t) \geq P(X_2 \geq t)$, and for some t, we have $P(X_1 \geq t) > P(X_2 \geq t)$.

Observation 24.5.5. *If U is monotone in X_1, and $\mathbb{P}(X_1|a)$ stochastically dominates $\mathbb{P}(X_1|b)$ for actions a, b, then a is always the better choice than b, with all other attributes X_i being equal.*

$\Rightarrow$ If some action $\mathbb{P}(X_i|a)$ stochastically dominates $\mathbb{P}(X_i|b)$ for all attributes X_i, we can ignore b.

Observation: Stochastic dominance can often be determined without exact distributions using *qualitative* reasoning.

Example 24.5.6 (Construction cost increases with distance). If airport location S_1 is closer to the city than $S_2 \rightsquigarrow S_1$ stochastically dominates S_2 on cost.q

We have seen how we can do inference with attribute-based utility functions, let us consider the computational implications. We observe that we have just replaced one evil – exponentially many states (in terms of the attributes) – by another – exponentially many parameters of the utility functions.

Wo we do what we always do in AI-2: we look for structure in the domain, do more theory to be able to turn such structures into computationally improved representations.

Preference structure: Deterministic

- **Recall:** In deterministic environments an agent has a value function.
- **Definition 24.5.7.** X_1 and X_2 preferentially independent of X_3 iff preference between $\langle x_1, x_2, z\rangle$ and $\langle x'_1, x'_2, z\rangle$ does not depend on z. (i.e. the tradeoff between x_1 and x_2 is independent of z)
- **Example 24.5.8.** E.g., $\langle \text{Noise}, \text{Cost}, \text{Safety}\rangle$: are preferentially independent $\langle$20,000 suffer, 4.6 G\$, 0.06 deaths/mpm$\rangle$ vs. $\langle$70,000 suffer, 4.2 G\$, 0.06 deaths/mpm$\rangle$
- **Theorem 24.5.9 (Leontief, 1947).** *If every pair of attributes is preferentially independent of its complement, then every subset of attributes is preferentially independent of its complement: mutual preferential independence.*
- **Theorem 24.5.10 (Debreu, 1960).** *Mutual preferential independence implies that there is an additive value function: $V(S) = \sum_i V_i(X_i(S))$, where V_i is a value function referencing just one variable X_i.*
- Hence assess n single-attribute functions. (often a good approximation)
- **Example 24.5.11.** The value function for the airport decision might be

$$V(noise, cost, deaths) = -noise \cdot 10^4 - cost - deaths \cdot 10^{12}$$

Preference structure: Stochastic

Definition 24.5.12. $\mathbf{X}$ is utility independent of $\mathbf{Y}$ iff preferences over lotteries in $\mathbf{X}$ do not depend on particular values in $\mathbf{Y}$

Definition 24.5.13. A set $\mathbf{X}$ is mutually utility independent (MUI), iff each subset is utility independent of its complement.

Theorem 24.5.14. *For a MUI set of attributes $\mathcal{X}$, there is a multiplicative utility function of the form:* [Kee74]

$$U = \sum_{(\{X_0,\ldots,X_k\}\subseteq\mathcal{X})} \prod_{i=1}^{k} U_i(X_i = x_i)$$

⇒ *U can be represented using n single-attribute utility functions.*

System Support: Routine procedures and software packages for generating preference tests to identify various canonical families of utility functions.

Decision networks - Improvements

Ways to improve inference in decision networks:

- ▷ Exploit "inner structure" of the utility function to simplify the computation,
- ▷ eliminate dominated actions,
- ▷ label pairs of nodes with *stochastic dominance*: If (the utility of) some attribute dominates (the utility of) another attribute, focus on the dominant one (e.g. if price is always more important than quality, ignore quality whenever the price between two choices differs)
- ▷ various techniques for variable elimination,
- ▷ policy iteration (more on that when we talk about Markov decision procedures)

24.6 The Value of Information

So far we have tacitly been concentrating on actions that directly affect the environment. We will now come to a type of action we have hypothesized in the beginning of the course, but have completely ignored up to now: information gathering actions.

What if we do not have all information we need?

We now know how to exploit the information we have to make decisions. But if we knew more, we might be able to make even better decisions in the long run - potentially at the cost of gaining utility. (exploration vs. exploitation)

Example 24.6.1 (Medical Diagnosis).

- ▷ We do not expect a doctor to already know the results of the diagnostic tests when the patient comes in.
- ▷ Tests are often expensive, and sometimes hazardous. (directly or by delaying treatment)

- ▷ **Therefore**: Only test, if
  - ▹ knowing the results lead to a significantly better treatment plan,
  - ▹ information from test results is not drowned out by a-priori likelihood.

Definition 24.6.2. Information value theory is concerned with agent making decisions on information gathering rationally.

Value of Information by Example

Idea: Compute the expected *gain in utility* from acquring information.
Example 24.6.3 (Buying Oil Drilling Rights). There are n blocks of drilling rights available, exactly one block actually has oil worth k€, in particular:

- ▷ The prior probability of a block having oil is $\frac{1}{n}$ each (mutually exclusive).
- ▷ The current price of each block is $\frac{k}{n}$€.
- ▷ A "consultant" offers an accurate survey of block (say) 3. How much should we be willing to pay for the survey?

Solution: Compute the expected value of the best action given the information, minus the expected value of the best action without information.
Example 24.6.4 (Oil Drilling Rights contd.).

- ▷ Survey may say *oil in block 3 with probability* $\frac{1}{n}$ $\leadsto$ we buy block 3 for $\frac{k}{n}$€ and make a profit of $(k-\frac{k}{n})$€.
- ▷ Survey may say *no oil in block 3 with probability* $\frac{n-1}{n}$ $\leadsto$ we buy another block, and make an expected profit of $\frac{k}{n-1}-\frac{k}{n}$€.
- ▷ Without the survery, the expected profit is 0
- ▷ Expected profit is $\frac{1}{n}\cdot\frac{(n-1)k}{n}+\frac{n-1}{n}\cdot\frac{k}{n(n-1)}=\frac{k}{n}$.
- ▷ So, we should pay up to $\frac{k}{n}$€ for the information. (as much as block 3 is worth!)

General formula (VPI)

Definition 24.6.5. Let A the set of available actions and F a random variable. Given evidence $E_i=e_i$, let α be the action that maximizes expected utility a priori, and α_f the action that maximizes expected utility given $F=f$, i.e.: $\alpha=\underset{a\in A}{\operatorname{argmax}}\ \mathrm{EU}(a|E_i=e_i)$ and $\alpha_f=\underset{a\in A}{\operatorname{argmax}}\ \mathrm{EU}(a|E_i=e_i,F=f)$

The value of perfect information (VPI) on F given evidence $E_i=e_i$ is defined as

$$\mathrm{VPI}_{E_i=e_i}(F):=\left(\sum_{f\in\mathbf{dom}(F)}P(F=f|E_i=e_i)\cdot\mathrm{EU}(\alpha_f|E_i=e_i,F=f)\right)-\mathrm{EU}(\alpha|E_i=e_i)$$

Intuition: The VPI is the expected gain from knowing the value of F relative to the current expected utility, and considering the relative probabilities of the possible outcomes of F.

Properties of VPI

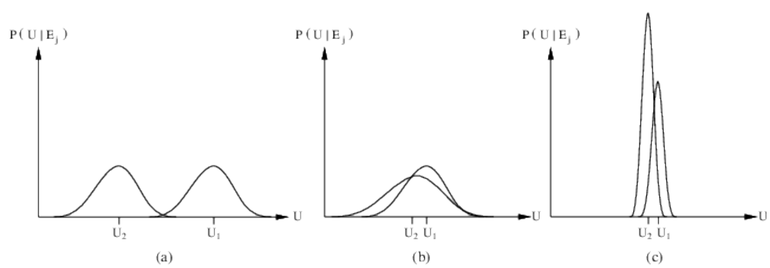
- **Observation 24.6.6 (VPI is Non-negative).** $\mathrm{VPI}_E(F) \geq 0$ *for all* j *and* E (*in* expectation, *not* post hoc)
- **Observation 24.6.7 (VPI is Non-additive).** $\mathrm{VPI}_E(F,G) \neq \mathrm{VPI}_E(F) + \mathrm{VPI}_E(G)$ (*consider, e.g., obtaining* F *twice*)
- **Observation 24.6.8 (VPI is Order-independent).**

$$\mathrm{VPI}_E(F,G) = \mathrm{VPI}_E(F) + \mathrm{VPI}_{E,F}(G) = \mathrm{VPI}_E(G) + \mathrm{VPI}_{E,G}(F)$$

- **Note:** When more than one piece of evidence can be gathered, maximizing VPI for each to select one is not always optimal $\rightsquigarrow$ evidence-gathering becomes a sequential decision problem.

Qualitative behavior of VPI

- **Question:** Say we have three distributions for $P(U|E_j)$



Qualitatively: What is the value of information (VPI) in these three cases?

- **Answers:** reserved for the plenary sessions $\rightsquigarrow$ be there!

We will now use information value theory to specialize our utility-based agent from above.

A simple Information-Gathering Agent

- **Definition 24.6.9.** A simple information gathering agent. (gathers info before acting)

function Information−Gathering−Agent (percept) **returns** an action
  persistent: D, a decision network
  integrate percept into D
  $j := \operatorname{argmax}_k \mathrm{VPI}_E(E_k)/Cost(E_k)$

if $\mathrm{VPI}_E(E_j) > Cost(E_j)$ **return** Request(E_j)
else return the best action from D

The next percept after Request(E_j) provides a value for E_j.

- **Problem:** The information gathering implemented here is myopic, i.e. only acquires a single evidence variable, or acts immediately. (cf. greedy search)
- But it works relatively well in practice. (e.g. outperforms humans for selecting diagnostic tests)
- Strategies for nonmyopic information gathering exist (Not discussed in this course)

Summary

- An MEU agent maximizes expected utility.
- Decision theory provides a framework for rational decision making.
- Decision networks augment Bayesian networks with action nodes and a utility node.
- rational preferences allow us to obtain a utility function (orderability, transitivity, continuity, substitutability, monotonicity, decomposability)
- multi-attribute utility functions can usually be "destructured" to allow for better inference and representation (can be monotone, attributes may dominate others, actions may dominate others, may be multiplicative,...)
- information value theory tells us when to explore rather than exploit, using
- VPI (value of perfect information) to determine how much to "pay" for information.

Chapter 25

Temporal Probability Models

25.1 Modeling Time and Uncertainty

Stochastic Processes

The world changes in *stochastically predictable ways.*

Example 25.1.1.

- ▷ The weather changes, but the weather tomorrow is somewhat predictable *given* today's weather and other factors, (which in turn (somewhat) depends on yesterday's weather, which in turn...)
- ▷ the stock market changes, but the stock price tomorrow is probably related to today's price,
- ▷ A patient's blood sugar changes, but their blood sugar is related to their blood sugar 10 minutes ago (in particular if they didn't eat anything in between)

How do we model this?

Definition 25.1.2. Let $\langle\Omega, P\rangle$ a probability space and $\langle S, \preceq\rangle$ a (not necessarily *totally*) ordered set.

A sequence of random variables $(X_t)_{t\in S}$ with $\mathbf{dom}(X_t) = D$ is called a stochastic process over the time structure S.

Intuition: X_t models the outcome of the random variable X at time step t. The sample space Ω corresponds to the set of all possible sequences of outcomes.

Note: We will almost exclusively use $\langle S, \preceq\rangle = \langle\mathbb{N}, \leq\rangle$.

Definition 25.1.3. Given a stochastic process X_t over S and $a, b \in S$ with $a \preceq b$, we write $\mathbf{X}_{a:b}$ for the sequence $X_a, X_{a+1}, \ldots, X_{b-1}, X_b$ and $E_{a:b}^{=e}$ for $E_a = e_a, \ldots, E_b = e_b$.

FAU Michael Kohlhase: Artificial Intelligence 2 842 2025-05-01

Stochastic Processes (Running Example)

Example 25.1.4 (Umbrellas). You are a security guard in a secret underground facility, want to know it if is raining outside. Your only source of information is whether the director comes in with an umbrella.

- ▷ We have a stochastic process $\mathtt{Rain}_0, \mathtt{Rain}_1, \mathtt{Rain}_2, \ldots$ of hidden variables, and
- ▷ a related stochastic process $\mathtt{Umbrella}_0, \mathtt{Umbrella}_1, \mathtt{Umbrella}_2, \ldots$ of evidence variables.

...and a combined stochastic process $\langle \texttt{Rain}_0, \texttt{Umbrella}_0 \rangle, \langle \texttt{Rain}_1, \texttt{Umbrella}_1 \rangle, \ldots$

Note that $\texttt{Umbrella}_t$ only depends on $\texttt{Rain}_t$, not on e.g. $\texttt{Umbrella}_{t-1}$ (except indirectly through $\texttt{Rain}_t$ / $\texttt{Rain}_{t-1}$).

Definition 25.1.5. We call a stochastic process of *hidden* variables a state variable.

Markov Processes

Idea: Construct a Bayesian network from these variables (parents?)

...without everything exploding in size...?

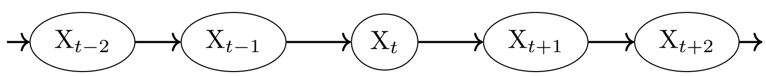
Definition 25.1.6. Let $(X_t)_{t \in S}$ a stochastic process. X has the (nth order) Markov property iff X_t only depends on a bounded subset of $\mathbf{X}_{0:t-1}$ – i.e. for all $t \in S$ we have $\mathbb{P}(X_t | X_0, \ldots X_{t-1}) = \mathbb{P}(X_t | X_{t-n}, \ldots X_{t-1})$ for some $n \in S$.

A stochastic process with the Markov property for some n is called a (nth order) Markov process.

Important special cases:

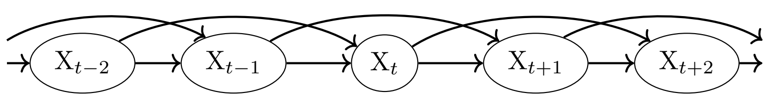
Definition 25.1.7.

- First-order Markov property: $\mathbb{P}(\mathbf{X}_t | \mathbf{X}_{0:t-1}) = \mathbb{P}(\mathbf{X}_t | \mathbf{X}_{t-1})$



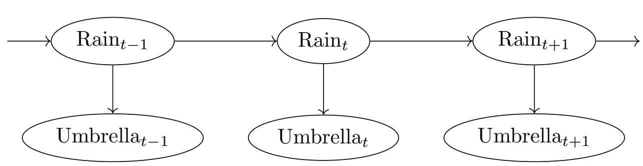
A first order Markov process is called a Markov chain.

- Second-order Markov property: $\mathbb{P}(\mathbf{X}_t | \mathbf{X}_{0:t-1}) = \mathbb{P}(\mathbf{X}_t | \mathbf{X}_{t-2}, \mathbf{X}_{t-1})$



Markov Process Example: The Umbrella

Example 25.1.8 (Umbrellas continued). We model the situation in a Bayesian network:



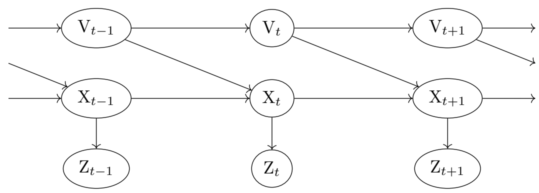
Problem: This network does not actually have the First-order Markov property...

Possible fixes: We have two ways to fix this:

1. Increase the order of the Markov process. (more dependencies $\Rightarrow$ more complex inference)
2. Add more state variables, e.g., Temp_t, $\mathrm{Pressure}_t$. (more information sources)

Markov Process Example: Robot Motion

Example 25.1.9 (Random Robot Motion). Assume we want to track a robot wandering randomly on the X/Y plane, whose position we can only observe roughly (e.g. by approximate GPS coordinates:) Markov chain



- ▷ the velocity V_i may change unpredictably.
- ▷ the exact position X_i depends on previous position X_{i-1} and velocity V_{i-1}
- ▷ the position X_i influences the observed position Z_i.

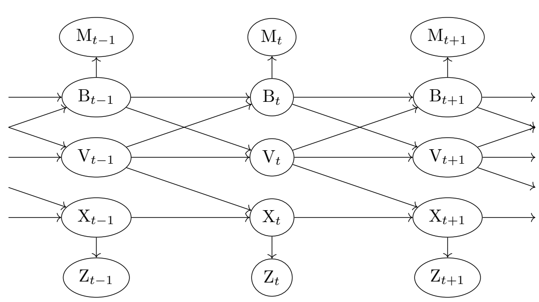
Example 25.1.10 (Battery Powered Robot). If the robot has a *battery*, the Markov property is violated!

- ▷ Battery exhaustion has a systematic effect on the change in velocity.
- ▷ This depends on how much power was used by all previous manoeuvres.

Markov Process Example: Robot Motion

Idea: We can restore the Markov property by including a state variable for the charge level B_t. (Better still: Battery level sensor)

Example 25.1.11 (Battery Powered Robot Motion).



- ▷ Battery level B_i is influenced by previous level $B_{i-1}$and velocity V_{i-1}.
- ▷ Velocity V_i is influenced by previous level $B_{i-1}$and velocity V_{i-1}.
- ▷ Battery meter M_i is only influenced by Battery level B_i.

Stationary Markov Processes as Transition Models

Remark 25.1.12. Given a stochastic process with state variables X_t and evidence variables E_t, then $\mathbb{P}(X_t|\mathbf{X}_{0:t})$ is a transition model and $\mathbb{P}(E_t|\mathbf{X}_{0:t}, \mathbf{E}_{1:t-1})$ a sensor model in the sense of a model-based agent.

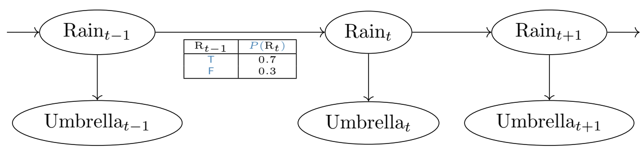
Note that we assume that the X_t do not depend on the E_t.

Also note that with the Markov property, the transition model simplifies to $\mathbb{P}(\mathbf{X}_t|\mathbf{X}_{t-n})$.

Problem: Even with the Markov property the transition model is infinite. $(t \in \mathbb{N})$

Definition 25.1.13. A Markov chain is called stationary if the transition model is independent of time, i.e. $\mathbb{P}(X_t|X_{t-1})$ is the same for all t.

Example 25.1.14 (Umbrellas are stationary). $\mathbb{P}(\mathtt{Rain}_t|\mathtt{Rain}_{t-1})$ does not depend on t.(need only one table)



⚠ Don't confuse "stationary" (Markov processes) with "static" (environments).

We restrict ourselves to stationary Markov processes in AI-2.

Markov Sensor Models

Recap: The sensor model $\mathbb{P}(E_t|\mathbf{X}_{0:t}, \mathbf{E}_{1:t-1})$ allows us (using Bayes rule et al) to update our belief state about X_t given the observations $\mathbf{E}_{0:t}$.

Problem: The evidence variables E_t could depend on any of the variables $\mathbf{X}_{0:t}, \mathbf{E}_{1:t-1}$...

Definition 25.1.15. We say that a sensor model has the sensor Markov property, iff $\mathbb{P}(E_t|\mathbf{X}_{0:t}, \mathbf{E}_{1:t-1}) = \mathbb{P}(E_t|X_t)$ – i.e., the sensor model depends only on the current state.

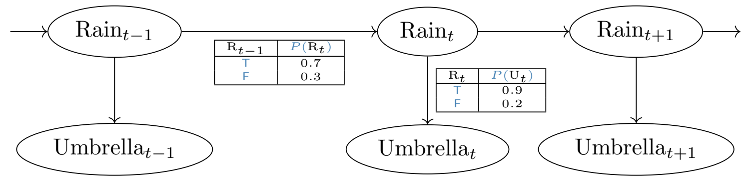
Assumptions on Sensor Models: We usually assume the sensor Markov property and make it stationary as well: $\mathbb{P}(E_t|X_t)$ is fixed for all t.

Definition 25.1.16 (Note).

- ▷ If a Markov chain X is stationary and discrete, we can represent the transition model as a matrix $\mathbf{T}_{ij} := P(X_t = j|X_{t-1} = i)$.
- ▷ If a sensor model has the sensor Markov property, we can represent each observation $E_t = e_t$ at time t as the diagonal matrix $\mathbf{O}_t$ with $\mathbf{O}_{tii} := P(E_t = e_t|X_t = i)$.
- ▷ A pair $\langle X, E\rangle$ where X is a (stationary) Markov chains, E_i only depends on X_i, and E has the sensor Markov property is called a (stationary) Hidden Markov Model (HMM). (X and E are single variables)

Umbrellas, the full Story

Example 25.1.17 (Umbrellas, Transition & Sensor Models).



This is a hidden Markov model

Observation 25.1.18. *If we know the initial prior probabilities $\mathbb{P}(X_0)$ ($\widehat{=}$ time $t = 0$), then we can compute the full joint probability distribution as*

$$\mathbb{P}(\mathbf{X}_{0:t}, \mathbf{E}_{1:t}) = \mathbb{P}(X_0) \cdot (\prod_{i=1}^{t} \mathbb{P}(X_i|X_{i-1}) \cdot \mathbb{P}(E_i|X_i))$$

25.2 Inference: Filtering, Prediction, and Smoothing

Inference tasks

Definition 25.2.1. Given a Markov process with state variables X_t and evidence variables E_t, we are interested in the following Markov inference tasks:

- ▷ Filtering (or monitoring) $\mathbb{P}(X_t|E_{1:t}^{=e})$: Given the sequence of observations up until time t, compute the likely state of the world at *current* time t.
- ▷ Prediction (or state estimation) $\mathbb{P}(X_{t+k}|E_{1:t}^{=e})$ for $k > 0$: Given the sequence of observations up until time t, compute the likely *future* state of the world at time $t + k$.
- ▷ Smoothing (or hindsight) $\mathbb{P}(X_{t-k}|E_{1:t}^{=e})$ for $0 < k < t$: Given the sequence of observations up until time t, compute the likely *past* state of the world at time $t - k$.
- ▷ Most likely explanation $\underset{\mathbf{x}_{1:t}}{\operatorname{argmax}} (P(X_{1:t}^{=x}|E_{1:t}^{=e}))$: Given the sequence of observations up until time t, compute the most likely sequence of states that led to these observations.

  Note: The most likely sequence of states is *not* (necessarily) the sequence of most likely states ;-)

In this section, we assume X and E to represent *multiple* variables, where X jointly forms a Markov chain and the E jointly have the sensor Markov property.

In the case where X and E are stationary *single* variables, we have a stationary hidden Markov model and can use the matrix forms.

Filtering (Computing the Belief State given Evidence)

Note:

- ▷ Using the full joint probability distribution, we can compute any conditional probability we want, but not necessarily efficiently.
- ▷ We want to use filtering to update our "world model" $\mathbb{P}(X_t)$ based on a new observation $E_t = e_t$ and our *previous* world model $\mathbb{P}(X_{t-1})$.

$\Rightarrow$ We want a function $\mathbb{P}(X_t|E_{1:t}^{=e}) = F(e_t, \underbrace{\mathbb{P}(X_{t-1}|E_{1:t-1}^{=e})}_{F(e_{t-1},\ldots)})$

Spoiler:

$$F(e_t, \mathbb{P}(X_{t-1}|E_{1:t-1}^{=e})) = \alpha(\mathbf{O}_t \cdot \mathbf{T}^T \cdot \mathbb{P}(X_{t-1}|E_{1:t-1}^{=e}))$$

Filtering Derivation

$$\begin{aligned}
\mathbb{P}(X_t|E_{1:t}^{=e}) &= \mathbb{P}(X_t|E_t = e_t, E_{1:t-1}^{=e}) && \text{(dividing up evidence)}\\
&= \alpha(\mathbb{P}(E_t = e_t|X_t, E_{1:t-1}^{=e}) \cdot \mathbb{P}(X_t|E_{1:t-1}^{=e})) && \text{(using Bayes' rule)}\\
&= \alpha(\mathbb{P}(E_t = e_t|X_t) \cdot \mathbb{P}(X_t|E_{1:t-1}^{=e})) && \text{(sensor Markov property)}\\
&= \alpha(\mathbb{P}(E_t = e_t|X_t) \cdot (\sum_{x\in\mathbf{dom}(X)} \mathbb{P}(X_t|X_{t-1} = x, E_{1:t-1}^{=e}) \cdot P(X_{t-1} = x|E_{1:t-1}^{=e}))) && \text{(marginalization)}\\
&= \alpha(\underbrace{\mathbb{P}(E_t = e_t|X_t)}_{\text{sensor model}} \cdot (\sum_{x\in\mathbf{dom}(X)} \underbrace{\mathbb{P}(X_t|X_{t-1} = x)}_{\text{transition model}} \cdot \underbrace{P(X_{t-1} = x|E_{1:t-1}^{=e})}_{\text{recursive call}})) && \text{(conditional independence)}
\end{aligned}$$

Reminder: In a stationary HMM, we have the matrices $\mathbf{T}_{ij} = P(X_t = j|X_{t-1} = i)$ and $\mathbf{O}_{tii} = P(E_t = e_t|X_t = i)$.

Then interpreting $\mathbb{P}(X_{t-1}|E_{1:t-1}^{=e})$ as a vector, the above corresponds exactly to the matrix multiplication $\alpha(\mathbf{O}_t \cdot \mathbf{T}^T \cdot \mathbb{P}(X_{t-1}|E_{1:t-1}^{=e}))$

Definition 25.2.2. We call the inner part of the above expression the forward algorithm, i.e. $\mathbb{P}(X_t|E_{1:t}^{=e}) = \alpha(\mathrm{FORWARD}(e_t, \mathbb{P}(X_{t-1}|E_{1:t-1}^{=e}))) =: \mathbf{f}_{1:t}$.

Filtering the Umbrellas

Example 25.2.3. Let's assume:

- $\triangleright$ $\mathbb{P}(\mathtt{R}_0) = \langle 0.5, 0.5\rangle$, (Note that with growing t (and evidence), the impact of the prior at $t = 0$ vanishes anyway)
- $\triangleright$ $P(\mathtt{R}_{t+1}|\mathtt{R}_t) = 0.6$, $P(\neg\mathtt{R}_{t+1}|\neg\mathtt{R}_t) = 0.8$, $P(\mathtt{U}_t|\mathtt{R}_t) = 0.9$ and $P(\neg\mathtt{U}_t|\neg\mathtt{R}_t) = 0.85$

$\Rightarrow$ $\mathbf{T} = \begin{pmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{pmatrix}$

- $\triangleright$ The director carries an umbrella on days 1 and 2, and *not* on day 3.

$\Rightarrow$ $\mathbf{O}_1 = \mathbf{O}_2 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.15 \end{pmatrix}$ and $\mathbf{O}_3 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.85 \end{pmatrix}$.

Then:

- $\triangleright$ $\mathbf{f}_{1:1} := \mathbb{P}(\mathtt{R}_1|\mathtt{U}_1 = \mathsf{T}) = \alpha(\mathbb{P}(\mathtt{U}_1 = \mathsf{T}|\mathtt{R}_1) \cdot (\sum_{b\in\{\mathsf{T},\mathsf{F}\}} \mathbb{P}(\mathtt{R}_1|\mathtt{R}_0 = b) \cdot P(\mathtt{R}_0 = b)))$

  $=\alpha(\langle 0.9, 0.15\rangle \cdot (\langle 0.6, 0.4\rangle \cdot 0.5 + \langle 0.2, 0.8\rangle \cdot 0.5)) = \alpha(\langle 0.36, 0.09\rangle) = \langle 0.8, 0.2\rangle$

▷ Using matrices: $\alpha(\mathbf{O}_1 \cdot \mathbf{T}^T \cdot \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}) = \alpha(\begin{pmatrix} 0.9 & 0 \\ 0 & 0.15 \end{pmatrix} \cdot \begin{pmatrix} 0.6 & 0.2 \\ 0.4 & 0.8 \end{pmatrix} \cdot \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix})$
$= \alpha(\begin{pmatrix} 0.9 \cdot 0.6 & 0.9 \cdot 0.2 \\ 0.15 \cdot 0.4 & 0.15 \cdot 0.8 \end{pmatrix} \cdot \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}) = \alpha(\begin{pmatrix} 0.9 \cdot 0.6 \cdot 0.5 + 0.9 \cdot 0.2 \cdot 0.5 \\ 0.15 \cdot 0.4 \cdot 0.5 + 0.15 \cdot 0.8 \cdot 0.5 \end{pmatrix}) = \alpha(\begin{pmatrix} 0.36 \\ 0.09 \end{pmatrix})$

Filtering the Umbrellas (Continued)

Example 25.2.4. $\mathbf{f}_{1:1} := \mathbb{P}(\mathtt{R}_1|\mathtt{U}_1 = \mathsf{T}) = \langle 0.8, 0.2\rangle$

▷ $\mathbf{f}_{1:2} := \mathbb{P}(\mathtt{R}_2|\mathtt{U}_2 = \mathsf{T}, \mathtt{U}_1 = \mathsf{T}) = \alpha(\mathbf{O}_2 \cdot \mathbf{T}^T \cdot \mathbf{f}_{1:1}) = \alpha(\mathbb{P}(\mathtt{U}_2 = \mathsf{T}|\mathtt{R}_2) \cdot (\sum_{b \in \{\mathsf{T},\mathsf{F}\}} \mathbb{P}(\mathtt{R}_2|\mathtt{R}_1 = b) \cdot \mathbf{f}_{1:1}(b)))$
$= \alpha(\langle 0.9, 0.15\rangle \cdot (\langle 0.6, 0.4\rangle \cdot 0.8 + \langle 0.2, 0.8\rangle \cdot 0.2)) = \alpha(\langle 0.468, 0.072\rangle) = \langle 0.87, 0.13\rangle$

▷ $\mathbf{f}_{1:3} := \mathbb{P}(\mathtt{R}_3|\mathtt{U}_3 = \mathsf{F}, \mathtt{U}_2 = \mathsf{T}, \mathtt{U}_1 = \mathsf{T}) = \alpha(\mathbf{O}_3 \cdot \mathbf{T}^T \cdot \mathbf{f}_{1:2})$
$= \alpha(\mathbb{P}(\mathtt{U}_3 = \mathsf{F}|\mathtt{R}_3) \cdot (\sum_{b \in \{\mathsf{T},\mathsf{F}\}} \mathbb{P}(\mathtt{R}_3|\mathtt{R}_2 = b) \cdot \mathbf{f}_{1:2}(b)))$
$= \alpha(\langle 0.1, 0.85\rangle \cdot (\langle 0.6, 0.4\rangle \cdot 0.87 + \langle 0.2, 0.8\rangle \cdot 0.13)) = \alpha(\langle 0.0547, 0.3853\rangle) = \langle 0.12, 0.88\rangle$

Prediction in Markov Chains

Prediction: $\mathbb{P}(X_{t+k}|E_{1:t}^{=e})$ for $k > 0$.
Intuition: Prediction is filtering without new evidence – i.e. we can use filtering until t, and then continue as follows:
Lemma 25.2.5. *By the same reasoning as filtering:*

$$\mathbb{P}(X_{t+k+1}|E_{1:t}^{=e}) = \sum_{x \in \mathbf{dom}(X)} \underbrace{\mathbb{P}(X_{t+k+1}|X_{t+k} = x)}_{\text{transition model}} \cdot \underbrace{P(X_{t+k} = x|E_{1:t}^{=e})}_{\text{recursive call}} \underbrace{= \mathbf{T}^T \cdot \mathbb{P}(X_{t+k} = x|E_{1:t}^{=e})}_{\text{HMM}}$$

Observation 25.2.6. *As $k \to \infty$, $\mathbb{P}(X_{t+k}|E_{1:t}^{=e})$ converges towards a fixed point called the stationary distribution of the Markov chain. (which we can compute from the equation $S = \mathbf{T}^T \cdot S$)*
⇒ the impact of the evidence vanishes.
⇒ The stationary distribution only depends on the transition model.
⇒ There is a small window of time (depending on the transition model) where the evidence has enough impact to allow for prediction beyond *the mere stationary distribution, called the mixing time of the Markov chain.*
⇒ Predicting the future is difficult, and the further into the future, the more difficult it is (Who knew...)

Smoothing

Smoothing: $\mathbb{P}(X_{t-k}|E_{1:t}^{=e})$ for $k > 0$.
Intuition: Use filtering to compute $\mathbb{P}(X_t|E_{1:t-k}^{=e})$, then recurse *backwards* from t until $t-k$.

$$\begin{aligned}
\mathbb{P}(X_{t-k}|E_{1:t}^{=e}) &= \mathbb{P}(X_{t-k}|E_{t-(k-1):t}^{=e}, E_{1:t-k}^{=e}) && \text{(Divide the evidence)}\\
&= \alpha(\mathbb{P}(E_{t-(k-1):t}^{=e}|X_{t-k}, E_{1:t-k}^{=e}) \cdot \mathbb{P}(X_{t-k}|E_{1:t-k}^{=e})) && \text{(Bayes Rule)}\\
&= \alpha(\underbrace{\mathbb{P}(E_{t-(k-1):t}^{=e}|X_{t-k})}_{=:\mathbf{b}_{t-(k-1):t}} \cdot \underbrace{\mathbb{P}(X_{t-k}|E_{1:t-k}^{=e})}_{=\mathbf{f}_{1:t-k}}) && \text{(cond. independence)}\\
&= \alpha(\mathbf{f}_{1:t-k} \times \mathbf{b}_{t-(k-1):t})
\end{aligned}$$

(where $\times$ denotes component-wise multiplication)

Smoothing (continued)

Definition 25.2.7 (Backward message). $\mathbf{b}_{t-k:t} = \mathbb{P}(E_{t-k:t}^{=e}|X_{t-(k+1)})$

$$\begin{aligned}
&= \sum_{x\in\mathbf{dom}(X)} \mathbb{P}(E_{t-k:t}^{=e}|X_{t-k}=x, X_{t-(k+1)}) \cdot \mathbb{P}(X_{t-k}=x|X_{t-(k+1)})\\
&= \sum_{x\in\mathbf{dom}(X)} P(E_{t-k:t}^{=e}|X_{t-k}=x) \cdot \mathbb{P}(X_{t-k}=x|X_{t-(k+1)})\\
&= \sum_{x\in\mathbf{dom}(X)} P(E_{t-k}=e_{t-k}, E_{t-(k-1):t}^{=e}|X_{t-k}=x) \cdot \mathbb{P}(X_{t-k}=x|X_{t-(k+1)})\\
&= \sum_{x\in\mathbf{dom}(X)} \underbrace{P(E_{t-k}=e_{t-k}|X_{t-k}=x)}_{\text{sensor model}} \cdot \underbrace{P(E_{t-(k-1):t}^{=e}|X_{t-k}=x)}_{=\mathbf{b}_{t-(k-1):t}} \cdot \underbrace{\mathbb{P}(X_{t-k}=x|X_{t-(k+1)})}_{\text{transition model}}
\end{aligned}$$

Note: in a stationary hidden Markov model, we get the matrix formulation $\mathbf{b}_{t-k:t} = \mathbf{T}\cdot\mathbf{O}_{t-k}\cdot\mathbf{b}_{t-(k-1):t}$

Definition 25.2.8. We call the associated algorithm the backward algorithm, i.e. $\mathbb{P}(X_{t-k}|E_{1:t}^{=e}) = \alpha(\underbrace{\mathrm{FORWARD}(e_{t-k}, \mathbf{f}_{1:t-(k+1)})}_{\mathbf{f}_{1:t-k}} \times \underbrace{\mathrm{BACKWARD}(e_{t-(k-1)}, \mathbf{b}_{t-(k-2):t})}_{\mathbf{b}_{t-(k-1):t}})$.

As a starting point for the recursion, we let $\mathbf{b}_{t+1:t}$ the uniform vector with 1 in every component.

Smoothing example

Example 25.2.9 (Smoothing Umbrellas). Reminder: We assumed $\mathbb{P}(R_0) = \langle 0.5, 0.5\rangle$, $P(R_{t+1}|R_t) = 0.6$, $P(\neg R_{t+1}|\neg R_t) = 0.8$, $P(U_t|R_t) = 0.9$, $P(\neg U_t|\neg R_t) = 0.85$

$\Rightarrow \mathbf{T} = \begin{pmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{pmatrix}$, $\mathbf{O}_1 = \mathbf{O}_2 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.15 \end{pmatrix}$ and $\mathbf{O}_3 = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.85 \end{pmatrix}$. (The director carries an umbrella on days 1 and 2, and *not* on day 3)

$\mathbf{f}_{1:1} = \langle 0.8, 0.2\rangle$, $\mathbf{f}_{1:2} = \langle 0.87, 0.13\rangle$ and $\mathbf{f}_{1:3} = \langle 0.12, 0.88\rangle$

Let's compute

$$\mathbb{P}(R_1|U_1 = T, U_2 = T, U_3 = F) = \alpha(\mathbf{f}_{1:1} \times \mathbf{b}_{2:3})$$

▷ We need to compute $\mathbf{b}_{2:3}$ and $\mathbf{b}_{3:3}$:

▷ $\mathbf{b}_{3:3} = \mathbf{T}\cdot\mathbf{O}_3\cdot\mathbf{b}_{4:3} = \begin{pmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{pmatrix} \cdot \begin{pmatrix} 0.1 & 0 \\ 0 & 0.85 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.4 \\ 0.7 \end{pmatrix}$

▷ $\mathbf{b}_{2:3} = \mathbf{T} \cdot \mathbf{O}_2 \cdot \mathbf{b}_{3:3} = \begin{pmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{pmatrix} \cdot \begin{pmatrix} 0.9 & 0 \\ 0 & 0.15 \end{pmatrix} \cdot \begin{pmatrix} 0.4 \\ 0.7 \end{pmatrix} = \begin{pmatrix} 0.258 \\ 0.156 \end{pmatrix}$

$\Rightarrow \alpha(\begin{pmatrix} 0.8 \\ 0.2 \end{pmatrix} \times \begin{pmatrix} 0.258 \\ 0.156 \end{pmatrix}) = \alpha(\begin{pmatrix} 0.2064 \\ 0.0312 \end{pmatrix}) = \begin{pmatrix} 0.87 \\ 0.13 \end{pmatrix}$

⇒ Given the evidence $\mathtt{U}_2, \neg\mathtt{U}_3$, the posterior probability for $\mathtt{R}_1$ went up from 0.8 to 0.87!

Forward/Backward Algorithm for Smoothing

Definition 25.2.10. Forward backward algorithm: returns the sequence of posterior distributions $\mathbb{P}(X_1)\ldots\mathbb{P}(X_t)$ given evidence $e_1,\ldots,e_t$:

```
function FORWARD-BACKWARD(⟨e_1,...,e_t⟩, ℙ(X_0))
    f := ⟨ℙ(X_0)⟩
    b := ⟨1, 1, ...⟩
    S := ⟨ℙ(X_0)⟩
    for i = 1,...,t do
        f_i := FORWARD(f_{i−1}, e_i)                /* filtering */
    for i = t,...,1 do
        S_i := α(f_i × b)                           /* smoothing */
        b := BACKWARD(b, e_i)
    return S
```

Time complexity linear in t (polytree inference), Space complexity $\mathcal{O}(t \cdot |\mathbf{f}|)$.

Country dance algorithm

Idea: If $\mathbf{T}$ and $\mathbf{O}_i$ are invertible, we can avoid storing all forward messages in the smoothing algorithm by running filtering backwards:

$$\mathbf{f}_{1:i+1} = \alpha(\mathbf{O}_{i+1} \cdot \mathbf{T}^T \cdot \mathbf{f}_{1:i})$$

$$\Rightarrow \mathbf{f}_{1:i} = \alpha({\mathbf{T}^T}^{-1} \cdot {\mathbf{O}_{i+1}}^{-1} \cdot \mathbf{f}_{1:i+1})$$

⇒ we can trade space complexity for time complexity:

▷ In the first for-loop, we only compute the final $\mathbf{f}_{1:t}$ (No need to store the intermediate results)

▷ In the second for-loop, we compute both $\mathbf{f}_{1:i}$ and $\mathbf{b}_{t-i:t}$ (Only one copy of $\mathbf{f}_{1:i}$, $\mathbf{b}_{t-i:t}$ is stored)

⇒ constant space.

But: Requires that both matrices are invertible, i.e. *every observation must be possible in every state.* (Possible hack: increase the probabilities of 0 to "negligibly small")

Most Likely Explanation

Smoothing allows us to compute the *sequence of most likely states* $X_1,\ldots,X_t$ given $E_{1:t}^{=e}$. What if we want the *most likely sequence* of states? i.e. $\max_{x_1,\ldots,x_t}(P(X_{1:t}^{=x}|E_{1:t}^{=e}))$?

Example 25.2.11. Given the sequence $\mathtt{U}_1,\mathtt{U}_2,\neg\mathtt{U}_3,\mathtt{U}_4,\mathtt{U}_5$, the most likely state for $\mathtt{R}_3$ is F, but the most likely sequence *might* be that it rained throughout...

Prominent Application: In speech recognition, we want to find the most likely word sequence, given what we have heard. (can be quite noisy)

Idea:

- ▷ For every $x_t \in \mathbf{dom}(X)$ and $0 \leq i \leq t$, recursively compute the most likely path $X_1,\ldots,X_i$ ending in $X_i = x_i$ given the observed evidence.
- ▷ remember the x_{i-1} that most likely leads to x_i.
- ▷ Among the resulting paths, pick the one to *the* $X_t = x_t$ with the most likely path,
- ▷ and then recurse backwards.

⇒ we want to know $\max_{x_1,\ldots,x_{t-1}} \mathbb{P}(X_{1:t-1}^{=x}, X_t|E_{1:t}^{=e})$, and then pick the x_t with the maximal value.

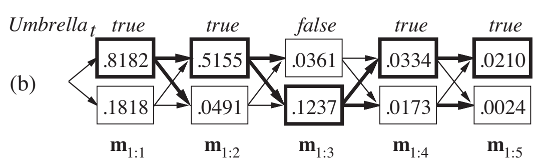
Most Likely Explanation (continued)

By the same reasoning as for filtering:

$$\max_{x_1,\ldots,x_{t-1}} \mathbb{P}(X_{1:t-1}^{=x}, X_t|E_{1:t}^{=e})$$
$$= \alpha(\underbrace{\mathbb{P}(E_t = e_t|X_t)}_{\text{sensor model}} \cdot \max_{x_{t-1}} (\underbrace{\mathbb{P}(X_t|X_{t-1} = x_{t-1})}_{\text{transition model}} \cdot \underbrace{\max_{x_1,\ldots,x_{t-2}} (P(X_{1:t-2}^{=x}, X_{t-1} = x_{t-1}|E_{1:t-1}^{=e}))}_{=:\mathbf{m}_{1:t-1}(x_{t-1})}))$$

$\mathbf{m}_{1:t}(i)$ gives the maximal probability that the most likely path up to t leads to state $X_t = i$. Note that we can leave out the α, since we're only interested in the maximum.

Example 25.2.12. For the sequence [T, T, F, T, T]:



bold arrows: best predecessor measured by "best preceding sequence probability × transition probability"

The Viterbi Algorithm

Definition 25.2.13. The Viterbi algorithm now proceeds as follows:

function $\textsc{Viterbi}(\langle e_1,\ldots,e_t\rangle, \mathbb{P}(X_0))$
  $m := \mathbb{P}(X_0)$ /* $\mathbf{m}_{1:i}$ */
  $\text{prev} := \langle\rangle$ /* *the most likely predecessor of each possible* x_i */
  for $i = 1,\ldots,t$ **do**
    $m' := \max_{x_{i-1}} (\mathbb{P}(E_i = e_i|X_i) \cdot \mathbb{P}(X_i|X_{i-1} = x_{i-1}) \cdot m_{x_{i-1}})$
    $\text{prev}_{i-1} := \operatorname{argmax}_{x_{i-1}} (\mathbb{P}(E_i = e_i|X_i) \cdot \mathbb{P}(X_i|X_{i-1} = x_{i-1}) \cdot m_{x_{i-1}})$
    $m \longleftarrow m'$
  $P := \langle 0, 0, \ldots, \operatorname{argmax}_{(x\in\mathbf{dom}(X))} m_x\rangle$
  for $i = t-1,\ldots,0$ **do**
    $P_i := \text{prev}_{i,P_{i+1}}$
  return P

Observation 25.2.14. *Viterbi has linear time complexity and linear space complexity (needs to keep the most likely sequence leading to each state).*

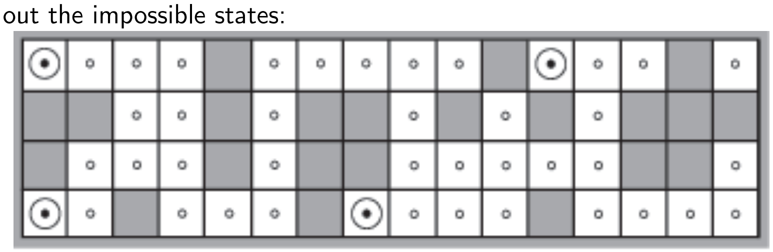
25.3 Hidden Markov Models – Extended Example

Example: Robot Localization using Common Sense

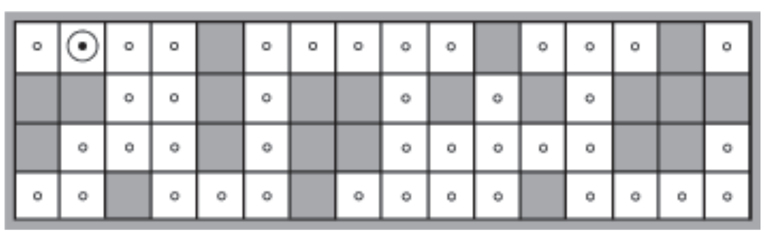
Example 25.3.1 (Robot Localization in a Maze). A robot has four sonar sensors that tell it about obstacles in four directions: N, S, W, E.

We write the result where the sensor that detects obstacles in the north, south, and east as $\mathrm{N\ S\ E}$.

We filter out the impossible states:



a) Possible robot locations after $\mathrm{e}_1 = \mathrm{N\ S\ W}$



b) Possible robot locations after $\mathrm{e}_1 = \mathrm{N\ S\ W}$ and $\mathrm{e}_2 = \mathrm{N\ S}$

Remark 25.3.2. This only works for perfect sensors. (else no impossible states)

What if our sensors are imperfect?

HMM Example: Robot Localization (Modeling)

Example 25.3.3 (HMM-based Robot Localization). We have the following setup:

▷ A hidden Random variable X_t for robot location (domain: 42 empty squares)

▷ Let $N(i)$ be the set of neighboring fields of the field $X_i = x_i$

▷ The Transition matrix for the move action ($\mathbf{T}$ has $42^2 = 1764$ entries)

$$P(X_{t+1} = j | X_t = i) = \mathbf{T}_{ij} = \begin{cases} \frac{1}{|N(i)|} & \text{if } j \in N(i) \\ 0 & \text{else} \end{cases}$$

▷ We do not know where the robot starts: $P(X_0) = \frac{1}{n}$ (here $n = 42$)

▷ Evidence variable E_t: four bit presence/absence of obstacles in N, S, W, E. Let d_{it} be the number of wrong bits and ϵ the error rate of the sensor. Then

$$P(E_t = e_t | X_t = i) = \mathbf{O}_{tii} = (1 - \epsilon)^{4 - d_{it}} \cdot \epsilon^{d_{it}}$$

(We assume the sensors are independent)

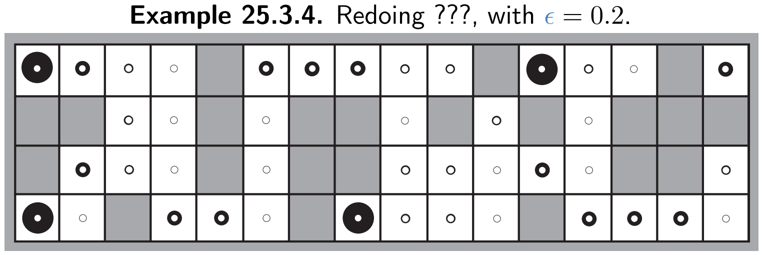
For example, the probability that the sensor on a square with obstacles in north and south would produce N S E is $(1 - \epsilon)^3 \cdot \epsilon^1$.

We can now use filtering for localization, smoothing to determine e.g. the starting location, and the Viterbi algorithm to find out how the robot got to where it is now.

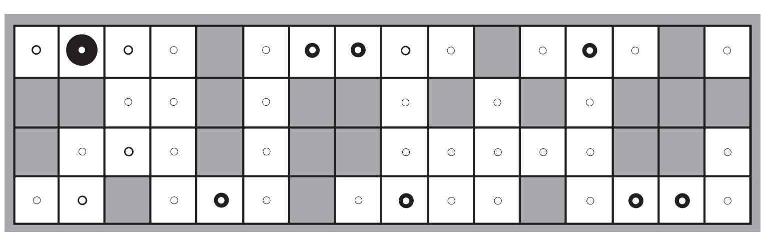
HMM Example: Robot Localization

We use HMM filtering equation $\mathbf{f}_{1:t+1} = \alpha \cdot \mathbf{O}_{t+1} \mathbf{T}^t \mathbf{f}_{1:t}$ to compute posterior distribution over locations. (i.e. robot localization)

Example 25.3.4. Redoing ???, with $\epsilon = 0.2$.



a) Posterior distribution over robot location after $\mathrm{E}_1 = \mathrm{N\ S\ W}$



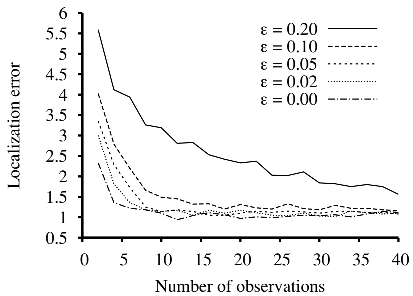
b) Posterior distribution over robot location after $\mathrm{E}_1 = \mathrm{N\ S\ W}$ and $\mathrm{E}_2 = \mathrm{N\ S}$

Still the same locations as in the "perfect sensing" case, but now other locations have non-zero probability.

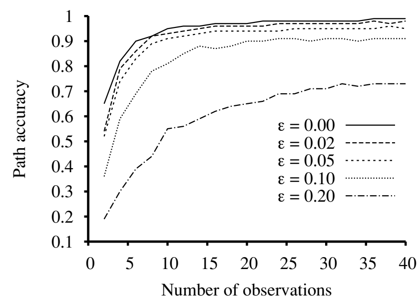
HMM Example: Further Inference Applications

Idea: We can use smoothing: $\mathbf{b}_{k+1:t} = \mathbf{TO}_{k+1}\mathbf{b}_{k+2:t}$ to find out where it started and the Viterbi algorithm to find the most likely path it took.

Example 25.3.5. Performance of HMM localization vs. observation length (various error rates ϵ)



Localization error (Manhattan distance from true location)



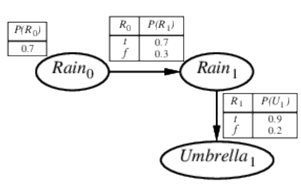
Viterbi path accuracy (fraction of correct states on Viterbi path)

25.4 Dynamic Bayesian Networks

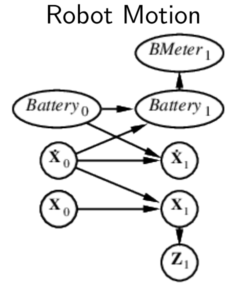
Dynamic Bayesian networks

- **Definition 25.4.1.** A Bayesian network $\mathcal{D}$ is called dynamic (a DBN), iff its random variables are indexed by a time structure. We assume that $\mathcal{D}$ is
  - time sliced, i.e. that the time slices $\mathcal{D}_t$ – the subgraphs of t-indexed random variables and the edges between them – are isomorphic.
  - a stationary Markov chain, i.e. that variables X_t can only have parents in $\mathcal{D}_t$ and $\mathcal{D}_{t-1}$.
- $\mathbf{X}_t$, $\mathbf{E}_t$ contain arbitrarily many variables in a replicated Bayesian network.
- **Example 25.4.2.**

Umbrellas



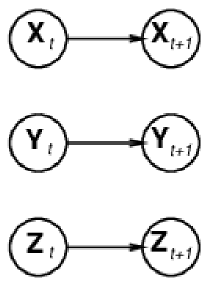
Robot Motion

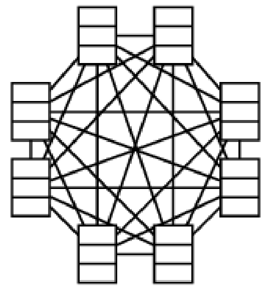


DBNs vs. HMMs

▷ **Observation 25.4.3.**

- ▹ *Every HMM is a single-variable DBN.* *(trivially)*
- ▹ *Every DBN can be turned into an HMM.* *(combine variables into tuple ⇒ lose information about dependencies)*
- ▹ *DBNs have sparse dependencies ⇝ exponentially fewer parameters;*

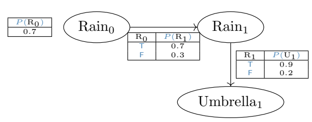


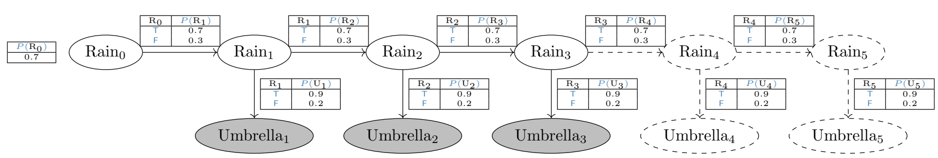


▷ **Example 25.4.4 (Sparse Dependencies).** With 20 Boolean state variables, three parents each, a DBN has $20 \cdot 2^3 = 160$ parameters, the corresponding HMM has $2^{20} \cdot 2^{20} \approx 10^{12}$.

Exact inference in DBNs

▷ **Definition 25.4.5 (Naive method).** Unroll the network and run any exact algorithm.





▷ **Problem:** Inference cost for each update grows with t.

▷ **Definition 25.4.6.** Rollup filtering: add slice $t+1$, "sum out" slice t using variable elimination.

▷ **Observation:** Largest factor is $\mathcal{O}(d^{n+1})$, update cost $\mathcal{O}(d^{n+2})$, where d is the maximal domain size.

▷ **Note:** Much better than the HMM update cost of $\mathcal{O}(d^{2n})$

Summary

▷ Temporal probability models use state and evidence variables replicated over time.

- ▷ Markov property and stationarity assumption, so we need both
  - ▷ a transition model and $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$
  - ▷ a sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$.
- ▷ Tasks are filtering, prediction, smoothing, most likely sequence; (all done recursively with constant cost per time step)
- ▷ Hidden Markov models have a single discrete state variable; (used for speech recognition)
- ▷ DBNs subsume HMMs, exact update intractable.

Chapter 26

Making Complex Decisions

We will now pick up the thread from ??? but using temporal models instead of simply probabilistic ones. We will first look at a sequential decision theory in the special case, where the environment is stochastic, but fully observable (Markov decision processes) and then lift that to obtain POMDPs and present an agent design based on that.

Outline

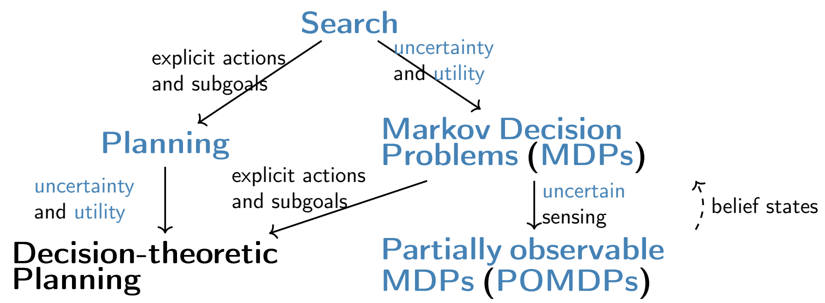
We will now combine the ideas of stochastic process with that of acting based on maximizing expected utility:

- ▷ Markov decision processes (MDPs) for sequential environments.
- ▷ Value/policy iteration for computing utilities in MDPs.
- ▷ Partially observable MDP (POMDPs).
- ▷ Decision theoretic agents for POMDPs.

26.1 Sequential Decision Problems

Sequential Decision Problems

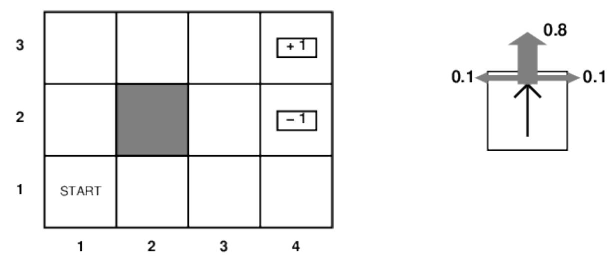
- ▷ **Definition 26.1.1.** In sequential decision problems, the agent's utility depends on a sequence of decisions (or their result states).
- ▷ **Definition 26.1.2.** Utility functions on action sequences are often expressed in terms of immediate rewards that are incurred upon reaching a (single) state.
- ▷ **Methods:** depend on the environment:
  - ▷ If it is fully observable $\rightsquigarrow$ Markov decision process (MDPs)
  - ▷ else $\rightsquigarrow$ partially observable MDP (POMDP).
- ▷ Sequential decision problems incorporate utilities, uncertainty, and sensing.
- ▷ **Preview:** Search problems and planning tasks are special cases.



We will fortify our intuition by an example. It is specifically chosen to be very simple, but to exhibit all the peculiarities of Markov decision problems, which we will generalize from this example.

Markov Decision Problem: Running Example

▷ **Example 26.1.3 (Running Example: The 4x3 World).** A (fully observable) 4×3 environment with non-deterministic actions:



- ▹ States $s \in \mathcal{S}$, actions $a \in \mathcal{A}s$.
- ▹ Transition model: $P(s'|s,a) \,\widehat{=}$ probability that a in s leads to s'.
- ▹ reward function:

$$R(s) := \begin{cases} -0.04 & \text{if (small penalty) for nonterminal states} \\ \pm 1 & \text{if for terminal states} \end{cases}$$

Perhaps what is more interesting than the components of an MDP is that is *not* a component: a belief and/or sensor model. Recall that MDPs are for fully observable environments.

Markov Decision Process

▷ **Motivation:** Let us (for now) consider sequential decision problems in a fully observable, stochastic environment with a Markovian transition model on a *finite* set of states and an additive reward function. (We will switch to partially observable ones later)

▷ **Definition 26.1.4.** A Markov decision process (MDP) $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, s_0, R \rangle$ consists of

- ▹ a set of $\mathcal{S}$ of states (with initial state $s_0 \in \mathcal{S}$),

- for every state s, a set of actions $\mathcal{A}s$.
- a transition model $\mathcal{T}(s,a) = \mathbb{P}(\mathcal{S}|s,a)$, and
- a reward function $R\colon \mathcal{S} \to \mathbb{R}$; we call $R(s)$ a reward.

▷ **Idea:** We use the rewards as a utility function: The goal is to choose actions such that the expected *cumulative* rewards for the "foreseeable future" is maximized

$\Rightarrow$ need to take future actions and future states into account

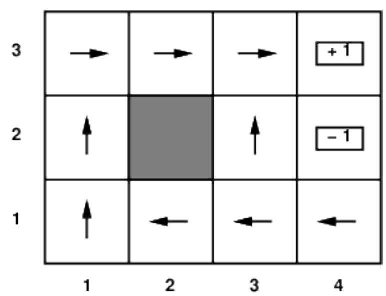
Solving MDPs

▷ In MDPs, the aim is to find an optimal policy $\pi(s)$, which tells us the best action for every possible state s. (because we can't predict where we might end up, we need to consider all states)

▷ **Definition 26.1.5.** A policy π for an MDP is a function mapping each state s to an action $a \in \mathcal{A}s$.

An optimal policy is a policy that maximizes the expected total rewards. (for some notion of "total"...)

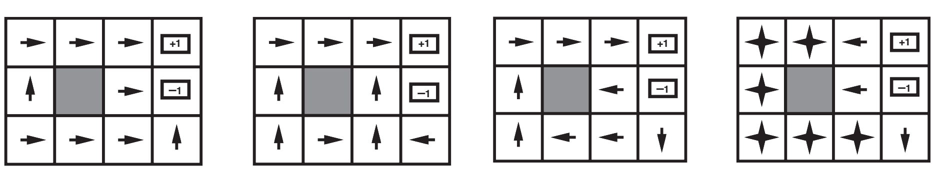
▷ **Example 26.1.6.** Optimal policy when state penalty $R(s)$ is 0.04:



Note: When you run against a wall, you stay in your square.

Risk and Reward

▷ **Example 26.1.7.** Optimal policy depends on the reward function $R(s)$.



$R(s) < -1.6284$ $\quad$ $-0.4278 < R(s) < -0.0850$ $\quad$ $-0.0221 < R(s) < 0$ $\quad$ $R(s) > 0$

▷ **Question:** Explain what you see in a qualitative manner!

▷ **Answer:** reserved for the plenary sessions ⇝ be there!

26.2 Utilities over Time

In this section we address the problem that even if the transition models are stationary, the utilities may not be. In fact we generally have to take the utilities of state sequences into account in sequential decision problems. If we can derive a notion of the utility of a (single) state from that, we may be able to reuse the machinery we introduced above, so that is exactly what we will attempt.

Utility of state sequences

Why rewards?

▷ **Recall:** We cannot observe/assess utility functions, only preferences ⇝ induce utility functions from rational preferences

▷ **Problem:** In MDPs we need to understand preferences between *sequences* of states.

▷ **Definition 26.2.1.** We call preferences on reward sequences stationary, iff

$$[r, r_0, r_1, r_2, \ldots] \succ [r, r'_0, r'_1, r'_2, \ldots] \Leftrightarrow [r_0, r_1, r_2, \ldots] \succ [r'_0, r'_1, r'_2, \ldots]$$

(i.e. rewards over time are "independent" of each other)

▷ Good news:

Theorem 26.2.2. *For stationary preferences, there are only two ways to combine rewards over time.*

- ▷ *additive rewards:* $U([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \cdots$
- ▷ *discounted rewards:* $U([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots$ *where* $0 \leq \gamma \leq 1$ *is called discount factor.*

⇒ we can reduce utilities over time to rewards on individual states

Utilities of State Sequences

Problem: Infinite lifetimes ⇝ additive rewards may become infinite.

Possible Solutions:

1. Finite horizon: terminate utility computation at a fixed time T

$$U([s_0, \ldots, s_\infty]) = R(s_0) + \cdots + R(s_T)$$

⇝ nonstationary policy: $\pi(s)$ depends on time left.

2. If there are absorbing states: for any policy π agent eventually "dies" with probability 1 ⇝ expected utility of every state is finite.

3. Discounting: assuming $\gamma < 1$, $R(s) \leq R_{\mathrm{max}}$,

$$U([s_0, s_1, \ldots]) = \sum_{t=0}^{\infty} \gamma^t R(s_t) \leq \sum_{t=0}^{\infty} \gamma^t R_{\mathrm{max}} = R_{\mathrm{max}}/(1-\gamma)$$

Smaller $\gamma \rightsquigarrow$ shorter horizon.

We will only consider discounted rewards in this course

Why discounted rewards?

Discounted rewards are both convenient and (often) realistic:

- ▷ stationary preferences imply (additive rewards or) discounted rewards anyway,
- ▷ discounted rewards lead to finite utilities for (potentially) infinite sequences of states (we can compute expected utilities for the entire future),
- ▷ discounted rewards lead to stationary policies, which are easier to compute and often more adequate (unless we know that remaining time matters),
- ▷ discounted rewards mean we value *short-term gains* over *long-term gains* (all else being equal), which is often realistic (e.g. the same amount of money gained *early* gives more opportunity to spend/invest $\Rightarrow$ potentially more utility in the long run)
- ▷ we can interpret the discount factor as a measure of *uncertainty about future rewards* $\Rightarrow$ more robust measure in uncertain environments.

Utility of States

Remember: Given a sequence of states $S = s_0, s_1, s_2, \ldots$, and a discount factor $0 \leq \gamma < 1$, the utility of the sequence is

$$U(S) = \sum_{t=0}^{\infty} \gamma^t R(s_t)$$

Definition 26.2.3. Given a policy π and a starting state s_0, let $S_{s_0}^{\pi}$ be the random variable giving the sequence of states resulting from executing π at every state starting at s_0. (Since the environment is stochastic, we don't know the exact sequence.)

Then the expected utility obtained by executing π starting in s_0 is given by

$$U^{\pi}(s_0) := \mathrm{EU}(S_{s_0}^{\pi}).$$

We define the optimal policy $\pi_{s_0}^{*} := \underset{\pi}{\operatorname{argmax}}\, U^{\pi}(s_0)$.

Note: This is perfectly well-defined, but almost always computationally infeasible. (requires considering *all possible (potentially infinite) sequences of states*)

Utility of States (continued)

Observation 26.2.4. *$\pi^*_{s_0}$ is independent of the state s_0.*

Proof sketch: If π^*_a and π^*_b reach point c, then there is no reason to disagree from that point on – or with π^*_c, and we expect optimal policies to "meet at some state" sooner or later.

⚠ Observation 26.2.4 does not hold for finite horizon policies!

Definition 26.2.5. We call $\pi^* := \pi^*_s$ for some s the optimal policy.

Definition 26.2.6. The utility $U(s)$ of a state s is $U^{\pi^*}(s)$.

Remark: $R(s) \mathrel{\hat{=}}$ "immediate reward", whereas $U \mathrel{\hat{=}}$ "long-term reward".

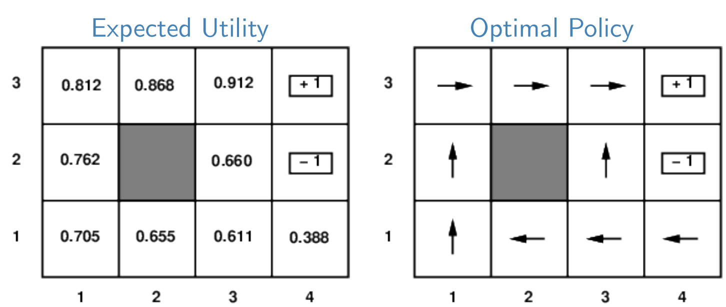
Given the utilities of the states, choosing the best action is just MEU: maximize the expected utility of the immediate successor states

$$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \left(\sum_{s'} P(s'|s,a) \cdot U(s')\right)$$

$\Rightarrow$ given the "true" utilities, we can compute the optimal policy and vice versa.

Utility of States (continued)

▷ **Example 26.2.7 (Running Example Continued).**



▷ **Question:** Why do we go left in $(3,1)$ and not up? (follow the utility)

26.3 Value/Policy Iteration

Dynamic programming: the Bellman equation

▷ **Problem:** We have defined $U(s)$ via the optimal policy: $U(s) := U^{\pi^*}(s)$, but how to compute it without knowing π^*?

▷ **Observation:** A simple relationship among utilities of neighboring states:

expected sum of rewards = current reward + $\gamma \cdot$ exp. reward sum after best action

▷ **Theorem 26.3.1 (Bellman equation (1957)).**

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

We call this equation the Bellman equation

▷ **Example 26.3.2.** $U(1,1) = -0.04$

$+\ \gamma\ \max\{0.8U(1,2) + 0.1U(2,1) + 0.1U(1,1),$ *up*

$0.9U(1,1) + 0.1U(1,2)$ *left*

$0.9U(1,1) + 0.1U(2,1)$ *down*

$0.8U(2,1) + 0.1U(1,2) + 0.1U(1,1)\}$ *right*

▷ **Problem:** One equation/state $\leadsto$ n nonlinear ($\max$ isn't) equations in n unknowns.
$\leadsto$ cannot use linear algebra techniques for solving them.

Value Iteration Algorithm

▷ **Idea:** We use a simple iteration scheme to find a fixpoint:

1. start with arbitrary utility values,
2. update to make them locally consistent with the Bellman equation,
3. everywhere locally consistent $\leadsto$ global optimality.

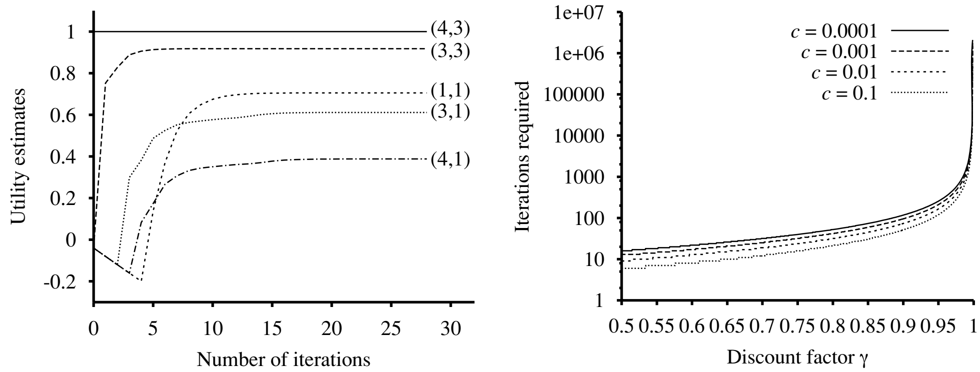
▷ **Definition 26.3.3.** The value iteration algorithm for utilitysutility function is given by

```
function VALUE−ITERATION (mdp,ε) returns a utility fn.
  inputs: mdp, an MDP with states S, actions A(s), transition model P(s′|s,a),
          rewards R(s), and discount γ
        ε, the maximum error allowed in the utility of any state
  local variables: U, U′, vectors of utilities for states in S, initially zero
          δ, the maximum change in the utility of any state in an iteration
  repeat
     U := U′; δ := 0
     for each state s in S do
        U′[s] := R(s) + γ · max_{a∈A(s)} (∑_{s′} U[s′] · P(s′|s,a))
        if |U′[s] − U[s]| > δ then δ := |U′[s] − U[s]|
     until δ < ε(1 − γ)/γ
     return U
```

▷ **Remark:** Retrieve the optimal policy with $\pi[s] := \operatorname*{argmax}_{a \in A(s)} \left(\sum_{s'} U[s'] \cdot P(s'|s,a)\right)$

Value Iteration Algorithm (Example)

▷ **Example 26.3.4 (Iteration on 4x3).**



(where $\varepsilon = c \cdot R_{max}$)

Michael Kohlhase: Artificial Intelligence 2 887 2025-05-01

Convergence

▷ **Definition 26.3.5.** The maximum norm is defined as $\|U\| = \max_s |U(s)|$, so $\|U - V\| =$ maximum difference between U and V.

▷ Let U^t and U^{t+1} be successive approximations to the true utility U during value iteration.

▷ **Theorem 26.3.6.** *For any two approximations U^t and V^t*

$$\left\|U^{t+1} - V^{t+1}\right\| \leq \gamma \left\|U^t - V^t\right\|$$

I.e., any distinct approximations get closer to each other over time
In particular, any approximation gets closer to the true U over time
$\Rightarrow$ value iteration converges to a unique, stable, optimal solution.

▷ **Theorem 26.3.7.** *If $\left\|U^{t+1} - U^t\right\| < \epsilon$, then $\left\|U^{t+1} - U\right\| < 2\epsilon\gamma/1 - \gamma$*
(once the change in U^t becomes small, we are almost done.)

▷ **Remark:** The policy resulting from U^t may be optimal long before the utilities convergence!

Michael Kohlhase: Artificial Intelligence 2 888 2025-05-01

So we see that iteration with Bellman updates will always converge towards the utility of a state, even without knowing the optimal policy. That gives us a first way of dealing with sequential decision problems: we compute utility functions based on states and then use the standard MEU machinery. We have seen above that optimal policies and state utilities are essentially interchangeable: we can compute one from the other. This leads to another approach to computing state utilities: policy iteration, which we will discuss now.

Policy Iteration

▷ **Recap:** Value iteration computes utilities $\rightsquigarrow$ optimal policy by MEU.

▷ This even works if the utility estimate is inaccurate. ($\leftsquigarrow$ policy loss small)

▷ **Idea:** Search for optimal policy and utility values simultaneously [How60]: Iterate

- policy evaluation: given policy π_i, calculate $U_i = U^{\pi_i}$, the utility of each state were π_i to be executed.
- policy improvement: calculate a new MEU policy π_{i+1} using 1 lookahead

Terminate if policy improvement yields no change in computed utilities.

- **Observation 26.3.8.** *Upon termination U_i is a fixpoint of Bellman update $\leadsto$ Solution to Bellman equation $\leadsto$ π_i is an optimal policy.*
- **Observation 26.3.9.** *Policy improvement improves policy and policy space is finite $\leadsto$ termination.*

Policy Iteration Algorithm

- **Definition 26.3.10.** The policy iteration algorithm is given by the following pseudocode:

```
function POLICY−ITERATION(mdp) returns a policy
  inputs: mdp, and 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
                   π a policy indexed by state, initially random,
  repeat
     U := POLICY−EVALUATION(π,U,mdp)
     unchanged? := true
     foreach state s in X do
        if max_{a∈A(s)} (∑_{s′} P(s′|s,a) · U(s′)) > ∑_{s′} P(s′|s,π[s]) · U(s′) then do
           π[s] := argmax_{b∈A(s)} (∑_{s′} P(s′|s,b) · U(s′))
           unchanged? := false
  until unchanged?
  return π
```


Policy Evaluation

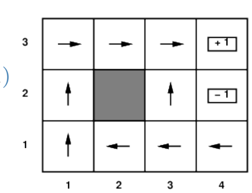
- **Problem:** How to implement the POLICY−EVALUATION algorithm?
- **Solution:** To compute utilities given a fixed π: For all s we have

$$U(s) = R(s) + \gamma(\sum_{s'} U(s') \cdot P(s'|s,\pi(s)))$$

(i.e. Bellman equation with the maximum replaced by the current policy π)

- **Example 26.3.11 (Simplified Bellman Equations for π).**

$$\begin{aligned} U_i(1,1) &= -0.04 + 0.8U_i(1,2) + 0.1U_i(1,1) + 0.1U_i(2,1) \\ U_i(1,2) &= -0.04 + 0.8U_i(1,3) + 0.1U_i(1,2) \\ &\vdots \end{aligned}$$



▷ **Observation 26.3.12.** *n simultaneous linear equations in n unknowns, solve in $\mathcal{O}(n^3)$ with standard linear algebra methods.*

Michael Kohlhase: Artificial Intelligence 2 891 2025-05-01

Modified Policy Iteration

▷ Value iteration requires many iterations, but each one is cheap.

▷ Policy iteration often converges in few iterations, but each is expensive.

▷ **Idea:** Use a few steps of value iteration (but with π fixed), starting from the value function produced the last time to produce an approximate value determination step.

▷ Often converges much faster than pure VI or PI.

▷ Leads to much more general algorithms where Bellman value updates and Howard policy updates can be performed locally in any order.

▷ **Remark:** Reinforcement learning algorithms operate by performing such updates based on the observed transitions made in an initially unknown environment.

Michael Kohlhase: Artificial Intelligence 2 892 2025-05-01

26.4 Partially Observable MDPs

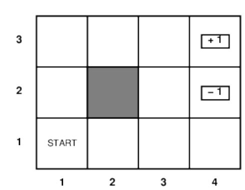
We will now lift the last restriction we made in the decision problems for our agents: in the definition of Markov decision processes we assumed that the environment was fully observable. As we have seen ??? this entails that the optimal policy only depends on the current state.

Partial Observability

▷ **Definition 26.4.1.** A partially observable MDP (a POMDP for short) is a MDP together with an observation model O that has the sensor Markov property and is stationary: $O(s,e) = P(e|s)$.

▷ **Example 26.4.2 (Noisy 4x3 World).**

Add a partial and/or noisy sensor.
e.g. count number of adjacent walls $(1 \leq w \leq 2)$
with 0.1 error (noise)
If sensor reports 1, we are in $(3,?)$ (probably)



▷ **Problem:** Agent does not know which state it is in $\leadsto$ makes no sense to talk about policy $\pi(s)$!

▷ **Theorem 26.4.3 (Astrom 1965).** *The optimal policy in a POMDP is a function $\pi(b)$ where b is the belief state (probability distribution over states).*

▷ **Idea:** Convert a POMDP into an MDP in belief state space, where $\mathcal{T}(b, a, b')$ is the probability that the new belief state is b' given that the current belief state is b and the agent does a. I.e., essentially a filtering update step.

POMDP: Filtering at the Belief State Level

▷ **Recap:** Filtering updates the belief state for new evidence.

▷ For POMDPs, we also need to consider actions. (but the effect is the same)

▷ If b is the previous belief state and agent does action $A = a$ and then perceives $E = e$, then the new belief state is

$$b' = \alpha(\mathbb{P}(E = e|s') \cdot (\sum_{s} \mathbb{P}(s'|S = s, A = a) \cdot b(s)))$$

We write $b' = \mathrm{FORWARD}(b, a, e)$ in analogy to recursive state estimation.

▷ **Fundamental Insight for POMDPs:** The optimal action only depends on the agent's current belief state. (good, it does not know the state!)

▷ **Consequence:** The optimal policy can be written as a function $\pi^*(b)$ from belief states to actions.

▷ **Definition 26.4.4.** The POMDP decision cycle is to iterate over

1. Given the current belief state b, execute the action $a = \pi^*(b)$
2. Receive percept e.
3. Set the current belief state to $\mathrm{FORWARD}(b, a, e)$ and repeat.

▷ **Intuition:** POMDP decision cycle is search in belief state space.

Partial Observability contd.

▷ **Recap:** The POMDP decision cycle is search in belief state space.

▷ **Observation 26.4.5.** *Actions change the belief state, not just the (physical) state.*

▷ **Thus** POMDP solutions automatically include information gathering behavior.

▷ **Problem:** The belief state is continuous: If there are n states, b is an n-dimensional real-valued vector.

▷ **Example 26.4.6.** The belief state of the 4x3 world is a 11 dimensional continuous space. (11 states)

▷ **Theorem 26.4.7.** *Solving POMDPs is very hard!* *(actually, **PSPACE** hard)*

▷ **In particular,** none of the algorithms we have learned applies. (discreteness assumption)

- The real world is a POMDP (with initially unknown transition model T and sensor model O)

Reducing POMDPs to Belief-State MDPs

- **Idea:** Calculating the probability that an agent in belief state b reaches belief state b' after executing action a.
  - if we knew the action and the *subsequent* percept e, then $b' = \mathrm{FORWARD}(b, a, e)$. (deterministic update to the belief state)
  - but we don't, since b' depends on e. (let's calculate $P(e|a, b)$)
- **Idea:** To compute $P(e|a, b)$ — the probability that e is perceived after executing a in belief state b — sum up over all actual states the agent might reach:

$$\begin{aligned} P(e|a,b) &= \sum_{s'} P(e|a,s',b) \cdot P(s'|a,b) \\ &= \sum_{s'} P(e|s') \cdot P(s'|a,b) \\ &= \sum_{s'} P(e|s') \cdot (\sum_{s} P(s'|s,a), b(s)) \end{aligned}$$

  Write the probability of reaching b' from b, given action a, as $P(b'|b, a)$, then

$$\begin{aligned} P(b'|b,a) &= P(b'|a,b) = \sum_{e} P(b'|e,a,b) \cdot P(e|a,b) \\ &= \sum_{e} P(b'|e,a,b) \cdot (\sum_{s'} P(e|s') \cdot (\sum_{s} P(s'|s,a), b(s))) \end{aligned}$$

  where $P(b'|e, a, b)$ is 1 if $b' = \mathrm{FORWARD}(b, a, e)$ and 0 otherwise.
- **Observation:** This equation defines a transition model for belief state space!
- **Idea:** We can also define a reward function for belief states:

$$\rho(b) := \sum_{s} b(s) \cdot R(s)$$

  i.e., the expected reward for the actual states the agent might be in.
- Together, $P(b'|b, a)$ and $\rho(b)$ define an (observable) MDP on the space of belief states.
- **Theorem 26.4.8.** *An optimal policy $\pi^*(b)$ for this MDP, is also an optimal policy for the original POMDP.*
- **Upshot:** Solving a POMDP on a physical state space can be reduced to solving an MDP on the corresponding belief state space.
- **Remember:** The belief state is always observable to the agent, by definition.

Ideas towards Value-Iteration on POMDPs

- **Recap:** The value iteration algorithm from ??? computes one utility value per state.
- **Problem:** We have infinitely many belief states $\rightsquigarrow$ be more creative!
- **Observation:** Consider an optimal policy π^*
  - applied in a specific belief state b: π^* generates an action,
  - for each subsequent percept, the belief state is updated and a new action is generated . . .

  For this specific b: $\pi^* \widehat{=}$ a conditional plan!
- **Idea:** Think about conditional plans and how the expected utility of executing a fixed conditional plan varies with the initial belief state. (instead of optimal policies)

Definition 26.4.9. Given a set of percepts E and a set of actions A, a conditional plan is either an action $a \in A$, or a tuple $\langle a, E', p_1, p_2 \rangle$ such that $a \in A, E' \subseteq E$, and p_1, p_2 are conditional plans.

It represents the strategy "First execute a, If we subsequently perceive $e \in E'$, continue with p_1, otherwise continue with p_2."

The depth of a conditional plan p is the maximum number of actions in any path from p before reading a single action plan.

Expected Utilities of Conditional Plans on Belief States

- **Observation 1:** Let p be a conditional plan and $\alpha_p(s)$ the utility of executing p in state s.
  - the expected utility of p in belief state b is $\sum_s b(s) \cdot \alpha_p(s) \widehat{=} b \cdot \alpha_p$ as vectors.
  - the expected utility of a fixed conditional plan varies linearly with b
  - $\rightsquigarrow$ the "*best* conditional plan to execute" corresponds to a hyperplane in belief state space.
- **Observation 2:** We can replace the *original* actions by conditional plans on those actions!

  Let π^* be the subsequent optimal policy. At any given belief state b,
  - π^* will choose to execute the conditional plan with highest expected utility
  - the expected utility of b under the π^* is the utility of that plan:

$$U(b) = U^{\pi^*}(b) = \max_b (b \cdot \alpha_p)$$

  - If the optimal policy π^* chooses to execute p starting at b, then it is reasonable to expect that it might choose to execute p in belief states that are very close to b;
  - if we bound the depth of the conditional plans, then there are only finitely many such plans
  - the continuous space of belief states will generally be divided into regions, each corresponding to a particular conditional plan that is optimal in that region.
- **Observation 3 (conbined):** The utility function $U(b)$ on belief states, being the maximum of a collection of hyperplanes, is piecewise linear and convex.

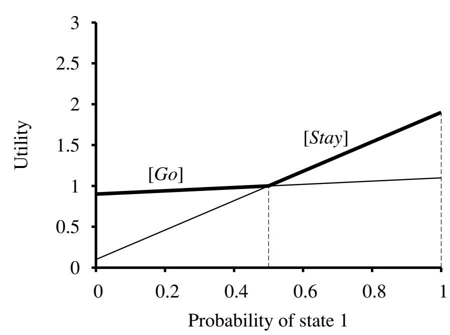
A simple Illustrating Example

- **Example 26.4.10.** A world with states S_0 and S_1, where $R(S_0) = 0$ and $R(S_1) = 1$ and two actions:
  - "Stay" stays put with probability 0.9
  - "Go" switches to the other state with probability 0.9.
  - The sensor reports the correct state with probability 0.6.

  Obviously, the agent should "Stay" when it thinks it's in state S_1 and "Go" when it thinks it's in state S_0.
- The belief state has dimension 1. (the two probabilities sum up to 1)
- Consider the one-step plans $[Stay]$ and $[Go]$ and their direct utilities:

$$
\begin{aligned}
\alpha_{([Stay])}(S_0) &= 0.9R(S_0) + 0.1R(S_1) = 0.1\\
\alpha_{([stay])}(S_1) &= 0.9R(S_1) + 0.1R(S_0) = 0.9\\
\alpha_{([go])}(S_0) &= 0.9R(S_1) + 0.1R(S_0) = 0.9\\
\alpha_{([go])}(S_1) &= 0.9R(S_0) + 0.1R(S_1) = 0.1
\end{aligned}
$$

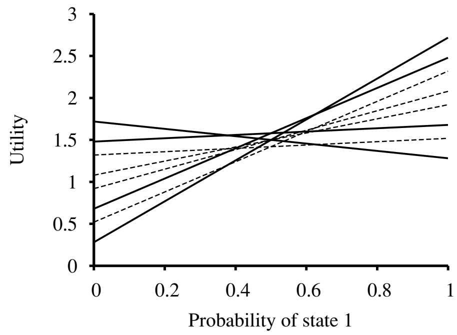
- Let us visualize the hyperplanes $b \cdot \alpha_{([Stay])}$ and $b \cdot \alpha_{([Go])}$.



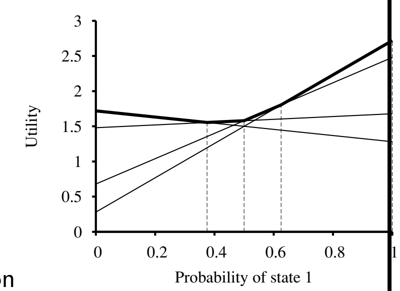
  - The maximum represents the utility function for the finite-horizon problem that allows just one action
  - in each "piece" the optimal action is the first action of the corresponding plan.
  - Here the optimal one-step policy is to "Stay" when $b(1) > 0.5$ and "Go" otherwise.
- compute the utilities for conditional plans of depth 2 by considering
  - each possible first action,
  - each possible subsequent percept, and then
  - each way of choosing a depth-1 plan to execute for each percept:

  There are eight of depth 2:

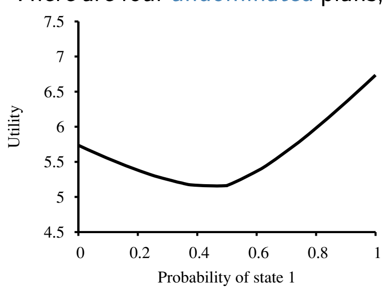
$$[Stay, \text{if } P = 0 \text{ then } Stay \text{ else } Stay \text{ fi}], [Stay, \text{if } P = 0 \text{ then } Stay \text{ else } Go \text{ fi}], \ldots$$



Four of them (dashed lines) are suboptimal for the whole belief space
We call them dominated (they can be ignored)



▷ There are four undominated plans, each optimal in their region



▷ **Idea:** Repeat for depth 3 and so on.

▷ **Theorem 26.4.11 (POMDP Plan Utility).** *Let p be a depth-d conditional plan whose initial action is a and whose depth-$d-1$-subplan for percept e is $p.e$, then*

$$\alpha_p(s) = R(s) + \gamma(\sum_{s'} P(s'|s,a)(\sum_e P(e|s') \cdot \alpha_{p.e}(s')))$$

▷ This recursion naturally gives us a value iteration algorithm,

A Value Iteration Algorithm for POMDPs

Definition 26.4.12. The POMDP value iteration algorithm for POMDPs is given by recursively updating

$$\alpha_p(s) = R(s) + \gamma(\sum_{s'} P(s'|s,a)(\sum_e P(e|s') \cdot \alpha_{p.e}(s')))$$

Observations: The complexity depends primarily on the generated plans:

- ▷ Given $|A|$ actions and $|E|$ possible observations, there are are $|A|^{|E|^{d-1}}$ distinct depth-d plans.
- ▷ Even for the example with $d = 8$, we have 2255 (144 undominated)
- ▷ The elimination of dominated plans is essential for reducing this doubly exponential growth (but they are already constructed)

Hopelessly inefficient in practice – even the 3x4 POMDP is too hard!

26.5 Online Agents with POMDPs

In the last section we have seen that even though we can in principle compute utilities of states – and thus use the MEU principle – to make decisions in sequential decision problems, all methods based on the "lifting idea" are hopelessly inefficient.

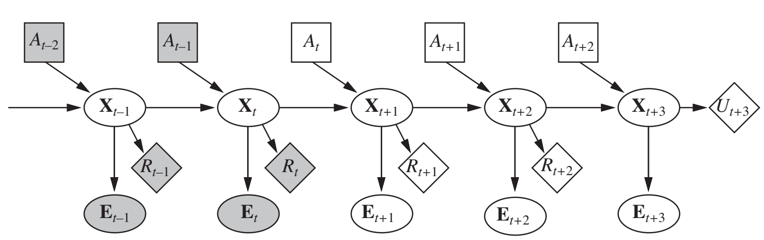
This section describes a different, approximate method for solving POMDPs, one based on look-ahead search.

DDN: Decision Networks for POMDPs

- ▷ **Idea:** Let's try to use the computationally efficient representations (dynamic Bayesian networks and decision networks) for POMDPs.
- ▷ **Definition 26.5.1.** A dynamic decision network (DDN) is a graph-based representation of a POMDP, where
  - ▷ Transition and sensor model are represented as a DBN.
  - ▷ Action nodes and utility nodes are added as in decision networks.
- ▷ In a DDN, a filtering algorithm is used to incorporate each new percept and action and to update the belief state representation.
- ▷ Decisions are made in DDN by projecting forward possible action sequences and choosing the best one.
- ▷ DDNs – like the DBNs they are based on – are factored representations $\rightsquigarrow$ typically exponential complexity advantages!

Structure of DDNs for POMDPs

- ▷ **DDN for POMDPs:** The generic structure of a dymamic decision network at time t is



- ▷ POMDP state S_t becomes a set of random variables $\mathbf{X}_t$
- ▷ there may be multiple evidence variables $\mathbf{E}_t$
- ▷ Action at time t denoted by A_t. agent must choose a value for A_t.
- ▷ Transition model: $\mathbb{P}(\mathbf{X}_{t+1}|\mathbf{X}_t, A_t)$; sensor model: $\mathbb{P}(\mathbf{E}_t|\mathbf{X}_t)$.
- ▷ Reward functions R_t and utility U_t of state S_t.
- ▷ Variables with known values are gray, rewards for $t = 0, \ldots, t+2$, but utility for $t+3$($\widehat{=}$ discounted sum of rest)

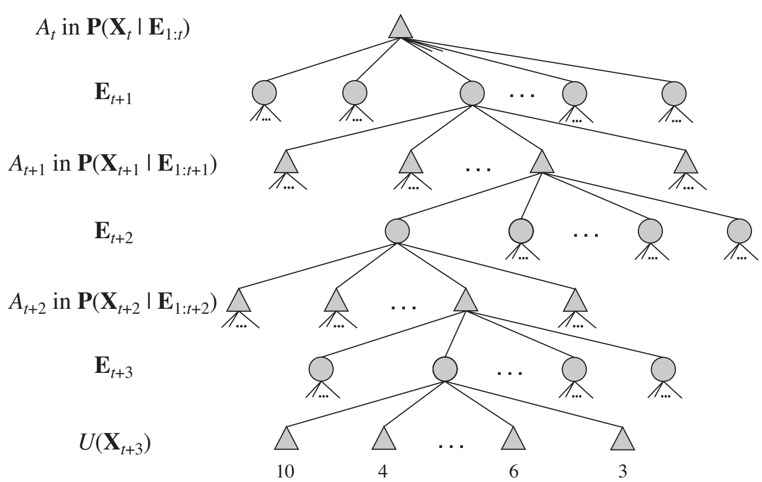
▷ **Problem:** How do we compute with that?

▷ **Answer:** All POMDP algorithms can be adapted to DDNs! (only need CPTs)

Lookahead: Searching over the Possible Action Sequences

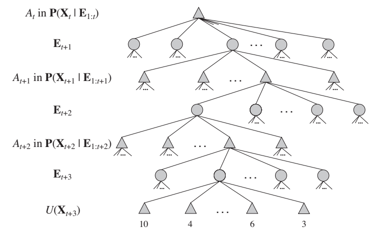
▷ **Idea:** Search over the tree of possible action sequences (like in game-play)

▷ Part of the lookahead solution of the DDN above (three steps lookahead)



- ▷ circle $\widehat{=}$ chance nodes (the environment decides)
- ▷ triangle $\widehat{=}$ belief state (each action decision is taken there)

Designing Online Agents for POMDPs



- ▷ Belief state at triangle computed by filtering with actions/percepts leading to it
  - ▷ for decision A_{t+i} will use percepts $\mathbf{E}_{t+1:t+i}$ (even if values at time t unknown)
  - ▷ thus a POMDP agent automatically takes into account the value of information and executes information gathering actions where appropriate.
- ▷ **Observation:** Time complexity for exhaustive search up to depth d is $\mathcal{O}(|A|^d \cdot |\mathbf{E}|^d)$($|A| \widehat{=}$ number of actions, $|\mathbf{E}| \widehat{=}$ number of percepts)
- ▷ **Upshot:** Much better than POMDP value iteration with $\mathcal{O}(|A|^{|E|^{d-1}})$.
- ▷ **Empirically:** For problems in which the discount factor γ is not too close to 1, a shallow search is often good enough to give near-optimal decisions.

Summary

- ▷ Decision theoretic agents for sequential environments
- ▷ Building on temporal, probabilistic models/inference (dynamic Bayesian networks)
- ▷ MDPs for fully observable case.
- ▷ Value/Policy Iteration for MDPs $\rightsquigarrow$ optimal policies.
- ▷ POMDPs for partially observable case.
- ▷ POMDPs$\widehat{=}$ MDP on belief state space.
- ▷ The world is a POMDP with (initially) unknown transition and sensor models.

Bibliography

[DF31] B. De Finetti. "Sul significato soggettivo della probabilita". In: *Fundamenta Mathematicae* 17 (1931), pp. 298–329.

[How60] R. A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, 1960.

[Kee74] R. L. Keeney. "Multiplicative utility functions". In: *Operations Research* 22 (1974), pp. 22–34.

[Luc96] Peter Lucas. "Knowledge Acquisition for Decision-theoretic Expert Systems". In: *AISB Quarterly* 94 (1996), pp. 23–33. URL: `https://www.researchgate.net/publication/2460438_Knowledge_Acquisition_for_Decision-theoretic_Expert_Systems`.

[Pra+94] Malcolm Pradhan et al. "Knowledge Engineering for Large Belief Networks". In: *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*. UAI'94. Seattle, WA: Morgan Kaufmann Publishers Inc., 1994, pp. 484–490. ISBN: 1-55860-332-8. URL: `http://dl.acm.org/citation.cfm?id=2074394.2074456`.

[RN03] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 2nd ed. Pearso n Education, 2003. ISBN: 0137903952.