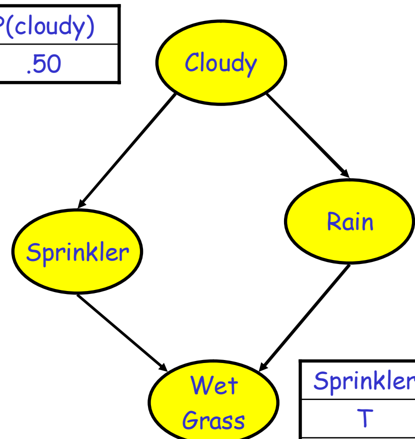
14.5 Approximate Inference

- Given the intractability of exact inference, it is essential to consider approximate inference methods
- Approximation is based on random sampling from a known probability distribution (Monte Carlo algorithms)
- E.g., an unbiased coin can be thought of as a random variable Coin with values [heads, tails] and a prior distribution

  $\underline{P}(Coin) = [0.5, 0.5]$
- Sampling from this distribution is exactly like flipping the coin: with probability 0.5 it will return heads, and with probability 0.5 it will return tails
- Given a source of random numbers in the range [0, 1], it is a simple matter to sample any distribution on a single variable

P(cloudy)
.50



Cloudy	P(sprinkler)
T	.10
F	.50

Cloudy	P(rain)
T	.80
F	.20

Sprinkler	Rain	P(wet)
T	T	.99
T	F	.90
F	T	.90
F	F	.00

14.5.1 Direct Sampling Methods

- From a Bayesian network that has no evidence associated with it, we can sample each variable in turn, in topological order
- When the values of parent nodes have been drawn, we know from which distribution we have to sample the child
- Let is fix a topological order for the nodes of our network: [Cloudy, Sprinkler, Rain, WetGrass]
  1. Sample from **P**(Cloudy) = [0.5, 0.5]; suppose this returns True
  2. Sample from **P**(Sprinkler | cloudy) = [0.1, 0.9]; suppose this returns False
  3. Sample from **P**(Rain | cloudy) = [0.8, 0.2]; suppose this returns True
  4. Sample from **P**(WetGrass | ¬sprinkler, rain) = [0.9, 0.1]; suppose this returns True

- From the prior joint distribution specified by the network we have drawn the event [True, False, True, True]
- Let $S_{PS}(x_1, \ldots, x_n)$ be the probability that a specific event is generated by this prior sampling algorithm
- Just looking at the sampling process, we have

$$S_{PS}(x_1, \ldots, x_n) = \prod_{i=1,\ldots,n} P(x_i \mid parents(X_i))$$

- On the other hand, this is also the probability of the event according to the Bayesian net's representation of the joint distribution; i.e.:

$$S_{PS}(x_1, \ldots, x_n) = P(x_1, \ldots, x_n)$$

- Let $N_{PS}(x_1, \ldots, x_n)$ be the frequency of the specific event $x_1, \ldots, x_n$ and that there are N total samples

- We expect this frequency to converge in the limit to its expected value according to the sampling probability:

$$\lim_{N\to\infty} N_{PS}(x_1, \ldots, x_n)/N = S_{PS}(x_1, \ldots, x_n) = P(x_1, \ldots, x_n)$$

- E.g., $S_{PS}([True, False, True, True]) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324$, hence in the limit of large N, we expect 32.4% of the samples to be of this event
- The estimate of prior sampling is *consistent* in the sense that the estimated probability becomes exact in the large-sample limit

- One can also produce a consistent estimate of the probability of any partially specified event $x_1, \ldots, x_m$, where $m \le n$:

$$P(x_1, \ldots, x_m) \approx N_{PS}(x_1, \ldots, x_m)/N$$

- Let us denote by $P_{data}(\cdot)$ the probability estimated from a sample

Rejection Sampling

- To determine a conditional probability $P(X \mid \mathbf{e})$ we could apply the following simple sampling approach:
  1. Generate samples from the prior distribution specified by the network
  2. Reject all those that do not match the evidence $\mathbf{e}$
  3. The estimate $P_{data}(X = x \mid \mathbf{e})$ is obtained by counting how often $X = x$ occurs in the remaining samples

- The estimated distribution $\underline{\mathbf{P}}_{data}(X \mid \mathbf{e})$ that the algorithm returns is, by the definition of the algorithm

$$\alpha\, \mathbf{N}_{PS}(X, \mathbf{e}) = \mathbf{N}_{PS}(X, \mathbf{e}) / N_{PS}(\mathbf{e})$$

- As an estimate of the probability of a partially specified event it is consistent

$$\underline{\mathbf{P}}_{data}(X \mid \mathbf{e}) \approx \underline{\mathbf{P}}(X, \mathbf{e}) / P(\mathbf{e}) = \underline{\mathbf{P}}(X \mid \mathbf{e})$$

- Let us generate 100 samples in order to estimate the distribution **P**(Rain | Sprinkler = True)
  - Suppose that 73 of those that we generate have Sprinkler = False and are rejected
  - The remaining 27 have Sprinkler = True
  - Out of them 8 have Rain = True and 19 have Rain = False
- Hence, we now have $\mathbf{P}_{data}$(Rain | sprinkler) ≈ [0.296, 0.704], while the true distribution is [0.3, 0.7]
- As more samples are collected, the estimate will converge to the true answer
- The standard deviation of the error in each probability will be proportional to $1/\sqrt{n}$, where n is the number of samples
- The large number of rejected samples is a big problem:

  The fraction of samples consistent with the evidence drops exponentially as the number of evidence variables grows

Likelihood weighting

- Rejection sampling is inefficient because it ends up rejecting so many of the generated samples
- To avoid generating needles samples that anyhow get rejected, let us fix the values for the evidence variables **E** and sample only the remaining variables X and **Y**
- Not all events are equal, however
- Each event is weighted by the *likelihood* that the event accords to the evidence
- The likelihood is measured by the product of the conditional probabilities for each evidence variable, given its parents
- Intuitively, events in which the actual evidence appears unlikely should be given less weight

- To answer the query $\underline{P}$(Rain | sprinkler, wetgrass), the weight w is first set to 1.0
- Sample from $\underline{P}$(Cloudy) = [0.5, 0.5]; suppose this returns True
- Sprinkler is an evidence variable with value True, therefore we update the weight

$$w \leftarrow w \times P(sprinkler \mid cloudy) = 0.1$$

- Sample from $\underline{P}$(Rain | cloudy) = [0.8, 0.2]; suppose this returns True
- WetGrass is an evidence variable with value True ⇨

$$w \leftarrow w \times P(wetgrass \mid sprinkler, rain) = 0.099$$

- Hence, the algorithm returns the event [True, True, True, True] with weight 0.099 and this is tallied under Rain = True

- Let us denote **Z** = { X } ∪ **Y**
- The weighted sample algorithm samples each variable in **Z** given its parent values

$$S_{WS}(\mathbf{z}, \mathbf{e}) = \prod_{i=1,\dots,l} P(z_i \mid parents(Z_i))$$

- $Parents(Z_i)$ can include both hidden variables and evidence variables

- The sampling distribution S_{WS} pays some attention to the evidence, unlike the prior distribution P(**z**)
- In S_{WS} the sampled values for each Z_i will be influenced by evidence among Z_i's ancestors
- On the other hand, the true posterior distribution P(**z** | **e**) also takes non-ancestor evidence into account

- The likelihood weight w makes up for the difference between the actual and desired sampling distributions

- Let a given sample $\mathbf{x}$ be composed from $\mathbf{z}$ and $\mathbf{e}$, then

$$w(\mathbf{z}, \mathbf{e}) = \prod_{i=1,\dots,m} P(e_i \mid parents(E_i))$$

- The *weighted* probability of a sample, $S_{WS}(\mathbf{z}, \mathbf{e}) \cdot w(\mathbf{z}, \mathbf{e})$,is

$$\prod_{i=1,\dots,l} P(z_i \mid parents(Z_i)) \cdot \prod_{i=1,\dots,m} P(e_i \mid parents(E_i))$$
$$= P(\mathbf{z}, \mathbf{e}),$$

because the two products cover all the variables in the network

- Now it is easy to show that likelihood weighting estimates are consistent:

$$\begin{aligned}
P_{data}(x \mid \mathbf{e}) &= \alpha \textstyle\sum_{\mathbf{y}} N_{WS}(x, \mathbf{y}, \mathbf{e})\, w(x, \mathbf{y}, \mathbf{e}) && \text{algorithm} \\
&\approx \alpha' \textstyle\sum_{\mathbf{y}} S_{WS}(x, \mathbf{y}, \mathbf{e})\, w(x, \mathbf{y}, \mathbf{e}) && \text{for large } N \\
&= \alpha' \textstyle\sum_{\mathbf{y}} P(x, \mathbf{y}, \mathbf{e}) && \text{by prev. slide} \\
&= \alpha' P(x, \mathbf{e}) \\
&= P(x \mid \mathbf{e})
\end{aligned}$$

- Because likelihood weighting uses all the samples generated, it can be much more efficient than rejection sampling
- It will, however, suffer a degradation in performance as the number of evidence variables increases
- Because most samples will have very low weights, the weighted estimate will be dominated by a tiny fraction of samples

14.5.2 Inference by Markov Chain Simulation

- Markov chain Monte Carlo (MCMC)
- A Monte Carlo algorithm is a randomized algorithm, which can give the false answer with a small probability (vs. Las Vegas algorithm)
- MCMC generates each event by making a random change to the preceding event
- The next state is generated by randomly sampling a value for one of the nonevidence variables X_i, conditioned on the current values in its *Markov blanket*
- The Markov blanket of a variable consists of its parents, children, and children's parents
- MCMC therefore wanders randomly around the state space flipping one variable at a time, but keeping the evidence variables fixed

16 MAKING SIMPLE DECISIONS

- Let us associate each state S with a numeric *utility* $U(S)$, which expresses the desirability of the state
- A nondeterministic action a will have possible outcome states $Result(a) = s'$
- Prior to the execution of a the agent assigns probability $P(Result(a) = s' \mid a, \mathbf{e})$ to each outcome, where $\mathbf{e}$ summarizes the agent's available evidence of the world
- The expected utility of a can now be calculated:

$$EU(a \mid \mathbf{e}) = \sum_{s'} P(Result(a) = s' \mid a, \mathbf{e}) \cdot U(s')$$

- The principle of *maximum expected utility* (MEU) says that a rational agent should choose an action that maximizes the agent's expected utility $\text{argmax}_a\ EU(a \mid \mathbf{e})$
- If we wanted to choose the best sequence of actions using this equation, we would have to enumerate all action sequences, which is clearly infeasible for long sequences
- If the utility function correctly reflects the performance measure by which the behavior is being judged ⇨

  using MEU the agent will achieve the highest possible performance score averaged over the environments in which it could be placed
- Let us model a nondeterministic action with a *lottery* L, where possible outcomes $S_1, \ldots, S_n$ can occur with probabilities $p_1, \ldots, p_n$

$$L = [p_1, S_1; p_2, S_2; \ldots; p_n, S_n]$$

16.2 The Basis of Utility Theory

$A \succ B$ Agent prefers lottery A over B

$A \sim B$ The agent is indifferent between A and B

$A \succeq B$ The agent prefers A to B or is indifferent between them

- Deterministic lottery $[1,A] \equiv A$
- Reasonable constraints on the preference relation (in the name of rationality)
  - **Orderability**: given any two states, a rational agent must either prefer one to the other or else rate the two as equally preferable.

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

  - **Transitivity:**

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

- **Continuity**:

  $A \succ B \succ C \Rightarrow \exists\, p: [p, A; 1\text{-}p, C] \sim B$
- **Substitutability**:

  $A \sim B \Rightarrow [p, A; 1\text{-}p, C] \sim [p, B; 1\text{-}p, C]$
- **Monotonicity**:

  $A \succ B \Rightarrow (p \geq q \Leftrightarrow [p, A; 1\text{-}p, B] \succeq [q, A; 1\text{-}q, B])$
- **Decomposability**: Compound lotteries can be reduced to simpler ones by the laws of probability

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

- Notice that these axioms of utility theory do not say anything about utility
- The existence of a utility function follows from them

Preferences lead to utility

1. **Existence of Utility Function**:

   If an agent's preferences follow the axioms of utility, then there exists a real-valued function U s.t.

   $U(A) > U(B) \Leftrightarrow A \succ B$

   $U(A) = U(B) \Leftrightarrow A \sim B$

2. **Expected Utility of a Lottery**:

   The utility of a lottery is

   $U([p_1, S_1; \ldots; p_n, S_n]) = \sum_{i=1,\ldots,n} p_i \cdot U(S_i)$

   Because the outcome of a nondeterministic action is a lottery, this gives us the MEU decision rule from slide 254

- The axioms of utility do not specify a unique utility function for an agent
- For example, we can transform a utility function $U(S)$ into

$$U'(S) = aU(S) + b$$

where b is a constant and a is any positive constant
- Clearly, this affine transformation leaves the agent's behavior unchanged
- In deterministic contexts, where there are states but no lotteries, behavior is unchanged by any monotonic transformation
- E.g., the cube root of the utility $\sqrt[3]{(U(S))}$
- Utility function is ordinal — it really provides just rankings of states rather than meaningful numerical values

16.3 Utility Functions

- Money (or an agent's total net assets) would appear to be a straightforward utility measure
- The agent exhibits a monotonic preference for definite amounts of money
- We need to determine a model for lotteries involving money
    - We have won a million euros in a TV game show
    - The host offers to flip a coin, if the coin comes up heads, we end up with nothing, but if it comes up tails, we win three million euros
    - Is the only rational choice to accept the offer which has the expected monetary value of 1,5 million euros?

- The true question is maximizing total wealth (not winnings)

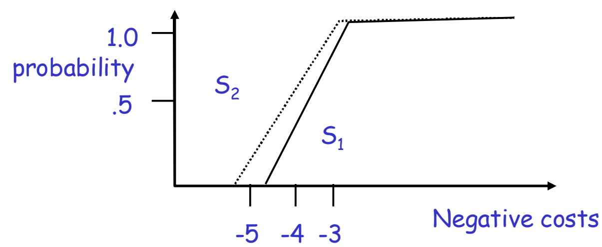
Normalized utilities

- The scale of utilities reaches from the best possible prize $u_\top$ to the worst possible catastrophe $u_\perp$
- *Normalized utilities* use a scale with $u_\perp = 0$ and $u_\top = 1$
- Utilities of intermediate outcomes are assessed by asking the agent to indicate a preference between the given outcome state S and a standard lottery $[p, u_\top; 1\text{-}p, u_\perp]$
- The probability p is adjusted until the agent is indifferent between S and the standard lottery
- Assuming normalized utilities, the utility of S is given by p

16.4 Multiattribute Utility Functions

- Most often the utility is determined by the values $\mathbf{x} = [x_1, \ldots, x_n]$ of multiple variables (attributes) $\mathbf{X} = X_1, \ldots, X_n$
- For simplicity, we will assume that each attribute is defined in such a way that, all other things being equal, higher values of the attribute correspond to higher utilities
- If for a pair of attribute vectors $\mathbf{x}$ and $\mathbf{y}$ it holds that $x_i \geq y_i \ \ \forall\ i$, then $\mathbf{x}$ *strictly dominates* $\mathbf{y}$
- Suppose that airport site S_1 costs less, generates less noise pollution, and is safer than site S_2, one would not hesitate to reject the latter
- In the general case, where the action outcomes are uncertain, strict dominance occurs less often than in the deterministic case
- *Stochastic dominance* is more useful generalization

- Suppose we believe that the cost of siting an airport is uniformly distributed between
  - S_1: 2.8 and 4.8 billion euros
  - S_2: 3.0 and 5.2 billion euros
- Then by examining the cumulative distributions, we see that S_1 stochastically dominates S_2 (because costs are negative)



- Cumulative distribution integrates the original distribution
- If two actions A_1 and A_2 lead to probability distributions $p_1(x)$ and $p_2(x)$ on attribute X
- A_1 stochastically dominates A_2 on X if

$$\forall\, x: \int_{-\infty,\ldots,x} p_1(x')\, dx' \leq \int_{-\infty,\ldots,x} p_2(x')\, dx'$$

- If
  - A_1 stochastically dominates A_2,
  - then for any monotonically nondecreasing utility function $U(x)$,

  the expected utility of A_1 is at least as high as that of A_2
- Hence, if an action is stochastically dominated by another action on all attributes, then it can be discarded

16.6 The Value of Information

- BP is hoping to buy one of n indistinguishable blocks of ocean drilling rights at the Gulf of Mexico
- Exactly one of the blocks contains oil worth C euros
- The price for each block is C/n euros
- A seismologist offers BP the results of a survey of block #3, which indicates definitively whether the block contains oil
- How much should BP be willing to pay for the information?
  - With probability $1/n$, the survey will indicate oil in block #3, in which case BP will buy the block for C/n euros and make a profit of $(n-1)C/n$ euros
  - With probability $(n-1)/n$, the survey will show that the block contains no oil, in which case BP will buy a different block

- Now the probability of finding oil in one of the other blocks changes to $1/(n-1)$, so BP makes an expected profit of $C/(n-1) - C/n = C/n(n-1)$ euros
- Now we can calculate the expected profit, given the survey information:

$$(1/n)\cdot((n-1)C/n) + ((n-1)/n)\cdot(C/n(n-1)) = C/n$$

- Therefore, BP should be willing to pay the seismologist up to the price of the block itself
- With the information, one's course of action can be changed to suit the actual situation
- Without the information, one has to do what's best on average over the possible situations