

Probability & Bayes Nets :

Conditional : $P(a | b) = P(a \wedge b) / P(b)$

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

Prove the conditionalized version of the general product rule:

$$P(A \wedge B | E) = P(A | B \wedge E) P(B | E)$$

$$P(A \wedge B | E) = P(A \wedge B \wedge E) / P(E)$$

$$P(A | B \wedge E) P(B | E) =$$

$$(P(A \wedge B \wedge E) / P(B \wedge E)) * (P(B \wedge E) / P(E)) =$$

$$P(A \wedge B \wedge E) / P(E)$$

Therefore $P(A \wedge B | E)$ is equivalent to $P(A | B \wedge E) P(B | E)$

Independence : $P(A \wedge B) = P(A) P(B)$

- $P(A) = P(A | B)$ and $P(B) = P(B | A)$

- $P(A \wedge B | C) = P(A | C) P(B | C)$

- A node is **conditionally independent** of its **non-descendants** given its **parents**

- A node is **conditionally independent** of all other nodes in the network given its parents, children, and children's parents also known as **Markov blanket**)

- The method called **d-separation** can be applied to decide whether a set of nodes X is independent of another set Y, given a third set Z

Bayes : $P(Y | X) = P(X | Y) P(Y) / P(X)$

Prove the conditionalized version of Bayes' rule:

$$P(A | B \wedge C) = P(B | A \wedge C) P(A | C) / P(B | C)$$

$$P(A \wedge B | C) = P(B | A \wedge C) P(A | C)$$

[equivalent to derivation in part 'a']

$$P(B | A \wedge C) P(A | C) = P(A | B \wedge C) P(B | C)$$

[restatement of above with product rule]

$$P(B | A \wedge C) P(A | C) / P(B | C) = P(A | B \wedge C)$$

[divide both sides by $P(B | C)$]

Using conditional independence, compute $P(\sim g, a | s)$ and $P(\sim g, a | \sim s)$. Then use Bayes' rule to compute $P(s | \sim g, a)$.

- $P(\sim g, a | s) = P(\sim g | s) P(a | s) = (0.2) * (0.75) = 0.15$

- $P(\sim g, a | \sim s) = P(\sim g | \sim s) P(a | \sim s) = (0.7) * (0.2) = 0.14$

- $P(s | \sim g, a) = P(\sim g, a | s) P(s) / (0.4) / 0.144 = .39$

Chaining : $P(a, b, c, d, e) =$

- $P(e | a, b, c, d) P(a, b, c, d) =$

- $P(e | c) P(a, b, c, d)$

Compute $P(a, \sim s, g)$ using chain rule:

- Reorder (order doesn't effect outcome) and apply chain rule: $P(\sim s) P(a | \sim s) P(g | a \wedge \sim s)$

- Since $P(G)$ is conditionally independent of $P(A)$:

$$P(\sim s) P(a | \sim s) P(g | \sim s)$$

$$P(\sim s) P(a | \sim s) P(g | \sim s) = 0.6 * 0.2 * 0.3 = 0.036$$

Chain rule comp. prob. $P(\sim g, a, s, c, \sim f)$:

- $P(\sim g, a, s, c, \sim f) = P(\sim g | s) P(a | s) P(s | c, \sim f)$

$$P(c) P(\sim f) =$$

$$.2 * .75 * .6 * .3 * .6 = 0.162$$

Inference By Enumeration:

Compute $P(a)$ using inference by enumeration.

$$P(a) = \sum_s (0.4 * 0.75) + \sum_{\sim s} (0.6 * 0.2) = (0.4 * 0.75) + (0.6 * 0.2) = 0.42$$

$P(a, \sim f)$ using inference by enumeration:

$$P(a, \sim f) = \sum_{C,S,G} (P(C) P(\sim f) P(S | C \wedge \sim f) P(a | S) P(G | S)) =$$

Knowledge Representation:

Semantic Networks: A semantic network is a simple representation scheme that uses a graph of labeled nodes and labeled, directed arcs to encode knowledge.

Deduction reasons from causes to effects

Abduction reasons from effects to causes

Induction reasons from specific cases to general rules

$$I(S) = \sum (\text{Prob of this event} * -1 * (P/N * \log(P/N) + D/N * \log(D/N)))$$

P = play, num of positive outcomes in this event, D = don't play, negative outcomes, N = total outcomes in this event

$$I(T,O) = 5/14 * -1 * (2/5 * \log(2/5) + 3/5 * \log(3/5))$$

Sunny: PPNNN

$$+ 4/14 * -1 * (4/4 * \log(4/4) + 0/4 * \log(0/4)) =$$

0 Overcast: PPPP

$$+ 5/14 * -1 * (3/5 * \log(3/5) + 2/5 * \log(2/5))$$

Rain: PPPNN

Info gain = Initial info - Info after

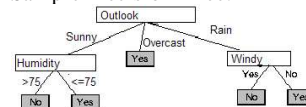
$$\text{Info gain} = I(T) - I(T,O)$$

$$\text{Split info} = -1 * ((N0/T) \log(N0/T) + (N1/T) \log(N1/T) + (N2/T) \log(N2/T))$$

$N0$, is number of outcomes in event 0, $N1$ is for event 1 and so on, T is total outcomes ($N0 + N1 + N2$)

Gain Ratio = info gain / split info

Sample Decision Tree:



General set: the most general set that includes all positive instances and none of the negative ones

Specific set: all positive instances

If you add a positive instance, general set remains unchanged, s set gets that instance

If you add a negative instance, the general set will become more specified, s set unchanged unless that instance is in the s set, then falls apart