1 handout: slides
Bayesian Networks
Figure 14.2 A typical Bayesian network, showing both the topology and the conditional probability tables (CPTs). In the CPTs, the letters $B$, $E$, $A$, $J$, and $M$ stand for Burglary, Earthquake, Alarm, JohnCalls, and MaryCalls, respectively.
Bayes Net = joint probability distribution
specifies independence:

\[ P(X_i|X_{i-1}, \ldots, X_1) = P(X_i|\text{parents}(X_i)) \]

joint:

\[ P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\text{parents}(X_i)) \]

What is distribution of \( X \) given evidence \( e \) and unobserved \( Y \)?
What is distribution of $X$ given evidence $e$ and unobserved $Y$?

$$P(X|e) = \frac{P(e|X)P(X)}{P(e)}$$

$$= \alpha P(X, e)$$

$$= \alpha \sum_y P(X, e, y)$$

$$= \alpha \sum_y \prod_{i=1}^{n} P(V_i|\text{parents}(V_i))$$
\[
P(B|j, m) = \alpha \sum_{e} \sum_{a} \prod_{i=1}^{n} P(V_i|\text{parents}(V_i))
\]
\[
P(b|j, m) = \alpha \sum_{e} \sum_{a} P(b)P(e)P(a|b, e)P(j|a)P(m|a)
\]
\[
= \alpha P(b) \sum_{e} P(e) \sum_{a} P(a|b, e)P(j|a)P(m|a)
\]
- Wed May 2: HMMs, ?
- Mon May 7: special guest Scott Kiesel on robot planning
- Wed May 9, 9-noon: project presentations
- Thur May 10, 8am: paper drafts (optional for some)
- Fri May 11, 10:30: exam 3 (N133)
- Tues May 15, 3pm: papers (one hardcopy + electronic PDF)
Particle Filters
Monte Carlo Localization

\[ S \leftarrow \text{samples from prior} \]
\[ w \leftarrow \text{uniform distribution} \]
repeat forever:
\[ \text{for each sample } s_i \text{ and weight } w_i, \]
\[ s_i \leftarrow \text{sample from } P(S_i'|s_i) \]
\[ w_i \leftarrow P(e|s_i) \]
\[ S \leftarrow \text{sample from } S \text{ with } P(s_i) \propto w_i \]
\[ +: \text{nonparametric, scalable computation and accuracy, simple} \]
\[ -: \text{high D, accurate sensors, kidnapping} \]
Hidden Markov Models
MDPs:
Naive Bayes:
k-Means:
Markov chain:
Hidden Markov model:
\[ P(x_t = j) = \sum_i P(x_{t-1} = i)P(x_t = j | x_{t-1} = i) \]

\[ P(e_t = k) = \sum_i P(x_t = i)P(e = k | x = i) \]
The Model

Bayesian Networks
Particle Filters
HMMs
Models
The Model
Viterbi Decoding

\[ P(x_t = j) = \sum_i P(x_{t-1} = i)P(x_t = j | x_{t-1} = i) \]

\[ P(e_t = k) = \sum_i P(x_t = i)P(e = k | x = i) \]

More concisely:

\[ P(x_t) = \sum_{x_{t-1}} P(x_{t-1})P(x_t | x_{t-1}) \]

\[ P(e_t) = \sum_{x_t} P(x_t)P(e | x) \]
Viterbi Decoding
probabilty of a sequence multiplies forward in time
dynamic programming backward through time
The Algorithm

given: transition model \( T(s, s') \)
sensing model \( S(s, o) \)
observations \( o_1, \ldots, o_T \)
find: most probable \( s_1, \ldots, s_T \)

initialize \( S \times T \) matrix \( v \) with 0s
\( v_{0,0} \leftarrow 1 \)
for each time \( t = 0 \) to \( T - 1 \)
    for each state \( s \)
        for each new state \( s' \)
            score \( \leftarrow v_{s,t} \cdot T(s, s') \cdot S(s', o_t) \)
            if score \( > v_{s',t+1} \)
                \( v_{s', t+1} \leftarrow \) score
                best-parent\( (s') \leftarrow s \)
trace back from \( s \) with max \( v_{s,T} \)
What question didn’t you get to ask today?
What’s still confusing?
What would you like to hear more about?

Please write down your most pressing question about AI and put it in the box on your way out.

Thanks!