Today

- Last time:
  - Bayes nets
  - Answering queries with variable elimination

- Today:
  - Reasoning over time
  - Markov processes
  - Hidden Markov models
Reasoning over Time

- Often, we want to reason about a sequence of observations
  - Speech recognition
  - Robot localization
  - User attention

- Need to introduce time into our models
- Basic approach: hidden Markov models (HMMs)
- More general: dynamic Bayes’ nets

Markov Models

- A Markov model is a chain-structured BN
  - Each node is identically distributed (stationarity)
  - Value of X at a given time is called the state
  - As a BN:

\[
X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow \ldots
\]

\[
P(X_1) \quad \quad P(X|X_{-1})
\]

- Parameters: called transition probabilities, specify how the state evolves over time (also, initial probs)
Conditional Independence

- **Basic conditional independence:**
  - Past and future independent of the present
  - Each time step only depends on the previous
  - This is called the (first order) Markov property

Example

- **Weather:**
  - States: \( X = \{ \text{rain, sun} \} \)
  - Transitions:

    - Initial distribution: 1.0 sun
    - What’s the probability distribution after one step?

    \[
    P(X_2 = \text{sun}) = P(X_2 = \text{sun}|X_1 = \text{sun})P(X_1 = \text{sun}) + P(X_2 = \text{sun}|X_1 = \text{rain})P(X_1 = \text{rain}) \\
    = 0.9 \cdot 1.0 + 0.1 \cdot 0.0 = 0.9
    \]
Mini-Forward Algorithm

- Question: probability of being in state x at time t?
- Slow answer:
  - Enumerate all sequences of length t which end in s
  - Add up their probabilities
- Better answer: cached incremental belief update

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

\[
P(x_1) = \text{known}
\]

Forward simulation

Example

- From initial observation of sun

\[
\begin{bmatrix}
1.0 & 0.9 & 0.82 & 0.5 \\
0.0 & 0.1 & 0.18 & 0.5 \\
\end{bmatrix}
\]

\[
P(X_1) \quad P(X_2) \quad P(X_3) \quad P(X_x)
\]

- From initial observation of rain

\[
\begin{bmatrix}
0.0 & 0.1 & 0.18 & 0.5 \\
1.0 & 0.9 & 0.82 & 0.5 \\
\end{bmatrix}
\]

\[
P(X_1) \quad P(X_2) \quad P(X_3) \quad P(X_x)
\]
Stationary Distributions

- If we simulate the chain long enough:
  - What happens?
  - Uncertainty accumulates
  - Eventually, we have no idea what the state is!

- Stationary distributions:
  - For most chains, the distribution we end up in is independent of the initial distribution
  - Called the stationary distribution of the chain
  - Usually, can only predict a short time out

Web Link Analysis

- PageRank over a web graph
  - Each web page is a state
  - Initial distribution: uniform over pages
  - Transitions:
    - With prob. c, uniform jump to a random page (solid lines)
    - With prob. 1-c, follow a random outlink (dotted lines)

- Stationary distribution
  - Will spend more time on highly reachable pages
  - E.g. many ways to get to the Acrobat Reader download page!
  - Somewhat robust to link spam
  - Google 1.0 returned pages containing your keywords in decreasing rank, now all search engines use link analysis along with many other factors
Most Likely Explanation

- Question: most likely sequence ending in x at t?
  - E.g. if sun on day 4, what's the most likely sequence?
  - Intuitively: probably sun all four days
- Slow answer: enumerate and score

\[
P(X_1 = \text{sun})P(X_2 = \text{sun} | X_1 = \text{sun})P(X_3 = \text{sun} | X_2 = \text{sun})P(X_4 = \text{sun} | X_3 = \text{sun})
\]

\[
P(X_1 = \text{sun})P(X_2 = \text{rain} | X_1 = \text{sun})P(X_3 = \text{sun} | X_2 = \text{rain})P(X_4 = \text{sun} | X_3 = \text{sun})
\]

\[\vdots\]

Mini-Viterbi Algorithm

- Better answer: cached incremental updates

- Define: \( m_t[x] = \max_{x_{1:t-1}} P(x_{1:t-1}, x) \)
  \( a_t[x] = \arg \max_{x_{1:t-1}} P(x_{1:t-1}, x) \)
- Read best sequence off of m and a vectors
Mini-Viterbi

\[ m_t[x] = \max_{x_{1:t-1}} P(x_{1:t-1}, x) \]
\[ = \max_{x_{1:t-1}} P(x_{1:t-1}) P(x|x_{t-1}) \]
\[ = \max_{x_{t-1}} P(x_t|x_{t-1}) \max_{x_{1:t-2}} P(x_{1:t-1}) \]
\[ = \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x] \]
\[ m_1[x] = P(x_1) \]

Hidden Markov Models

- Markov chains not so useful for most agents
  - Eventually you don’t know anything anymore
  - Need observations to update your beliefs

- Hidden Markov models (HMMs)
  - Underlying Markov chain over states S
  - You observe outputs (effects) at each time step
  - As a Bayes’ net:
Example

- An HMM is
  - Initial distribution: \( P(X_1) \)
  - Transitions: \( P(X|X_{-1}) \)
  - Emissions: \( P(E|X) \)

Conditional Independence

- HMMs have two important independence properties:
  - Markov hidden process, future depends on past via the present
  - Current observation independent of all else given current state

- Quiz: does this mean that observations are independent given no evidence? Why? [No, correlated by the hidden state]
Forward Algorithm

- Can ask the same questions for HMMs as Markov chains
- Given current belief state, how to update with evidence?
  - This is called monitoring or filtering
- Formally, we want: $P(X_t = x_t | e_{1:t})$

$$P(x_t | e_{1:t}) \propto P(x_t, e_{1:t})$$

$$= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t})$$

$$= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) P(x_t | x_{t-1}) P(e_t | x_t)$$

$$= P(e_t | x_t) \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1}, e_{1:t-1})$$

Example

$$P(x_t | e_{1:t}) \propto f_t[x_t] = P(x_t, e_{1:t})$$

$$f_t[x_t] = P(e_t | x_t) \sum_{x_{t-1}} P(x_t | x_{t-1}) f_{t-1}[x_{t-1}]$$
Viterbi Algorithm

- Question: what is the most likely state sequence given the observations?
  - Slow answer: enumerate all possibilities
  - Better answer: cached incremental version

\[
x^*_1:T = \arg \max_{x_1:T} P(x_1:T | e_1:T)
\]

\[
m_t[x_t] = \max_{x_{1:t-1}} P(x_1:t-1, x_t, e_{1:t})
\]

\[
= \max_{x_{1:t-1}} P(x_1:t-1, e_{1:t-1}) P(x_t|x_{t-1}) P(e_t|x_t)
\]

\[
= P(e_t|x_t) \max_{x_{1:t-1}} P(x_t|x_{t-1}) \max_{x_{1:t-2}} P(x_1:t-1, e_{1:t-1})
\]

\[
= P(e_t|x_t) \max_{x_{1:t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]
\]

Example

![Example Diagram]