Recap: Value of Information

- Current evidence $E=e$, utility depends on $S=s$
  \[
  \text{MEU}(e) = \max_{a} \sum_{s} P(s|e) \cdot U(s, a)
  \]
- Potential new evidence $E'$: suppose we knew $E' = e'$
  \[
  \text{MEU}(e', e') = \max_{a} \sum_{s} P(s|e, e') \cdot U(s, a)
  \]
- BUT $E'$ is a random variable whose value is currently unknown, so:
  - Must compute expected gain over all possible values
  \[
  \text{VPI}(E') = \sum_{e'} P(e'|e) \left( \text{MEU}(e', e') - \text{MEU}(e) \right)
  \]
- (VPI = value of perfect information)

VPI Scenarios

- Imagine actions 1 and 2, for which $U_1 > U_2$
- How much will information about $E_j$ be worth?

  - Little – we’re sure action 1 is better.
  - A lot – either could be much better
  - Little – info likely to change our action but not our utility

Reasoning over Time

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

- 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:
    \[
    X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow \cdots
    \]
    \[
    P(X_1) \cdot P(X_i | X_{i-1})
    \]
  - Parameters: called transition probabilities or dynamics, specify how the state evolves over time (also, initial probs)

Announcements

- Optional midterm
  - On Tuesday 11/21 in class
  - We will count the midterms and final as 1 / 1 / 2, and drop the lowest (or halve the final weight)
  - Will also run grades as if this midterm did not happen
  - You will get the better of the two grades

- Projects
  - 3.2 up, do written questions before programming
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

- Note that the chain is just a (growing BN)
  - We can always use generic BN reasoning on it (if we truncate the chain)

Example: Markov Chain

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

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

  \[
  = 0.9 \cdot 1.0 + 0.1 \cdot 0.0 = 0.9
  \]

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

\[
P(X_1 = \text{sun}) = \sum_{x_1=\text{sun}, \ldots, x_{t-1}} P(x_1, \ldots, x_{t-1}, \text{sun})
\]

Mini-Forward Algorithm

- Better way: cached incremental belief updates

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

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 (dotted lines)
    - With prob. 1-c, follow a random outlink (solid 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 the set of pages containing all 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
  
  \[
P(X_t = \text{sun}) = \max_{x_{1:t-1}} P(x_1, \ldots, x_{t-1}, \text{sun})
  \]

  \[
P(X_t = \text{sun}) P(X_{t-1} = \text{sun}) P(X_{t-2} = \text{sun}) P(X_{t-3} = \text{sun}) P(X_{t-4} = \text{sun})
  \]

- **Better answer**: cached incremental updates

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

  \[
  m_1[x] = P(x_1)
  \]

  \[
  a_t[x] = \arg \max_{x_{1:t-1}} P(x_1, x_{t-1}, x)
  \]

### Mini-Viterbi Algorithm

- **Better answer**: cached incremental updates

  \[
  \begin{align*}
  s & \quad s' & \quad s(x|s') \\
  \text{sun} & \quad \text{sun} & \quad 3/5 \\
  \text{sun} & \quad \text{rain} & \quad 2/3 \\
  \text{rain} & \quad \text{sun} & \quad 4/5 \\
  \text{rain} & \quad \text{rain} & \quad 1/5 \\
  \text{sun} & \quad \text{sun} & \quad 1/3 \\
  \text{sun} & \quad \text{rain} & \quad 2/3 \\
  \end{align*}
  \]

### Example

- **Mini-Viterbi**

- **Read best sequence off of m and a vectors**

- **Example**
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:

  $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow \ldots$

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

  $X \rightarrow E_1 \rightarrow E_2 \rightarrow E_3 \rightarrow E_4$

- Quiz: does this mean that observations are independent given no evidence?
  - [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: $P(X_t = x_t | e_{1:t})$

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

\[ x_{1:T}^* = \arg \max_{x_{1:T}} P(x_{1:T}|e_{1:T}) \]

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

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

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

**Real HMM Examples**

- **Speech recognition HMMs:**
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)

- **Machine translation HMMs:**
  - Observations are words (tens of thousands)
  - States are translation positions (dozens)

- **Robot tracking:**
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)