Hidden Markov Models

George Konidaris
gdk@cs.brown.edu

Fall 2019
Recall: Bayesian Network

- Flu
- Allergy
- Nose
- Headache
Recall: BN

<table>
<thead>
<tr>
<th>Flu</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>0.6</td>
</tr>
<tr>
<td>False</td>
<td>0.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Allergy</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>0.2</td>
</tr>
<tr>
<td>False</td>
<td>0.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sinus</th>
<th>Flu</th>
<th>Allergy</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>0.9</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>0.1</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>0.6</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>0.4</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>0.2</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>False</td>
<td>0.8</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>0.4</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>True</td>
<td>0.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nose</th>
<th>Sinus</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>0.8</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>0.2</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>0.3</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>0.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Headache</th>
<th>Sinus</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>0.6</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>0.4</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>0.5</td>
</tr>
<tr>
<td>False</td>
<td>False</td>
<td>0.5</td>
</tr>
</tbody>
</table>

joint: 32 (31) entries
Inference

Given A compute $P(B \mid A)$. 
Bayesian Networks (so far) contain no notion of \textit{time}.

However, in many applications:
\begin{itemize}
  \item Target tracking
  \item Patient monitoring
  \item Speech recognition
  \item Gesture recognition
\end{itemize}

\ldots how a signal changes over time is critical.
States

In probability theory, we talked about atomic events:
  • All possible outcomes.
  • Mutually exclusive.

In time series, we have state:
  • System is in a state at time t.
  • Describes system completely.
  • Over time, transition from state to state.
Example

The weather today can be:

- Hot
- Cold
- Chilly
- Freezing

The weather has four states.

At each point in time, the system is in one (and only one) state.
Example

State at time $t$

Freezing
Chilly
Hot

Freezing
Chilly
Hot

Freezing
Chilly
Hot

Freezing
Chilly
Hot

State transition
The Markov Assumption

We are probabilistic modelers, so we’d like to model:

$$P(S_t | S_{t-1}, S_{t-2}, \ldots, S_0)$$

A state has the Markov property when we can write this as:

$$P(S_t | S_{t-1})$$

Special kind of independence assumption:

- Future independent of past given present.
Markov Assumption

Model that has it is a **Markov model**.

Sequence of states thus generated is a **Markov chain**.

**Definition of a state:**
- Sufficient statistic for history
- $P(S_t|S_{t-1}, \ldots, S_0) = P(S_t|S_{t-1})$

Can describe transition probabilities with matrix:
- $P(S_i | S_j)$
- Steady state probabilities.
- Convergence rates.
State Machines

\[ P(A \mid B) = 0.8 \]
\[ P(A \mid C) = 0.5 \]
\[ P(B \mid A) = 0.4 \]
\[ P(B \mid C) = 0.5 \]
\[ P(C \mid A) = 0.6 \]
\[ P(C \mid B) = 0.2 \]

Time implicit

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.0</td>
<td>0.8</td>
<td>0.5</td>
</tr>
<tr>
<td>B</td>
<td>0.4</td>
<td>0.0</td>
<td>0.5</td>
</tr>
<tr>
<td>C</td>
<td>0.6</td>
<td>0.2</td>
<td>0.0</td>
</tr>
</tbody>
</table>

states not state vars!
State Machines

Assumptions:

• Markov assumption.
• Transition probabilities don’t change with time.
• Event space doesn’t change with time.
• Time moves in discrete increments.
Hidden State

State machines are cool but:
- Often state is not observed directly.
- State is latent, or hidden.

Instead you see an observation, which contains information about the hidden state.

State:
forehand
Examples

State

Word

Chemical State

Flu?

Cardiac Arrest?

Observation

Phoneme

Color, Smell, etc.

Runny Nose

Pulse

Sensor
Hidden Markov Models

Must store:
- $P(O | S)$
- $P(S_{t+1} | S_t)$
HMMs

Monitoring/Filtering
• $P(S_t \mid O_0 \ldots O_t)$
• E.g., estimate patient disease state.

Prediction
• $P(S_t \mid O_0 \ldots O_k), k < t$.
• Given first two phonemes, what word?

Smoothing
• $P(S_t \mid O_0 \ldots O_k), k > t$
• What happened back there?

Most Likely Path
• $P(S_0 \ldots S_t \mid O_0 \ldots O_t)$
• How did I get here?
Example: Robot Localization

observations: walls each side?

states: position
Example: Robot Localization

We start off not knowing where the robot is.
Example: Robot Localization

Robot sense: obstacles up and down. Updates distribution.
Example: Robot Localization

Robot moves right: updates distribution.
Example: Robot Localization

Obstacles up and down, updates distribution.
What Happened

This is an instance of robot tracking - *filtering*.

Could also:

- Predict (where will the robot be in 3 steps?)
- Smooth (where was the robot?)
- Most likely path (what was the robot’s path?)

All of these are questions about the HMM’s state at various times.
How?

Let's look at $P(S_t) - no observations.$ Assime we have CPTs
\begin{align*}
P(S_1 = a) &= P(S_0 = a)P(a \mid a) + P(S_0 = b)P(a \mid b) \\
P(S_1 = b) &= P(S_0 = a)P(b \mid a) + P(S_0 = b)P(b \mid b)
\end{align*}
Prediction

\[ P(S_2 = a) = P(S_1 = a)P(a | a) + P(S_1 = b)P(a | b) \]

\[ P(S_2 = b) = P(S_1 = a)P(b | a) + P(S_1 = b)P(b | b) \]
Filtering

\[ \text{Max } P(S_t | O_0 \ldots O_t). \]
Filtering

Where to start?

\( P(S_t \mid O_0 \ldots O_t) \)? Let’s use \( P(S_t, O_0 \ldots O_t) \).

\[
P(S_t, O_0, ..., O_t) = \sum_i P(S_t, S_{t-1} = s_i, O_0, ..., O_t)
\]

\[
= \sum_i P(O_t \mid S_t) P(S_t \mid S_{t-1} = s_i) P(S_{t-1} = s_i, O_0, ..., O_{t-1})
\]

\[
= P(O_t \mid S_t) \sum_i P(S_t \mid S_{t-1} = s_i) P(S_{t-1} = s_i, O_0, ..., O_{t-1})
\]
Forward Algorithm

Let $F(k, 0) = P(S_0 = s_k)P(O_0 | S_0 = s_k)$.

For $t = 1, \ldots, T$:

For $k$ in possible states:

$$F(k, t) = P(O_t | S_t = s_k) \sum_{i} P(s_k | s_i)F(i, t - 1)$$

$F(k, T)$ is $P(S_T = s_k, O_0 \ldots O_T)$

(normalize to get $P(S_T | O_0 \ldots O_T)$)
Smoothing

\[ P(S_t \mid O_0 \ldots O_k), \ k > t \ \text{ - given data of length } k, \ \text{find } P(S_t) \ \text{for earlier } t. \]

Bayes Rule:

- \[ P(S_t \mid O_0 \ldots O_k) \propto P(O_0 \ldots O_k \mid S_t) \ P(S_t \mid O_0 \ldots O_k) \]
- \[ P(O_t \ldots O_k \mid S_t) \ P(S_t \mid O_0 \ldots O_t) \]

forward algorithm

Compute using backward pass:
\[ P(O_i \ldots O_k \mid S_i) \ \text{computed using similar recursion.} \]

**Forward-backward algorithm.**
Most Likely Path

\[ \max P(S_0 \ldots S_t \mid O_0 \ldots O_t) \]

\[ S_0 \ldots S_t \]
Viterbi

Similar logic to highest probability state, but:

- We seek a *path*, not a *state*.
- *Single highest probability state*.
- Therefore look for highest probability of *(ancestor probability times observation probability)*
- Maintain link matrix to read path backwards

Similar dynamic programming algorithm, replace *sum* with *max*. 
Viterbi Algorithm

Most likely path $S_0 \ldots S_n$:

$V_{i,k}$: probability of max prob. path at ending in state $s_k$, including observations up to $O_i$ ($t=i$).

$L_{i,k}$: most likely predecessor of state $s_k$ at time $i$.

For each state $s_k$:

$V_{0,k} = P(O_0 | s_k)P(s_k)$

$L_{0,k} = 0$

For $i = 1 \ldots n$,

For each $k$:

$V_{i,k} = P(O_i | s_k) \max_x P(s_k | s_x) V_{i-1,x}$

$L_{i,k} = \arg\max_x P(s_k | s_x)V_{i-1,x}$
Common Form

Very common form:
- Noisy observations of true state
Viterbi

“The algorithm has found universal application in decoding the convolutional codes used in both CDMA and GSM digital cellular, dial-up modems, satellite, deep-space communications, and 802.11 wireless LANs.” (wikipedia)

(photo credit: MIT)