Hidden Markov Models

and the Forward-Backward Algorithm
Please See the Spreadsheet

- I like to teach this material using an interactive spreadsheet:
  - [http://cs.jhu.edu/~jason/papers/#tnlp02](http://cs.jhu.edu/~jason/papers/#tnlp02)
  - Has the spreadsheet and the lesson plan

- I’ll also show the following slides at appropriate points.
## Marginalization

<table>
<thead>
<tr>
<th>SALES</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Widgets</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>...</td>
</tr>
<tr>
<td>Grommets</td>
<td>7</td>
<td>3</td>
<td>10</td>
<td>8</td>
<td>...</td>
</tr>
<tr>
<td>Gadgets</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
# Marginalization

Write the totals in the margins

<table>
<thead>
<tr>
<th>SALES</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>...</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Widgets</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>Grommets</td>
<td>7</td>
<td>3</td>
<td>10</td>
<td>8</td>
<td></td>
<td>80</td>
</tr>
<tr>
<td>Gadgets</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>TOTAL</td>
<td>99</td>
<td>25</td>
<td>126</td>
<td>90</td>
<td></td>
<td>1000</td>
</tr>
</tbody>
</table>
## Marginalization

Given a random sale, what & when was it?

<table>
<thead>
<tr>
<th>prob.</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>...</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Widgets</td>
<td>.005</td>
<td>0</td>
<td>.003</td>
<td>.002</td>
<td></td>
<td>.030</td>
</tr>
<tr>
<td>Grommets</td>
<td>.007</td>
<td>.003</td>
<td>.010</td>
<td>.008</td>
<td></td>
<td>.080</td>
</tr>
<tr>
<td>Gadgets</td>
<td>0</td>
<td>0</td>
<td>.001</td>
<td>0</td>
<td></td>
<td>.002</td>
</tr>
<tr>
<td>TOTAL</td>
<td>.099</td>
<td>.025</td>
<td>.126</td>
<td>.090</td>
<td></td>
<td>1.000</td>
</tr>
</tbody>
</table>
Given a random sale, what & when was it?

Marginalization

<table>
<thead>
<tr>
<th>prob</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>...</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Widgets</td>
<td>.005</td>
<td>0</td>
<td>.003</td>
<td>.002</td>
<td>...</td>
<td>.030</td>
</tr>
<tr>
<td>Grommets</td>
<td>.007</td>
<td>.003</td>
<td>.010</td>
<td>.008</td>
<td>...</td>
<td>.080</td>
</tr>
<tr>
<td>Gadgets</td>
<td>0</td>
<td>0</td>
<td>.001</td>
<td>0</td>
<td>...</td>
<td>.002</td>
</tr>
<tr>
<td>TOTAL</td>
<td>.099</td>
<td>.025</td>
<td>.126</td>
<td>.090</td>
<td></td>
<td>1.000</td>
</tr>
</tbody>
</table>

- **Joint prob:** \( p(\text{Jan}, \text{widget}) \)
- **Marginal prob:** \( p(\text{Jan}) \)
- **Marginal prob:** \( p(\text{widget}) \)
- **Marginal prob:** \( p(\text{anything in table}) \)
Given a random sale in Jan., what was it?

**Conditionalization**

<table>
<thead>
<tr>
<th>prob.</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>...</th>
<th>TOTAL</th>
<th>p(…</th>
<th>Jan)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Widgets</td>
<td>.005</td>
<td>0</td>
<td>.003</td>
<td>.002</td>
<td>...</td>
<td>.030</td>
<td>.005/.099</td>
<td></td>
</tr>
<tr>
<td>Grommets</td>
<td>.007</td>
<td>.003</td>
<td>.010</td>
<td>.008</td>
<td>...</td>
<td>.080</td>
<td>.007/.099</td>
<td></td>
</tr>
<tr>
<td>Gadgets</td>
<td>0</td>
<td>0</td>
<td>.001</td>
<td>0</td>
<td>...</td>
<td>.002</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>.099</td>
<td>.025</td>
<td>.126</td>
<td>.090</td>
<td></td>
<td>1.000</td>
<td>.099/.099</td>
<td></td>
</tr>
</tbody>
</table>

marginal prob: p(Jan)

conditional prob: p(widget|Jan)=.005/.099

Divide column through by $Z=0.99$ so it sums to 1
Marginalization & conditionalization in the weather example

- Instead of a 2-dimensional table, now we have a 66-dimensional table:
  - 33 of the dimensions have 2 choices: \{C,H\}
  - 33 of the dimensions have 3 choices: \{1,2,3\}
- Cross-section showing just 3 of the dimensions:

<table>
<thead>
<tr>
<th>Weather_3=C</th>
<th>Weather_2=C</th>
<th>Weather_2=H</th>
</tr>
</thead>
<tbody>
<tr>
<td>IceCream_2=1</td>
<td>0.000...</td>
<td>0.000...</td>
</tr>
</tbody>
</table>
Interesting probabilities in the weather example

- **Prior** probability of weather: 
  \[ p(\text{Weather}=\text{CHH}...) \]

- **Posterior** probability of weather (after observing evidence): 
  \[ p(\text{Weather}=\text{CHH}... \mid \text{IceCream}=233...) \]

- **Posterior marginal** probability that day 3 is hot: 
  \[ p(\text{Weather}_3=\text{H} \mid \text{IceCream}=233...) \]
  \[ = \sum_w \text{such that } w_3=\text{H} \ p(\text{Weather}=w \mid \text{IceCream}=233...) \]

- **Posterior conditional** probability that day 3 is hot if day 2 is: 
  \[ p(\text{Weather}_3=\text{H} \mid \text{Weather}_2=\text{H}, \text{IceCream}=233...) \]
The HMM trellis

The dynamic programming computation of $\alpha$ works forward from Start.

Day 1: 2 cones

- $\alpha = 0.1 \times 0.7 = 0.07$
- $p(C|H) \times p(3|H) = 0.8 \times 0.7 = 0.56$

Day 2: 3 cones

- $\alpha = 0.1 \times 0.08 + 0.1 \times 0.01 = 0.009$
- $p(C|C) \times p(3|C) = 0.8 \times 0.1 = 0.08$

Day 3: 3 cones

- $\alpha = 0.009 \times 0.08 + 0.009 \times 0.01 = 0.0063$
- $p(C|C) \times p(3|C) = 0.8 \times 0.1 = 0.08$

This “trellis” graph has $2^{33}$ paths.
- These represent all possible weather sequences that could explain the observed ice cream sequence 2, 3, 3, ...

What is the product of all the edge weights on one path $H, H, H, ...$?
- Edge weights are chosen to get $p(\text{weather}=H,H,H,... \& \text{icecream}=2,3,3,...)$

What is the $\alpha$ probability at each state?
- It’s the total probability of all paths from Start to that state.
- How can we compute it fast when there are many paths?
Computing $\alpha$ Values

All paths to state:

$$\alpha = (ap_1 + bp_1 + cp_1) + (dp_2 + ep_2 + fp_2)$$

$$= \alpha_1 \cdot p_1 + \alpha_2 \cdot p_2$$

Thanks, distributive law!
What is the $\beta$ probability at each state?
- It's the total probability of all paths from that state to Stop
- How can we compute it fast when there are many paths?
Computing $\beta$ Values

All paths from state:

$$\beta = (p_1 u + p_1 v + p_1 w) + (p_2 x + p_2 y + p_2 z)$$

$$= p_1 \cdot \beta_1 + p_2 \cdot \beta_2$$
Computing State Probabilities

All paths through state:

\[ ax + ay + az + bx + by + bz + cx + cy + cz \]

\[ = (a+b+c) \cdot (x+y+z) \]

\[ = \alpha(C) \cdot \beta(C) \]

Thanks, distributive law!
Computing Arc Probabilities

All paths through the p arc:

\[ apx + apy + apz + bpx + bpy + bpz + cpx + cpy + cpz \]

\[ = (a+b+c)p(x+y+z) \]

\[ = \alpha(H) \cdot p \cdot \beta(C) \]

Thanks, distributive law!
Posterior tagging

- Give each word its highest-prob tag according to forward-backward.
  - Do this independently of other words.
  - **Det Adj** 0.35 \(\Rightarrow\) exp # correct tags = 0.55 + 0.35 = 0.9
  - **Det N** 0.2 \(\Rightarrow\) exp # correct tags = 0.55 + 0.2 = 0.75
  - **N V** 0.45 \(\Rightarrow\) exp # correct tags = 0.45 + 0.45 = 0.9
- Output is
  - **Det V** 0 \(\Rightarrow\) exp # correct tags = 0.55 + 0.45 = 1.0
- Defensible: maximizes expected # of correct tags.
- But not a coherent sequence. May screw up subsequent processing (e.g., can’t find any parse).
Alternative: Viterbi tagging

- **Posterior tagging:** Give each word its highest-prob tag according to forward-backward.
  - Det Adj 0.35
  - Det N 0.2
  - N V 0.45

- **Viterbi tagging:** Pick the single best tag sequence (best path):
  - N V 0.45

- Same algorithm as forward-backward, but uses a semiring that maximizes over paths instead of summing over paths.
Max-product instead of sum-product

Use a semiring that maximizes over paths instead of summing.

We write $\mu, \nu$ instead of $\alpha, \beta$ for these “Viterbi forward” and “Viterbi backward” probabilities."

The dynamic programming computation of $\mu$. ($\nu$ is similar but works back from Stop.)

Day 1: 2 cones

\[
\begin{align*}
\mu &= \max(0.1 \times 0.07, 0.1 \times 0.56) \\
&= 0.056
\end{align*}
\]

Day 2: 3 cones

\[
\begin{align*}
\mu &= \max(0.008 \times 0.08, 0.008 \times 0.01) \\
&= 0.00064
\end{align*}
\]

Day 3: 3 cones

\[
\begin{align*}
\mu &= \max(0.008 \times 0.07, 0.056 \times 0.56) \\
&= 0.03136
\end{align*}
\]

$\alpha \times \beta$ at a state = total prob of all paths through that state

$\mu \times \nu$ at a state = max prob of any path through that state

Suppose max prob path has prob $p$: how to print it?

- Print the state at each time step with highest $\mu \times \nu$ ($= p$); works if no ties
- Or, compute $\mu$ values from left to right to find $p$, then follow backpointers