Classification Outline

• Classification Problem Overview
• Algorithm performance measures
• Classification Techniques
  – Decision Trees
    • ID3, MDL Pruning, CART, Gini
  – Naïve Bayes
  – Hidden Markov Models
• Basics in first half of the course (next two weeks), then advances after spring break.
Some Notes on Classification

• Many of the algorithms are covered in the ML course (CSI635) where we’ll analyze their behavior much more formally.

• Data mining in classification involves taking existing ML algorithms and adapting them to real-world difficulties.
## Data Set

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Classification Problem

• Given a database $D=\{t_1, t_2, \ldots, t_n\}$ and a set of classes $C=\{C_1, \ldots, C_m\}$, the Classification Problem is to define a mapping $f:D \rightarrow C$ where each $t_i$ is assigned to one class.

• Actually divides $D$ into equivalence classes.

• Our aim is to find the function $f$ that minimizes generalization error. Formalize …

• Different algorithms differ in what patterns they find to address the above problem.
Classification Examples

• Is a session/transaction malignant or benign
• Identify individuals with credit risks.
• Allocate a web session as being a buyer or browser
• Is a group of transactions fraudulent
• Challenges with classification mining … others …
  – Can’t store entire database in memory, mine a sample, but how big should the sample be?
  – Streaming data arrives so quickly, can’t store and process it.
  – Imbalanced phenomenon of interest.
  – Stationary distribution assumption violated
    • Past is not exactly like future
    • Environment is not benign changes patterns to mislead classifier
Classification Performance

True Positive

False Positive

False Negative

True Negative

Tall Classified Tall: 20
Tall Classified Not Tall: 10
Not Tall Classified Tall: 45
Not Tall Classified Not Tall: 25
Confusion Matrix Example

Using height data example with Output1 correct and Output2 actual assignment

<table>
<thead>
<tr>
<th>Actual Membership</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Short</td>
</tr>
<tr>
<td>Short</td>
<td>0</td>
</tr>
<tr>
<td>Medium</td>
<td>0</td>
</tr>
<tr>
<td>Tall</td>
<td>0</td>
</tr>
</tbody>
</table>
Classification

Functional Overview – Learning Stage

Examples

- + + - + -

Age, … income, \{Default | NoDefault\}

Case A. 30, …, $110K, Default
Case B. 50, …, $110K, NoDefault
Case C. 45, …, $90K, NoDefault
Case A. 32, …, $105K, Default
Case B. 49, …, $82K, NoDefault
Case C. 29, …, $50K, NoDefault
Classification
Functional Overview – Application Stage

Unlabeled Examples

Age, … income, \{**Default** | **NoDefault**\}
Case zx. 29, …, $113K, ?
Case zy. 42, …, $81K, ?
Case zz. 41, …, $92K, ?

Predictions

zx, Default
zy. NoDefault
zz. NoDefault
Classification Terminology

- Attribute, instances
- Training set, test set
- Training set accuracy
- Test set accuracy
- X-fold validation
- Confusion Matrix
- False positive
- False negative
- Cost sensitive classification
- Independent and dependent attributes
Classification Module

• Before Spring Break
  – Covering basic algorithms
    • Decision Trees, Naïve Bayes
  – Some limitations
    • Multiple passes of the database
    • Hold entire database in memory
    • Can not predict rare events well (an example)
    • Uses excessive data*

• After Spring Break
  – Advances to handle these and other limitations
## Classification Problem

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Decision Tree For Playing Tennis

ROOT NODE

BRANCH

INTERNAL NODE

LEAF NODE

Disjunction of conjunctions
Mutually exclusive and exhaustive/complete
Decision Tree

Given:

– $D = \{t_1, \ldots, t_n\}$ where $t_i = <t_{i1}, \ldots, t_{im}>$
– Database schema contains $\{A_1, A_2, \ldots, A_h\}$
– Classes $C=\{C_1, \ldots, C_l\}$

*Decision or Classification Tree* is a tree associated with $D$ such that

– Each internal node is labeled with attribute, $A_i$
– Each arc is labeled with predicate which can be applied to attribute at parent
– Each leaf node is labeled with a class, $C_j$
Top-Down Tree Induction

Main loop:
1. $A \leftarrow$ the “best” decision attribute for next node
2. Assign $A$ as decision attribute for node
3. For each value of $A$, create new descendant of node
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?

```
[29+, 35-]  A1=?  [29+, 35-]  A2=?
   t     t
[21+, 5-]  [8+, 30-]  [18+, 33-]  [11+, 2-]
   f     f
```
Which Column and Split Point?

- Multitude of techniques:
  - Entropy/Information gain
  - Chi square test (CHAID)
    - Test of independence
  - GINI index
Information Gain

\[ Gain(S, A) = \text{expected reduction in entropy due to sorting on } A \]

\[ Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \]
Entropy

$Entropy(S) = $ expected number of bits needed to encode class ($\oplus$ or $\ominus$) of randomly drawn member of $S$ (under the optimal, shortest-length code)

Why?

Information theory: optimal length code assigns $-\log_2 p$ bits to message having probability $p$.

So, expected number of bits to encode $\oplus$ or $\ominus$ of random member of $S$:

$$p_\oplus(-\log_2 p_\oplus) + p_\ominus(-\log_2 p_\ominus)$$

$Entropy(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$
Information/Entropy

• Given probabilities $p_1, p_2, \ldots, p_s$ whose sum is 1, *Entropy* is defined as:

$$H(p_1, p_2, \ldots, p_s) = \sum_{i=1}^{s} (p_i \log(1/p_i))$$

• Entropy measures the amount of randomness or surprise or uncertainty.

• Goal in classification
  – no surprise
  – entropy = 0
Entropy

\[ \log \left( \frac{1}{p} \right) \]

\[ H(p,1-p) \]
Data Set

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Choosing the Next Attribute - 1

Which attribute is the best classifier?

\[ S: [9+, 5-] \quad E = 0.940 \]

- **Humidity**
  - **High**
    - [3+, 4-]
      - \( E = 0.985 \)
  - **Normal**
    - [6+, 1-]
      - \( E = 0.592 \)

\[ \text{Gain} (S, \text{Humidity}) = 0.940 - (7/14) \cdot 0.985 - (7/14) \cdot 0.592 = 0.151 \]

\[ S: [9+, 5-] \quad E = 0.940 \]

- **Wind**
  - **Weak**
    - [6+, 2-]
      - \( E = 0.811 \)
  - **Strong**
    - [3+, 3-]
      - \( E = 1.00 \)

\[ \text{Gain} (S, \text{Wind}) = 0.940 - (8/14) \cdot 0.811 - (6/14) \cdot 1.0 = 0.048 \]
Choosing the Next Attribute - 2

Which attribute should be tested here?

\[ S_{\text{Sunny}} = \{ D_1, D_2, D_8, D_9, D_{11} \} \]

\[ \text{Gain}(S_{\text{Sunny}}, \text{Humidity}) = .970 - (3/5)0.0 - (2/5)0.0 = .970 \]

\[ \text{Gain}(S_{\text{Sunny}}, \text{Temperature}) = .970 - (2/5)0.0 - (2/5)1.0 - (1/5)0.0 = .570 \]

\[ \text{Gain}(S_{\text{Sunny}}, \text{Wind}) = .970 - (2/5)1.0 - (3/5).918 = .019 \]
### Information Gain For Rare Events

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
• Entropy at root of tree 0.111 (base 2)
• Split on Humidity gives entropy of 0.09
• IG(Humidity) = 0.0211
• Then split on Wind gives entropy of 0.12
• IG(Windy) = -0.03
• With rare events algorithms build small or no model.
• Why?
DT Issues

• Choosing Splitting Attributes
• Continuous attributes
• Splits
• Tree Structure
• Stopping Criteria
• Pruning
Confidences

outlook = sunny
| humidity = high: no (3.0)
| humidity = normal: yes (2.0)

outlook = overcast: yes (4.0)

outlook = rainy
| windy = TRUE: no (2.0)
| windy = FALSE: yes (3.0)

Number of Leaves : 5

Size of the tree : 8

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

How many possible combos
Test on ...
D15, Sunny, Mild, Low, Weak, ?
D16, Sunny, Mild, Low, Strong, ?
D17, Overcast, Hot, Normal, Strong, ?
Gains Chart

Lift, gains and response curves,
Up to Here
Classification Problem - Formally

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Decision Tree For Playing Tennis

ROOT NODE

BRANCH

INTERNAL NODE

LEAF NODE

Disjunction of conjunctions
Mutually exclusive and exhaustive/complete
Decision Tree

Given:

- \( D = \{t_1, \ldots, t_n\} \) where \( t_i = <t_{i1}, \ldots, t_{im}> \)
- Database schema contains \( \{A_1, A_2, \ldots, A_h\} \)
- Classes \( C = \{C_1, \ldots, C_l\} \)

**Decision or Classification Tree** is a tree associated with \( D \) such that

- Each internal node is labeled with attribute, \( A_i \)
- Each arc is labeled with predicate which can be applied to attribute at parent
- Each leaf node is labeled with a class, \( C_j \)
Top-Down Tree Induction

Main loop:
1. \( A \leftarrow \) the “best” decision attribute for next node
2. Assign \( A \) as decision attribute for node
3. For each value of \( A \), create new descendant of node
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?

\[
\begin{array}{c}
[29+, 35-] \quad \text{A1=} ? \\
& \quad \text{t} \quad \text{f} \\
& \quad \begin{array}{c}
[21+, 5-] \\
[8+, 30-] \\
\end{array}
\end{array}
\quad \begin{array}{c}
[29+, 35-] \quad \text{A2=} ? \\
& \quad \text{t} \quad \text{f} \\
& \quad \begin{array}{c}
[18+, 33-] \\
[11+, 2-] \\
\end{array}
\end{array}
\]
Which Column and Split Point?

• Multitude of techniques:
  • Entropy/Information gain
  • Chi square test (CHAID)
    • Test of independence
  • GINI index
Information/Entropy

- Given probabilities $p_1, p_2, \ldots, p_s$ whose sum is 1, *Entropy* is defined as:

\[
H(p_1, p_2, \ldots, p_s) = \sum_{i=1}^{s} (p_i \log(1/p_i))
\]

- Entropy measures the amount of randomness or surprise or uncertainty.
- Goal in classification
  - no surprise
  - entropy = 0
Information Gain

\[ Gain(S, A) = \text{expected reduction in entropy due to sorting on } A \]

\[ Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \]
## Data Set

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Choosing the Next Attribute - 1

Which attribute is the best classifier?

\[ \text{Gain}(S, \text{ Humidity }) \]
\[ = 0.940 - (7/14)(0.985 - (7/14)(0.592) \]
\[ = 0.151 \]

\[ \text{Gain}(S, \text{ Wind }) \]
\[ = 0.940 - (8/14)(0.811 - (6/14)(1.0) \]
\[ = 0.048 \]
Stopping Criteria

- What type of tree will perfectly classify the training data (i.e. 100% training set accuracy)?
- Is this a bad thing?, Why?
- What does this tell you about the relationship between the dependent and independent attributes?
- Apriori stopping criteria, when:
  - A certain tree depth is reached
  - Number of records at a node goes below some threshold.
  - All potential splits are insignificant
- Want the tree to be simple, but not too simple
Training Set Error Should Approximately Equal Test Set Error Why? Optimum model complexity?
Trimming/Pruning Trees

• Stopping criterion can be somewhat arbitrary.

• Automatic pruning of trees
  – Ask the data, “How far should we split the data”.
  – Two general approaches:
    • Use part of the training set as a validation set
    • Use entire training set (usually an MDL approach).
Using Pruning To Prevent Overfitting

How can we avoid overfitting?

- stop growing when data split not statistically significant
- grow full tree, then post-prune

How to select “best” tree:

- Measure performance over training data
- Measure performance over separate validation data set
- MDL: minimize 
  \[ \text{size}(\text{tree}) + \text{size}(\text{misclassifications}(\text{tree})) \]
Reduced Error Pruning

Split data into training and validation set

Do until further pruning is harmful:

1. Evaluate impact on validation set of pruning each possible node (plus those below it)

2. Greedily remove the one that most improves validation set accuracy

- produces smallest version of most accurate subtree
- What if data is limited?
Reduced Error Pruning
<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Results of Reduced Error Pruning

Consider the use of learning a tree is to make prediction
What is the fundamental assumption that this learning algorithm is making
Lots of Interesting Issues … We Won’t Cover

- Complexity vs Error
- Occam’s razor
- Fixed training set size vs Error
- How to construct our training set
- Learning in the presence of an Oracle etc.

- Instead we will focus on more pragmatic issues.
Summary of DT

• Simple and efficient classification algorithm.
• Divides instance space into regions with hyper-planes such that one side is always parallel to one of the axes. Good, bad?
• What about naïve Bayes?
Bayes Theorem

\[ P(h|D) = \frac{P(D|h)P(h)}{P(D)} \]

- \( P(h) \) = prior probability of hypothesis \( h \)
- \( P(D) \) = prior probability of training data \( D \)
- \( P(h|D) \) = probability of \( h \) given \( D \)
- \( P(D|h) \) = probability of \( D \) given \( h \)
# Data Set

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Interpretation of a Probability

• Frequentist
  – Relative frequency of an event occurring
  – What about rare events?

• Degree of belief
  – Our belief that the event will occur
About the Hypothesis Space $P(h)$

• Priors
  – Each $h_i$ should be *Mutually exclusive*
  – Together the hypotheses must be *Totally exhaustive*
  – $\Sigma P(h_i)=1$
  – Priors encode knowledge before we see the data!
About the Data $P(D)$ and $P(D|H)$

- **Data, $P(D)$**
  - Data is considered to be a sample of all available data.
  - $P(D)$, probability the data will be observed given no knowledge of the hypothesis.
  - Constant for fixed data and if comparing hypotheses, can be ignored

- **Likelihood, $P(D|h)$**
  - Probability a hypothesis generated the observed data or probability of observing data given the hypothesis is true.
  - If the $n$ instances are independent then
    - $P(D|h) = P(D_1|h). P(D_2|h) \ldots P(D_n|h)$
  - Often use the Loglikelihood ($P(D|h)$).
Bayesian Posterior

• $P(h|D)$ is the posterior probability of the hypothesis (given the data).

• Usual aim of Bayesian learning is to find the MAP estimate
  – Most probable model in the model space
  – May be many highly probable models
A Simple Example

Does patient have cancer or not?

A patient takes a lab test and the result comes back positive. The test returns a correct positive result in only 98% of the cases in which the disease is actually present, and a correct negative result in only 97% of the cases in which the disease is not present. Furthermore, .008 of the entire population have this cancer.

\[
\begin{align*}
P(\text{cancer}) &= \\
P(+ | \text{cancer}) &= \\
P(+ | \neg \text{cancer}) &= \\
P(\neg \text{cancer}) &= \\
P(- | \text{cancer}) &= \\
P(- | \neg \text{cancer}) &= 
\end{align*}
\]
Choosing the Hypothesis

\[ P(h|D) = \frac{P(D|h)P(h)}{P(D)} \]

Generally want the most probable hypothesis given the training data

*Maximum a posteriori* hypothesis \( h_{MAP} \):

\[
\begin{align*}
  h_{MAP} &= \arg\max_{h \in H} P(h|D) \\
           &= \arg\max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\
           &= \arg\max_{h \in H} P(D|h)P(h)
\end{align*}
\]

Assume all hypothesis have equal probability

\[
h_{ML} = \arg\max_{h_i \in H} P(D|h_i)
\]
Brute Force MAP Learner

1. For each hypothesis $h$ in $H$, calculate the posterior probability

\[ P(h|D) = \frac{P(D|h)P(h)}{P(D)} \]

2. Output the hypothesis $h_{MAP}$ with the highest posterior probability

\[ h_{MAP} = \arg \max_{h \in H} P(h|D) \]
Naïve Bayes Classifier

Along with decision trees, neural networks, nearest nbr, one of the most practical learning methods.

When to use

• Moderate or large training set available

• Attributes that describe instances are conditionally independent given classification

Successful applications:

• Diagnosis

• Classifying text documents
Definition of NBC

Assume target function $f : X \rightarrow V$, where each instance $x$ described by attributes $\langle a_1, a_2 \ldots a_n \rangle$. Most probable value of $f(x)$ is:

$$v_{MAP} = \arg \max_{v_j \in V} P(v_j|a_1, a_2 \ldots a_n)$$

$$v_{MAP} = \arg \max_{v_j \in V} \frac{P(a_1, a_2 \ldots a_n|v_j)P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \arg \max_{v_j \in V} P(a_1, a_2 \ldots a_n|v_j)P(v_j)$$

Naive Bayes assumption:

$$P(a_1, a_2 \ldots a_n|v_j) = \prod_i P(a_i|v_j)$$

which gives

**Naive Bayes classifier:** $v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$
Coding the Algorithm

NaiveBayesLearn(examples)

For each target value $v_j$

$\hat{P}(v_j) \leftarrow \text{estimate } P(v_j)$

For each attribute value $a_i$ of each attribute $a$

$\hat{P}(a_i|v_j) \leftarrow \text{estimate } P(a_i|v_j)$

ClassifyNewInstance($x$)

$v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$
Data Set

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
NB: Simple Example

Consider *PlayTennis* again, and new instance

\[
\langle \text{Outlk} = \text{sun}, \text{Temp} = \text{cool}, \text{Humid} = \text{high}, \text{Wind} = \text{strong} \rangle
\]

Want to compute:

\[
v_{NB} = \arg\max_{v_j \in V} \prod_{i} P(a_i|v_j)
\]

\[
P(y) \ P(\text{sun}|y) \ P(\text{cool}|y) \ P(\text{high}|y) \ P(\text{strong}|y) = .005
\]

\[
P(n) \ P(\text{sun}|n) \ P(\text{cool}|n) \ P(\text{high}|n) \ P(\text{strong}|n) = .021
\]

\[
\rightarrow v_{NB} = n
\]
NB: Subtleties

Conditional independence assumption is often violated

\[ P(a_1, a_2 \ldots a_n|v_j) = \prod_i P(a_i|v_j) \]

- ...but it works surprisingly well anyway. Note don’t need estimated posteriors \( \hat{P}(v_j|x) \) correct; need only that

\[
\arg\max_{v_j \in V} \hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = \arg\max_{v_j \in V} P(v_j)P(a_1 \ldots, a_n|v_j)
\]

- see [Domingos & Pazzani, 1996] for analysis
- Naive Bayes posteriors often unrealistically close to 1 or 0
NB: Subtleties

2. what if none of the training instances with target value $v_j$ have attribute value $a_i$? Then

$$\hat{P}(a_i|v_j) = 0, \text{ and...}$$

$$\hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = 0$$

Typical solution is Bayesian estimate for $\hat{P}(a_i|v_j)$

$$\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$$

where

- $n$ is number of training examples for which $v = v_j$,
- $n_c$ number of examples for which $v = v_j$ and $a = a_i$
- $p$ is prior estimate for $\hat{P}(a_i|v_j)$
- $m$ is weight given to prior (i.e. number of “virtual” examples)
Performance Issues

• Very fast to compute (just need to count)
• What type of probabilities are used to make predictions, what are two differences to decision trees?
• What form is the function F for naïve Bayes

5 Conclusions

Despite its unrealistic independence assumption, the naive Bayes classifier is surprisingly effective in practice since its classification decision may often be correct even if its probability estimates are inaccurate. Although some optimality conditions of naive Bayes have been already identified in the past [2], a deeper understanding of data characteristics that affect the performance of naive Bayes is still required.
Adding Data Removes Uncertainty

Figure 1. Posterior Distribution for a Univariate Mixture Gaussian Model with 3 Labeled Examples

Figure 2. Posterior Distribution for a Univariate Mixture Gaussian Model with 4 Labeled and 4 Unlabeled Examples
Decision Regions for NB
## Data Set

Measure of quality of NB Model

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

Classification and Assumptions
Text Document Classification

Why?

- Learn which news articles are of interest
- Learn to classify web pages by topic

Naive Bayes is among most effective algorithms

What attributes shall we use to represent text documents??

Tell me all the assumptions made about the domain…
Target concept *Interesting*: Document $\to \{+, -\}$

1. Represent each document by vector of words
   - one attribute per word position in document

2. Learning: Use training examples to estimate
   - $P(\ +\ )$
   - $P(\ -\ )$
   - $P(docc|\ +\ )$
   - $P(docc|\ -\ )$

Naive Bayes conditional independence assumption

$$P(docc|v_j) = \prod_{i=1}^{\text{length}(doc)} P(a_i = w_k|v_j)$$

where $P(a_i = w_k|v_j)$ is probability that word in position $i$ is $w_k$, given $v_j$

one more assumption: $P(a_i = w_k|v_j) = P(a_m = w_k|v_j), \forall i, m$
\textsc{Learn\_Naive\_Bayes\_Text}(\textit{Examples}, V)

1. collect all words and other tokens that occur in \textit{Examples}
   
   \begin{itemize}
   \item \textit{Vocabulary} $\leftarrow$ all distinct words and other tokens in \textit{Examples}
   \end{itemize}

2. calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms
   
   \begin{itemize}
   \item For each target value $v_j$ in $V$ do
     
     \begin{itemize}
     \item $\textit{docs}_j \leftarrow$ subset of \textit{Examples} for which the target value is $v_j$
     \item $P(v_j) \leftarrow \frac{|\textit{docs}_j|}{|\textit{Examples}|}$
     \item $\textit{Text}_j \leftarrow$ a single document created by concatenating all members of $\textit{docs}_j$
     \item $n \leftarrow$ total number of words in $\textit{Text}_j$ (counting duplicate words multiple times)
     \item for each word $w_k$ in \textit{Vocabulary}
       
       \begin{itemize}
       \item $n_k \leftarrow$ number of times word $w_k$ occurs in $\textit{Text}_j$
       \item $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|\textit{Vocabulary}|}$
       \end{itemize}
   \end{itemize}
   \end{itemize}
\text{Classify\_naive\_Bayes\_text}(Doc)

- \textit{positions} \leftarrow \text{all word positions in } Doc \text{ that contain tokens found in } Vocabulary
- \text{Return } v_{NB}, \text{ where}

\[
v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_{i \in \text{positions}} P(a_i|v_j)
\]

Given 1000 training documents from each group
Learn to classify new documents according to which newsgroup it came from

\[
\text{comp.graphics} \quad \text{misc.forsale} \\
\text{comp.os.ms-windows.misc} \quad \text{rec.autos} \\
\text{comp.sys.ibm.pc.hardware} \quad \text{rec.motorcycles} \\
\text{comp.sys.mac.hardware} \quad \text{rec.sport.baseball} \\
\text{comp.windows.x} \quad \text{rec.sport.hockey}
\]
Accuracy

Article from rec.sport.hockey

Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year’s biggest and worst (opin
Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most
obvious candidate for pleasant surprise is Alex
Zhitnik. He came highly touted as a defensive
defenseman, but he’s clearly much more than that
Great skater and hard shot (though wish he were
more accurate). In fact, he pretty much allowed
the Kings to trade away that huge defensive
liability Paul Coffey. Kelly Hrudey is only the
biggest disappointment if you thought he was any
good to begin with. But, at best, he’s only a
mediocre goaltender. A better choice would be
Tomas Sandstrom, though not through any fault of
his own, but because some thugs in Toronto decid
Type of Model

- DT hyper-rectangles?
- NB decision surface?
- Some important properties of these algorithms
  - Order does not matter
  - One model/function or two?
  - Generative models need more data
  - Missing column values
  - Continuous column values
Another Motivating Example Not Involving “Carnies”

Consider the simple example of someone trying to deduce the weather from a piece of seaweed - folklore tells us that ‘soggy’ seaweed means wet weather, while ‘dry’ seaweed means sun. If it is in an intermediate state (‘damp’), then we cannot be sure. However, the state of the weather is not restricted to the state of the seaweed, so we may say on the basis of an examination that the weather is probably raining or sunny. A second useful clue would be the state of the weather on the preceding day (or, at least, its probable state) - by combining knowledge about what happened yesterday with the observed seaweed state, we might come to a better forecast for today.

What are the hidden states, What are the observables?
Finding the probability of an observed sequence

1. Exhaustive search for solution

We want to find the probability of an observed sequence given an HMM - that is, the parameters \((\pi, A, B)\) are known. Consider the weather example; we have a HMM describing the weather and its relation to the state of the seaweed, and we also have a sequence of seaweed observations. Suppose the observations for 3 consecutive days are \((\text{dry, damp, soggy})\) - on each of these days, the weather may have been sunny, cloudy or rainy. We can picture the observations and the possible hidden states as a trellis.
HMM Notation

- $a_{i,j}$ transition probability b/w state $i$ and $j$
- $b_{j,k}$ probability of emitting symbol/event $k$ when in state $j$
- $\theta = \{a_{i,j}, b_{j,k}\}, i,j = 1 \ldots c, k \in E$
- $V^T$ is a sequence of events $= \{v(1), \ldots, v(T)\}$
- $W$ is a random variable of the current state
- $V$, $v$, $W$, $w$ random variable companion function notation
Three Problems

• Consider the speech recognition example
• Evaluation
  – Did this particular HMM generate an event sequence.
    • \( P(V_T | \theta) \) (note dependency on model)
• Decoding
  – What sequence of states generated the event sequence
    • \( P(w_1, w_3, w_2, w_5, V_T | \theta) \)
• Learning/Induction/Parameter Estimation
  – Given many event sequences, find the most likely \( \theta \)
    • \( P(\theta_{Best} | V_T) \)
• Applications of these problem solutions
It can be seen that one method of calculating the probability of the observed sequence would be to find each possible sequence of the hidden states, and sum these probabilities. For the above example, there would be $3^3 = 27$ possible different weather sequences, and so the probability is

$$\Pr(\text{dry, damp, soggy } | \text{ HMM}) = \Pr(\text{dry, damp, soggy } | \text{ sunny, sunny, sunny}) + \Pr(\text{dry, damp, soggy } | \text{ sunny, sunny, cloudy}) + \Pr(\text{dry, damp, soggy } | \text{ sunny, sunny, rainy}) + \ldots$$

Complexity is approx. $O(c^T)$
We can calculate the probability of reaching an intermediate state in the trellis as the sum of all possible paths to that state.

For example, the probability of it being cloudy at $t = 2$ is calculated from the paths;
Decoding Problem

Trellis representation of an HMM
We denote the partial probability of state \( j \) at time \( t \) as \( \alpha_t(j) \).

This partial probability is calculated as:

\[
\alpha_t(j) = Pr(\text{observation | hidden state is } j) \times Pr(\text{all paths to state } j \text{ at time } t)
\]

The partial probabilities for the final observation hold the probability of reaching those states going through all possible paths - e.g., for the above trellis, the final partial probabilities are calculated from the paths.
The number of paths needed to calculate $\alpha$ increases exponentially as the length of the observation sequence increases but the $\alpha$'s at time $t-1$ give the probability of reaching that state through all previous paths, and we can therefore define $\alpha$'s at time $t$ in terms of those at time $t-1$ - i.e.,

$$
\alpha_{t+1}(j) = b_{jk_{t+1}} \sum_{i=1}^{n} \alpha_{t}(i) a_{ij}
$$
Forward Algorithm

- Trick?
- Input: t=0, a_{i,j}, b_{j,k}, V^T, alpha(0)
- Repeat t=t+1
  - alpha(t) = b_{j,v(t)} \sum_i=1 \alpha_i(t-1) a_{ij}
- Until t=T
- Return ?????
- Complexity
Verterbi Algorithm

- For the decoding problem. Use the Trellis again.

1. Exhaustive search for a solution

We can use a picture of the execution trellis to visualise the relationship between states and observations.

- Sunny
- Cloudy
- Rainy

Observations: dry, damp, soggy
Markov Assumption

• Sequence of events
  – 1\textsuperscript{st} order Markov chain, \( P(S^t|S^{t-1}, \ldots S^1) = P(S^t|S^{t-1}) \)
  – \( k\)\textsuperscript{th} order \( P(S^t|S^{t-1}, \ldots S^1) = P(S^t|S^{t-1} \ldots S^{t-k}) \)

• Markov chains vs Markov processes

• Stationary distribution assumption

• Ergodicity
Gene Finding
(An Ideal HMM Domain)

• Objective:
  – To find the coding and non-coding regions of an unlabeled string of DNA nucleotides

• Our Motivation:
  – Assist in the annotation of genomic data produced by genome sequencing methods
  – Gain insight into the mechanisms involved in transcription, splicing and other processes
Gene Finding Terminology

• A string of DNA nucleotides containing a gene will have separate regions (lines):
  – Introns – non-coding regions within a gene
  – Exons – coding regions
Gene Finding Challenges
Protein Family Modeling
(A clever fit of HMMs)

• I have a protein sequence.

• What family is it in?
• Can you give me a quick alignment to the other members of the family?
• These amino acids here, do they match the families consensus positions, or are they inserts?
Trivial Evaluation Algorithm? It’s complexity?
Evaluation - Trivial

- \( \text{P}(V_T) = \text{Sum}_r \text{P}(V_T|w_r)\text{P}(w_r) \)  \hspace{1cm} (1)
- \( \text{P}(w_r) = \text{Prod} \text{P}(w(t)|w(t-1)) \)  \hspace{1cm} (2)
- \( \text{P}(V_T|w_r) = \text{Prod}_t (\text{P}(v(t)|w(t)) \)  \hspace{1cm} (3)
- Substitute it together.
- What is the complexity?
Evaluation

• Can use total enumeration of all possible state paths to calculate it: $O(c^T T)$
• Keep ongoing variable $\alpha_j(t): O(c^2 T)$
• Forward Algorithm
• Exploiting structure to make evaluation quicker
Up To Here
Data Set

Measure of quality of NB Model

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

Classification and Assumptions
Text Document Classification

Why?

- Learn which news articles are of interest
- Learn to classify web pages by topic

Naive Bayes is among most effective algorithms

What attributes shall we use to represent text documents??

Tell me all the assumptions made about the domain…
Target concept *Interesting:? : Document → {+, −}*

1. Represent each document by vector of words
   - one attribute per word position in document

2. Learning: Use training examples to estimate
   - \( P(+) \)
   - \( P(−) \)
   - \( P(\text{doc}|+) \)
   - \( P(\text{doc}|−) \)

Naive Bayes conditional independence assumption

\[
P(\text{doc}|v_j) = \prod_{i=1}^{\text{length}(\text{doc})} P(a_i = w_k|v_j)
\]

where \( P(a_i = w_k|v_j) \) is probability that word in position \( i \) is \( w_k \), given \( v_j \)

one more assumption: \( P(a_i = w_k|v_j) = P(a_m = w_k|v_j), \forall i, m \)
**Learn Naive Bayes Text** (Examples, V)

1. **Collect all words and other tokens that occur in Examples**
   - Vocabulary ← all distinct words and other tokens in Examples

2. **Calculate the required** $P(v_j)$ **and** $P(w_k|v_j)$ **probability terms**
   - For each target value $v_j$ in V do
     - $docs_j$ ← subset of Examples for which the target value is $v_j$
     - $P(v_j) ← \frac{|docs_j|}{|Examples|}$
     - $Text_j$ ← a single document created by concatenating all members of $docs_j$
     - $n ←$ total number of words in $Text_j$ (counting duplicate words multiple times)
     - for each word $w_k$ in Vocabulary
       * $n_k ←$ number of times word $w_k$ occurs in $Text_j$
       * $P(w_k|v_j) ← \frac{n_k+1}{n+|Vocabulary|}$
CLASSIFY_naive_BAYES_TEXT(Doc)

- \( positions \leftarrow \) all word positions in Doc that contain tokens found in Vocabulary

- Return \( v_{NB} \), where

\[
v_{NB} = \text{argmax}_{v_j \in V} P(v_j) \prod_{i \in positions} P(a_i|v_j)
\]

Given 1000 training documents from each group
Learn to classify new documents according to which newsgroup it came from

- comp.graphics
- comp.os.ms-windows.misc
- comp.sys.ibm.pc.hardware
- comp.sys.mac.hardware
- comp.windows.x
- misc.forsale
- rec.autos
- rec.motorcycles
- rec.sport.baseball
- rec.sport.hockey
HMM Introduction

• A motivating example
  – Consider tossing a fair coin
  – A fair and a biased coin that I swap between

• **State** and transitions are hidden, hence the term Hidden Markov Model

• All you see are the **Events**: HTHTTTTHHH

• Speech recognition example

• Other applications, bioinformatics, web-page browsing behavior, human movement tracking, various types of sequence analysis
Another Motivating Example Not Involving “Carnies”

Consider the simple example of someone trying to deduce the weather from a piece of seaweed - folklore tells us that `soggy` seaweed means wet weather, while `dry` seaweed means sun. If it is in an intermediate state (`damp`), then we cannot be sure. However, the state of the weather is not restricted to the state of the seaweed, so we may say on the basis of an examination that the weather is probably raining or sunny. A second useful clue would be the state of the weather on the preceding day (or, at least, its probable state) - by combining knowledge about what happened yesterday with the observed seaweed state, we might come to a better forecast for today.

What are the hidden states, What are the observables?
Consider a set of traffic lights; the sequence of lights is red - red/amber - green - amber - red. The sequence can be pictured as a state machine, where the different states of the traffic lights follow each other.
Finding the probability of an observed sequence

1. Exhaustive search for solution

We want to find the probability of an observed sequence given an HMM - that is, the parameters \((\pi, A, B)\) are known. Consider the weather example; we have a HMM describing the weather and its relation to the state of the seaweed, and we also have a sequence of seaweed observations. Suppose the observations for 3 consecutive days are \((\text{dry}, \text{damp}, \text{soggy})\) - on each of these days, the weather may have been sunny, cloudy or rainy. We can picture the observations and the possible hidden states as a trellis.

![Trellis diagram for weather states and observations](image-url)
It can be seen that one method of calculating the probability of the observed sequence would be to find each possible sequence of the hidden states, and sum these probabilities. For the above example, there would be $3^3=27$ possible different weather sequences, and so the probability is

$$\Pr(\text{dry, damp, soggy} \mid \text{HMM}) = \Pr(\text{dry, damp, soggy} \mid \text{sunny, sunny, sunny}) + \Pr(\text{dry, damp, soggy} \mid \text{sunny, sunny, cloudy}) + \Pr(\text{dry, damp, soggy} \mid \text{sunny, sunny, rainy}) + \ldots \ldots \Pr(\text{dry, damp, soggy} \mid \text{rainy, rainy, rainy})$$

Complexity is approx. $O(c^T)$
Efficient Computation Using ???

We can calculate the probability of reaching an intermediate state in the trellis as the sum of all possible paths to that state.

For example, the probability of it being cloudy at \( t = 2 \) is calculated from the paths;
We denote the partial probability of state \( j \) at time \( t \) as \( \alpha_t (j) \) - this partial probability is calculated as:

\[
\alpha_t (j) = \Pr(\text{observation} | \text{hidden state is } j) \times \Pr(\text{all paths to state } j \text{ at time } t)
\]

The partial probabilities for the final observation hold the probability of reaching those states going through all possible paths - e.g., for the above trellis, the final partial probabilities are calculated from the paths:
The number of paths needed to calculate $\alpha$ increases exponentially as the length of the observation sequence increases but the $\alpha$'s at time $t-1$ give the probability of reaching that state through all previous paths, and we can therefore define $\alpha$'s at time $t$ in terms of those at time $t-1$ -i.e.,

$$\alpha_{t+1}(j) = b_{j k_{t+1}} \sum_{i=1}^{n} \alpha_t(i) a_{ij}$$
Forward Algorithm

- Trick?
- Input: $t=0, a_{i,j}, b_{j,k}, V^T, \alpha(0)$
- Repeat $t=t+1$
  - $\alpha(t) = b_{jk}v(t)\sum_{i=1}^{t} \alpha_i(t-1)a_{ij}$
- Until $t=T$
- Return ????
- Complexity
Verterbi Algorithm

- For the decoding problem. Use the Trellis again.

1. Exhaustive search for a solution

We can use a picture of the execution trellis to visualise the relationship between states and observations.

Observations: dry damp soggy
Markov Assumption

- Sequence of events
  - 1st order Markov chain, $P(S_t|S_{t-1},...S^l)=P(S_t|S_{t-1})$
  - $k^{th}$ order $P(S_t|S_{t-1},...S^l)=P(S_t|S_{t-1}...S^{t-k})$

- Markov chains vs Markov processes
- Stationary distribution assumption
- Ergodicity
HMM Notation

• $a_{i,j}$ transition probability b/w state $i$ and $j$
• $b_{j,k}$ probability of emitting symbol/event $k$ when in state $j$
• $\theta=\{a_{i,j}, b_{j,k}\}, i,j = 1 \ldots c, k \in E$
• $V^T$ is a sequence of events $= \{v(1), \ldots, v(T)\}$
• $W$ is a random variable of the current state
• $V, v, W, w$ random variable companion function notation
Three Problems

- Consider the web browsing behavior example
- Evaluation
  - Did this particular HMM generate an event sequence.
  - $P(V^T | \theta)$ (note dependency on model)
- Decoding
  - What sequence of states generated the event sequence
  - $P(w_1,w_3,w_2,w_5, V^T | \theta)$
- Learning/Induction/Parameter Estimation
  - Given many event sequences, find the most likely $\theta$
  - $P(\theta_{Best} | V^T)$
- Applications of these problem solutions
Gene Finding  
(An Ideal HMM Domain)

• Objective:
  – To find the coding and non-coding regions of an unlabeled string of DNA nucleotides

• Our Motivation:
  – Assist in the annotation of genomic data produced by genome sequencing methods
  – Gain insight into the mechanisms involved in transcription, splicing and other processes
Gene Finding Terminology

• A string of DNA nucleotides containing a gene will have separate regions (lines):
  – Introns – non-coding regions within a gene
  – Exons – coding regions
Gene Finding Challenges
Protein Family Modeling
(A clever fit of HMMs)

• I have a protein sequence.

• What family is it in?
• Can you give me a quick alignment to the other members of the family?
• These amino acids here, do they match the families consensus positions, or are they inserts?
Decoding Problem

Trellis representation of an HMM
Trivial Evaluation Algorithm?
It’s complexity?
Evaluation - Trivial

- $P(V^T) = \text{Sum}_r P(V^T|w_r)P(w_r)$ (1)
- $P(w_r) = \text{Prod} P(w(t)|w(t-1)$ (2)
- $P(V^T|w_r) = \text{Prod}_t (P(v(t)|w(t))$ (3)
- Substitute it together.
- What is the complexity?
Evaluation

- Can use total enumeration of all possible state paths to calculate it: $O(c^T T)$
- Keep ongoing variable $\alpha_j(t): O(c^2 T)$
- Forward Algorithm
- Exploiting structure to make evaluation quicker
Example

- A “carnie” comes to town and starts asking for bets on the outcome of coin tosses. You can only bet it will be “H.” Is he using one fair coin or two coins (one fair, one biased)?

- Two models (one: there is a fair coin toss, two: multiple coins are being used)

- Note, there could be other explanations, just asking which is “best”

- M1 a=?, b=?

- M2 a={⅜, ¼, ⅜, ¼}, b={1/2, 3/4}

- Observation of strings is THHTTTTHH (note #T equal #F)

- Is he using a single fair coin?
Decoding Algorithm

• Maybe the “carnie” systematically changes between the biased and fair coin. If you decode when these changes occur you could win.
• Path={}, t=0
• For(t=t+1)
  – For(j=j+1)
    • $\text{Alpha}_j(t) = b_{jk} v(t) \sum_i \alpha_i(t-1) a_{ij}$
    – Until $j=c$
    – $J' = \text{argmax}_j(\text{alpha}_j(t))$
    – Append $w_{j'}$ to Path
• Until $t=T$
Estimating HMM

- How: using EM. Why?
- View data as what?? “TFFFTTFF”
- Iterative algorithm, like k-means, we are continually updating what?
- Two extra variables $\alpha_i(t)$, $\beta_i(t)$
- $\Gamma_{ij} = \alpha_i(t-1)a_{ij}\beta_j(t) / P(V^T|\theta)$
- $a_{ij-hat} = \sum_t \gamma_{ij}(t) / \sum_t \sum_k \gamma_{ik}(t)$
- $b_{jk-hat} = \sum_{t,v(t)=vk} \sum_l \gamma_{jl}(t) / \sum_t \sum_l \gamma_{jl}(t)$
- What is the correct probability for the HMM
Reading

• Next class, clustering in graphs (i.e. social network finding)

• Optional, “A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models”
http://citeseer.nj.nec.com/bilmes98gentle.html
HMM – Three Perspectives

\[ P(\{x\}, \{y\}) = \prod_{t=1}^{T} P(x_t|x_{t-1})P(y_t|x_t) \]

- you can think of this as:
  
  Markov chain with stochastic measurements.

or

Mixture model with states coupled across time.
How Can We Estimate the Parameters

- Transition probability parameters (A)
- Symbol generation probabilities (B)
- Recall we only have the string:
  - TTTTTFFFFFTTTTTTTFFFFFTTT?
- What would make the problem easy?
  - Think of divide and conquer …
- Let’s draw the problem graphically.
- Classic solution
Graph of HMM Estimation
Problem – How is X Generated

A

Z

X

B
Graph of HMM Estimation
Problem – How is X Generated

A → Z → X → B

[Diagram of HMM estimation showing nodes A, Z, X, and B with arrows connecting them]
Graph of HMM Estimation
Problem – How is A, B Calculated

A

Z

X

B
Graph of HMM Estimation
Problem – How is A, B Calculated
Graph of HMM Estimation Problem – Putting it Together
Classic Approach to Such Problems

- When two unknowns depend on one another.
  - a) Hold one of them fixed
  - b) Estimate the other
  - c) Repeat
- In the case of HMM this estimation algorithm is known as the Baum-Welch alg and is a special case of generalized EM.
Baum-Welch Algorithm - 1

• I’ll assume there is one big sequence …
• We can never know with certainty what state the machine is in …
• So let $z_{tij}$ be the chance I’m in state i at time $t$ and then transition to j at $t+1$.
• $z_{tij} = ??$
• Recall from last week …
Decoding Problem

Trellis representation of an HMM

\[ \text{Time=} \quad 1 \quad k \quad k+1 \quad K \]
Baum-Welch Algorithm - 2

- $\alpha_{i,t}$ is ...
- $\beta_{j,t+1}$ is ...
- $z_{tij} =$
- $\gamma_{t,i} =$
- $\gamma_{t,i} =$
Baum-Welch Algorithm - 3

• $\alpha_{i,t}$ is …
• $\beta_{j,t+1}$ is …
• $z_{tij} = \alpha_{i,t} b_{v(t)} a_{ij} \beta_{j,t+1} / \sum_{i,j} \alpha_{i,t} b_{v(t)} a_{ij} \beta_{j,t+1}$ (1)
• $\gamma_{t,i} = \alpha_{i,t} \beta_{j,t} / \sum_i \alpha_{i,t} \beta_{j,t}$ (2)
• $\gamma_{t,i} = \sum_j z_{tij}$ (3)
Baum-Welch Algorithm -4

• \( z_{tij} = \alpha_{i,t} b_{v(t)} a_{ij} \beta_{j,t+1} / \sum_{i,j} \alpha_{i,t} b_{v(t)} a_{ij} \beta_{j,t+1} \) (1)
• \( \gamma_{t,i} = \alpha_{i,t} \beta_{j,t} / \sum_{i} \alpha_{i,t} \beta_{j,t} \) (2)
• \( \gamma_{t,i} = \sum_{j} z_{tij} \) (3)
• \( a'_{i,j} = \)
• \( b'_{i,j} = \)
Baum-Welch Algorithm -5

- \( z_{tij} = \alpha_{i,t} b_{v(t)} a_{ij} \beta_{j,t+1} / \sum_{i,j} \alpha_{i,t} b_{v(t)} a_{ij} \beta_{j,t+1} \)  
  \( i = 1, \ldots, N \) \( j = 1, \ldots, M \) \( t = 1, \ldots, T \) (1)

- \( \gamma_{t,i} = \alpha_{i,t} \beta_{j,t} / \sum_i \alpha_{i,t} \beta_{j,t} \)  
  \( i = 1, \ldots, N \) \( j = 1, \ldots, M \) \( t = 1, \ldots, T \) (2)

- \( \gamma_{t,i} = \sum_j z_{tij} \)  
  \( i = 1, \ldots, N \) \( t = 1, \ldots, T \) (3)

- \( a'_{i,j} = \sum_t z_{tij} / \sum_t \gamma_{t,i} \)  
  \( i = 1, \ldots, N \) \( j = 1, \ldots, M \) \( t = 1, \ldots, T \) (4)

- \( b'_{i,j} = \sum_{t,v(t)=j} \gamma_{t,i} / \sum_t \gamma_{t,i} \)  
  \( i = 1, \ldots, N \) \( j = 1, \ldots, M \) \( t = 1, \ldots, T \) (5)
Baum-Welch Algorithm -6

- Randomly Guess A and B, Repeat two steps

- **E Step**
  - \( z_{tij} = \alpha_{i,t} b_{v(t)} a_{ij} \beta_{j,t+1} / \sum_{i,j} \alpha_{i,t} b_{v(t)} a_{ij} \beta_{j,t+1} \) (1)
  - \( \gamma_{t,i} = \alpha_{i,t} \beta_{j,t} / \sum_{i} \alpha_{i,t} \beta_{j,t} \) (2)
  - \( \gamma_{t,i} = \sum_{j} z_{tij} \) (3)

- **M Step**
  - \( a'_{i,j} = \sum_{t} z_{tij} / \sum_{t} \gamma_{t,i} \) (4)
  - \( b'_{i,j} = \sum_{t,v(t)=j} \gamma_{t,i} / \sum_{t} \gamma_{t,i} \) (5)