Today’s Schedule

1. Bayes decision rule review
2. The curse of dimensionality
3. Naive Bayes
4. Text classification using Naive Bayes (introduction)

Bayes decision rule (recap)

Class $C = \{1, \ldots, K\}$; $C_k$ to denote $C = k$; input features $X = x$

Most probable class: (maximum posterior class)

$$k_{\text{max}} = \arg \max_{k \in C} P(C_k | x) = \arg \max_{k \in C} \frac{P(x | C_k) P(C_k)}{\sum_{j=1}^{K} P(x | C_j) P(C_j)}$$

$$= \arg \max_{k \in C} P(x | C_k) P(C_k)$$

where $P(C_k | x)$: posterior

$P(x | C_k)$: likelihood

$P(C_k)$: prior

$\Rightarrow$ Minimum error (misclassification) rate classification

(PRML C. M. Bishop (2006) Section 1.5)

Fish classification (revisited)

Bayesian class estimation:

$$P(C_k | x) = \frac{P(x | C_k) P(C_k)}{P(x)}$$

Estimating the terms: (Non-Bayesian)

Priors: $P(C = M) \approx \frac{N_M}{N_M + N_F}$

Likelihoods: $P(x | C = M) \approx \frac{m_M(x)}{N_M}$

NB: These approximations work only if we have enough data

Fish classification (revisited)

$$P(C_k | x) = \frac{P(x | C_k) P(C_k)}{P(x)}$$

More features?

$$P(x | C_k) \approx \frac{n_{C_k}(x_1, \ldots, x_0)}{N_{C_k}}$$

1D histogram: $n_{C_k}(x_1)$

2D histogram: $n_{C_k}(x_1, x_2)$

3D cube of numbers: $n_{C_k}(x_1, x_2, x_3)$

100 binary variables, $2^{100}$ settings

In high dimensions almost all $n_{C_k}(x_1, \ldots, x_0)$ are zero

$\Rightarrow$ Bellman’s “curse of dimensionality”

Avoiding the Curse of Dimensionality

Apply the chain rule:

$$P(x | C_k) = P(x_1, x_2, \ldots, x_0 | C_k)$$

$$= P(x_1 | x_2, \ldots, x_0, C_k) P(x_2 | x_3, x_1, C_k) P(x_3 | x_4, x_2, C_k) \cdots$$

$$\cdots P(x_{n-1} | x_n, \ldots, x_0, C_k) P(x_n | x_0, \ldots, x_1, C_k)$$

Solution: assume structure in $P(x | C_k)$

For example,

- Assume $x_{n+1}$ depends on $x_0$ only

  $$P(x | C_k) \approx P(x_1, x_2, x_3 | x_0, C_k)$$

- Assume $x \in \mathcal{R}^D$ distributes in a low dimensional vector space
  - Dimensionality reduction by PCA (Principal Component Analysis) / KL-transform
Avoiding the Curse of Dimensionality (cont.)

- Apply smoothing windows (e.g., Parzen windows)
- Apply a probability distribution model (e.g., Normal dist.)
- Assume $x_1, \ldots, x_D$ are conditionally independent given class
  
  $\Rightarrow$ Naive Bayes rule/model/assumption
  
  (or idiot Bayes rule)

  $$P(x_1, x_2, \ldots, x_D | C_k) = P(x_1 | C_k) P(x_2 | C_k) \cdots P(x_D | C_k)$$

  $$= \prod_{d=1}^{D} P(x_d | C_k)$$

- Is it reasonable?
  
  Often not, of course!
  Although it can still be useful.

---

Example - game played depending on the weather

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>NO</td>
</tr>
<tr>
<td>sunny</td>
<td>hot</td>
<td>high</td>
<td>true</td>
<td>NO</td>
</tr>
<tr>
<td>overcast</td>
<td>hot</td>
<td>high</td>
<td>false</td>
<td>YES</td>
</tr>
<tr>
<td>rainy</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>NO</td>
</tr>
<tr>
<td>rainy</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>YES</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>false</td>
<td>NO</td>
</tr>
<tr>
<td>sunny</td>
<td>cool</td>
<td>normal</td>
<td>false</td>
<td>YES</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>normal</td>
<td>true</td>
<td>YES</td>
</tr>
<tr>
<td>overcast</td>
<td>cool</td>
<td>normal</td>
<td>true</td>
<td>YES</td>
</tr>
<tr>
<td>sunny</td>
<td>mild</td>
<td>high</td>
<td>false</td>
<td>NO</td>
</tr>
</tbody>
</table>

$$P(Play | O, T, H, W) = \frac{P(O, T, H, W | Play) P(Play)}{P(O, T, H, W)}$$

If we use histograms for this

4D data: $\eta_{st}(O, T, H, W)$

- sunny overcast $\propto$ mild high
- rainy $\propto$ cool normal
- overcast $\propto$ mild normal
- windy $\propto$ true false

# of bins in the histogram = $3 \times 3 \times 2 \times 2 = 36$

# of samples available = 9 for play:yes, 5 for play:no

---

Weather data summary

<table>
<thead>
<tr>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outlook</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>sunny</td>
</tr>
<tr>
<td>sunny</td>
</tr>
<tr>
<td>overcast</td>
</tr>
<tr>
<td>rainy</td>
</tr>
<tr>
<td>overcast</td>
</tr>
<tr>
<td>rainy</td>
</tr>
<tr>
<td>sunny</td>
</tr>
<tr>
<td>windy</td>
</tr>
</tbody>
</table>

Relative frequencies $P(x | Play = Y)$, $P(x | Play = N)$

Test example

$$x = (\text{ sunny, cool, high, true })$$

---

Applying Naive Bayes

- Posterior prob. of "play" given $x = (\text{sunny, cool, humid, windy})$

  $$P(Play | x) \propto P(x | Play) P(Play)$$

  $$P(Play = Y | x) \propto P(O = o | Y) P(T = t | Y) P(H = h | Y) P(W = w | Y)$$

    $$\propto \frac{2}{9} \frac{3}{9} \frac{6}{9} \frac{6}{9} \approx 0.0053$$

  $$P(Play = N | x) \propto P(O = o | N) P(T = t | N) P(H = h | N) P(W = w | N)$$

    $$\propto \frac{3}{5} \frac{1}{5} \frac{4}{5} \frac{5}{14} \approx 0.0206$$

Exercise: find the odds of play, $P(\text{play} = Y | x)/P(\text{play} = N | x)$

---

Naive Bayes properties

- Easy and cheap:
  
  Record counts, convert to frequencies, score each class by multiplying prior and likelihood terms

  $$P(C_k | x) \propto \prod_{D=1} P(x_d | C_k) P(C_k)$$

- Statistically viable:
  
  Simple count-based estimates work in 1D

- Often overconfident:
  
  Treats dependent evidence as independent
Identifying Spam

Question

How can we identify an email as spam automatically?

Text classification: classify email messages as spam or non-spam (ham), based on the words they contain.

With the Bayes decision rule,
\[ P(\text{Spam}|x_1, \ldots, x_L) \propto P(x_1, \ldots, x_L|\text{Spam})P(\text{Spam}) \]

Using the naive Bayes assumption,
\[ P(x_1, \ldots, x_L|\text{Spam}) = P(x_1|\text{Spam}) \cdots P(x_L|\text{Spam}) \]
Summary

- The curse of dimensionality
- Approximation by the Naive Bayes rule
- Example: classifying multidimensional data using Naive Bayes
- Next lecture: Text classification using Naive Bayes