Artificial Intelligence

Naïve Bayes Classifier
Ch. 20.2
Decision Tree

• **Strength**
  – Decision trees are able to generate understandable rules.
  – Decision trees perform classification without requiring much computation.
  – Decision trees are able to handle both continuous and categorical variables.
  – Decision trees provide a clear indication of which fields are most important for prediction or classification

• **Weakness**
  – Error-prone with many classes
  – Cannot handle missing data
  – Hard to update
Bayesian Methods

- Learning and classification methods based on probability theory.
- Bayes theorem plays a critical role in probabilistic learning and classification.
- Uses *prior* probability of each category given no information about an item.
- Categorization produces a *posterior* probability distribution over the possible categories given a description of an item.
Bayes Rule

• Given a hypothesis $h$ and data $D$ which bears on the hypothesis:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- $P(h)$: independent probability of $h$: prior probability
- $P(D)$: independent probability of $D$
- $P(D|h)$: conditional probability of $D$ given $h$: likelihood
- $P(h|D)$: conditional probability of $h$ given $D$: posterior probability
Maximum A Posterior

- Based on Bayes Theorem, we can compute the **Maximum A Posterior** (MAP) hypothesis for the data.
- We are interested in the best hypothesis for some space $H$ given observed training data $D$.

$$h_{MAP} \equiv \arg\max_{h \in H} P(h \mid D)$$

$$= \arg\max_{h \in H} \frac{P(D \mid h)P(h)}{P(D)}$$

$$= \arg\max_{h \in H} P(D \mid h)P(h)$$

$H$: set of all hypothesis.

Note that we can drop $P(D)$ as the probability of the data is constant (and independent of the hypothesis).
Maximum Likelihood

• Now assume that all hypotheses are equally probable a priori, i.e., $P(h_i) = P(h_j)$ for all $h_i, h_j$ belong to $H$.

• This is called assuming a uniform prior. It simplifies computing the posterior:

$$h_{ML} = \arg \max_{h \in H} P(D | h)$$

• This hypothesis is called the maximum likelihood hypothesis.
Desirable Properties of Bayes Classifier

- **Incrementality**: with each training example, the prior and the likelihood can be updated dynamically: flexible and robust to errors.
- **Combines prior knowledge and observed data**: prior probability of a hypothesis multiplied with probability of the hypothesis given the training data.
- **Probabilistic hypothesis**: outputs not only a classification, but a probability distribution over all classes.
Bayes Classifiers

**Assumption:** training set consists of instances of different classes described \( c_j \) as conjunctions of attributes values

**Task:** Classify a new instance \( d \) based on a tuple of attribute values into one of the classes \( c_j \in C \)

**Key idea:** assign the most probable class \( c_{MAP} \) using Bayes Theorem.

\[
c_{MAP} = \arg \max_{c_j \in C} P(c_j \mid x_1, x_2, \ldots, x_n) = \arg \max_{c_j \in C} \frac{P(x_1, x_2, \ldots, x_n \mid c_j)P(c_j)}{P(x_1, x_2, \ldots, x_n)} = \arg \max_{c_j \in C} P(x_1, x_2, \ldots, x_n \mid c_j)P(c_j)
\]
**Parameters estimation**

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.
- $P(x_1, x_2, \ldots, x_n | c_j)$
  - $O(|X|^n \cdot |C|)$ parameters
  - Could only be estimated if a very, very large number of training examples was available.
- **Independence Assumption**: attribute values are conditionally independent given the target value: naïve Bayes.

\[
P(x_1, x_2, \ldots, x_n \mid c_j) = \prod_i P(x_i \mid c_j)
\]

\[
c_{NB} = \arg \max_{c_j \in C} P(c_j) \prod_i P(x_i \mid c_j)
\]
Properties

- Estimating $P(x_i \mid c_j)$ instead of $P(x_1, x_2, \ldots, x_n \mid c_j)$ greatly reduces the number of parameters (and the data sparseness).
- The learning step in Naïve Bayes consists of estimating $P(x_i \mid c_j)$ and $P(c_j)$ based on the frequencies in the training data.
- An unseen instance is classified by computing the class that maximizes the posterior.
- When conditioned independence is satisfied, Naïve Bayes corresponds to MAP classification.
Underflow Prevention

• Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow.
• Since \( \log(xy) = \log(x) + \log(y) \), it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.
• Class with highest final un-normalized log probability score is still the most probable.

\[ c_{NB} = \arg \max_{c_j \in C} \log P(c_j) + \sum_{i \in \text{positions}} \log P(x_i | c_j) \]
Smoothing to Avoid Overfitting

\[ \hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k} \]

- Somewhat more subtle version

\[ \hat{P}(x_{i,k} \mid c_j) = \frac{N(X_i = x_{i,k}, C = c_j) + mp_{i,k}}{N(C = c_j) + m} \]

- Overall fraction in data where \( X_i = x_{i,k} \)
- Extent of "smoothing"
The table below represents a set of examples with attributes and their corresponding targets. The attributes include whether an item is an example (Alt), whether the bar is closed (Bar), whether the item is French (Fri), the quantity of honey (Hun), whether the price is full (Pat), the price level (Price), whether the rain is present (Rain), whether the item is a restaurant (Res), the type of cuisine (Type), the estimated wait time (Est), and the target attribute (Wait).

### Example Table

<table>
<thead>
<tr>
<th>Example</th>
<th>Alt</th>
<th>Bar</th>
<th>Fri</th>
<th>Hun</th>
<th>Pat</th>
<th>Price</th>
<th>Rain</th>
<th>Res</th>
<th>Type</th>
<th>Est</th>
<th>Wait</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Some</td>
<td>$$$$</td>
<td>F</td>
<td>T</td>
<td>French</td>
<td>0–10</td>
<td>T</td>
</tr>
<tr>
<td>X2</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Full</td>
<td>$</td>
<td>F</td>
<td>F</td>
<td>Thai</td>
<td>30–60</td>
<td>F</td>
</tr>
<tr>
<td>X3</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Some</td>
<td>$</td>
<td>F</td>
<td>F</td>
<td>Burger</td>
<td>0–10</td>
<td>T</td>
</tr>
<tr>
<td>X4</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>Full</td>
<td>$</td>
<td>F</td>
<td>F</td>
<td>Thai</td>
<td>10–30</td>
<td>T</td>
</tr>
<tr>
<td>X5</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>Full</td>
<td>$$$$</td>
<td>F</td>
<td>T</td>
<td>French</td>
<td>&gt;60</td>
<td>F</td>
</tr>
<tr>
<td>X6</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Some</td>
<td>$$</td>
<td>T</td>
<td>T</td>
<td>Italian</td>
<td>0–10</td>
<td>T</td>
</tr>
<tr>
<td>X7</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>None</td>
<td>$</td>
<td>T</td>
<td>F</td>
<td>Burger</td>
<td>0–10</td>
<td>T</td>
</tr>
<tr>
<td>X8</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>Some</td>
<td>$$</td>
<td>T</td>
<td>T</td>
<td>Thai</td>
<td>0–10</td>
<td>T</td>
</tr>
<tr>
<td>X9</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>Full</td>
<td>$</td>
<td>T</td>
<td>F</td>
<td>Burger</td>
<td>&gt;60</td>
<td>F</td>
</tr>
<tr>
<td>X10</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Full</td>
<td>$$$$</td>
<td>F</td>
<td>T</td>
<td>Italian</td>
<td>10–30</td>
<td>F</td>
</tr>
<tr>
<td>X11</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>None</td>
<td>$</td>
<td>F</td>
<td>F</td>
<td>Thai</td>
<td>0–10</td>
<td>F</td>
</tr>
<tr>
<td>X12</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Full</td>
<td>$</td>
<td>F</td>
<td>F</td>
<td>Burger</td>
<td>30–60</td>
<td>T</td>
</tr>
</tbody>
</table>

The model uses Bayes' theorem to predict the target attribute based on the given attributes. The formula for predicting the target attribute is:

\[
\Pr(A = t | D) = \Pr(D) \Pr(A) \Pr(D | A) \Pr(C | A) \ldots \Pr(Z | A)
\]