Bayesian Classification: Why?

- **Probabilistic learning**: Calculate explicit probabilities for hypothesis, among the most practical approaches to certain types of learning problems.
- **Incremental**: Each training example can incrementally increase/decrease the probability that a hypothesis is correct. Prior knowledge can be combined with observed data.
- **Probabilistic prediction**: Predict multiple hypotheses, weighted by their probabilities.
- **Standard**: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured.
Bayesian Theorem: Basics

- Let $X$ be a data sample whose class label is unknown
- Let $H$ be a hypothesis that $X$ belongs to class $C$
- For classification problems, determine $P(H/X)$: the probability that the hypothesis holds given the observed data sample $X$
- $P(H)$: prior probability of hypothesis $H$ (i.e. the initial probability before we observe any data, reflects the background knowledge)
- $P(X)$: probability that sample data is observed
- $P(X|H)$: probability of observing the sample $X$, given that the hypothesis holds

Informally, this can be written as
\[
\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}
\]

MAP (maximum posteriori) hypothesis
\[
h_{MAP} = \arg \max_{h \in H} P(h|D) = \arg \max_{h \in H} P(D|h) P(h).
\]

Practical difficulty: require initial knowledge of many probabilities, significant computational cost
Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent:
  \[ P(X | C_i) = \prod_{k=1}^{n} P(x_k | C_i) \]

- The product of occurrence of say 2 elements \( x_1 \) and \( x_2 \), given the current class is \( C \), is the product of the probabilities of each element taken separately, given the same class \( P([y_1, y_2], C) = P(y_1, C) \times P(y_2, C) \)

- No dependence relation between attributes

- Greatly reduces the computation cost, only count the class distribution.

- Once the probability \( P(X | C_i) \) is known, assign \( X \) to the class with maximum \( P(X | C_i) \times P(C_i) \)

---

Training dataset

<table>
<thead>
<tr>
<th>age</th>
<th>income</th>
<th>student</th>
<th>credit_rating</th>
<th>buys_computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>30...40</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31...40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31...40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31...40</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>
Naïve Bayesian Classifier: Example

- Compute $P(X|C_i)$ for each class

$$P(\text{age}=“<30” | \text{buys_computer}=“yes” ) = \frac{2}{9} = 0.222$$
$$P(\text{age}=“<30” | \text{buys_computer}=“no” ) = \frac{3}{5} = 0.6$$
$$P(\text{income}=“medium” | \text{buys_computer}=“yes” ) = \frac{4}{9} = 0.444$$
$$P(\text{income}=“medium” | \text{buys_computer}=“no” ) = \frac{2}{5} = 0.4$$
$$P(\text{student}=“yes” | \text{buys_computer}=“yes” ) = \frac{6}{9} = 0.667$$
$$P(\text{student}=“yes” | \text{buys_computer}=“no” ) = \frac{1}{5} = 0.2$$
$$P(\text{credit_rating}=“fair” | \text{buys_computer}=“yes” ) = \frac{6}{9} = 0.667$$
$$P(\text{credit_rating}=“fair” | \text{buys_computer}=“no” ) = \frac{2}{5} = 0.4$$

$X=(\text{age}<=30, \text{income}=\text{medium}, \text{student}=\text{yes}, \text{credit_rating}=\text{fair})$

$$P(X|C_i) : P(X|\text{buys_computer}=“yes” ) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$
$$P(X|\text{buys_computer}=“no” ) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$$
$$P(X|C_i)*P(C_i) : P(X|\text{buys_computer}=“yes” ) \times P(\text{buys_computer}=“yes” ) = 0.028$$
$$P(X|\text{buys_computer}=“no” ) \times P(\text{buys_computer}=“no” ) = 0.007$$

$X$ belongs to class “buys_computer=“yes”

---

Naïve Bayesian Classifier: Comments

- **Advantages:**
  - Easy to implement
  - Good results obtained in most of the cases

- **Disadvantages**
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
  - E.g., hospitals: patients: Profile: age, family history etc
    Symptoms: fever, cough etc., Disease: lung cancer, diabetes etc
  - Dependencies among these cannot be modeled by Naïve Bayesian Classifier

- **How to deal with these dependencies?**
  - Bayesian Belief Networks
Bayesian Networks

- Bayesian belief network allows a *subset* of the variables conditionally independent
- A graphical model of causal relationships
  - Represents dependency among the variables
  - Gives a specification of joint probability distribution

```
 Nodes: random variables
 Links: dependency
 X,Y are the parents of Z, and Y is the parent of P
 No dependency between Z and P
 Has no loops or cycles
```

Bayesian Belief Network: An Example

The conditional probability table for the variable LungCancer:
Shows the conditional probability for each possible combination of its parents

\[
P(z_1, ..., z_n) = \prod_{i=1}^{n} P(z_i | \text{Parents}(Z_i))
\]
Learning Bayesian Networks

- Several cases
  - Given both the network structure and all variables observable: learn only the CPTs
  - Network structure known, some hidden variables: method of gradient descent, analogous to neural network learning
  - Network structure unknown, all variables observable: search through the model space to reconstruct graph topology
  - Unknown structure, all hidden variables: no good algorithms known for this purpose

- D. Heckerman, Bayesian networks for data mining

---

SVM - Support Vector Machines

- Small Margin
- Large Margin

Support Vectors
Linear Support Vector Machine

Given a set of points $x_i \in \mathbb{R}^n$ with label $y_i \in \{-1,1\}$
The SVM finds a hyperplane defined by the pair $(w,b)$
(where $w$ is the normal to the plane and $b$ is the distance
from the origin)

$$y_i (x_i \cdot w + b) \geq +1 \quad i = 1, ..., n$$

$x$ – feature vector; $b$- bias, $y$- class label, $2/||w||$ - margin
What if the data is not linearly separable?
Project the data to high dimensional space where it is linearly separable and then we can use linear SVM – (Using Kernels)

Non-Linear SVM

Classification using SVM \((w, b)\)

\[ x_i \cdot w + b > 0 \]

In non linear case we can see this as

\[ K(x_i, w) + b > 0 \]

Kernel - Can be thought of as doing dot product in some high dimensional space
Example of Non-linear SVM

Results

Number of Support Vectors: 9 (-ve: 3, +ve: 6)  Total number of points: 60
SVM Related Links

- http://svm.dcs.rhbnc.ac.uk/
- http://www.kernel-machines.org/
- SVM\text{\textsuperscript{light}} – Software (in C) http://ais.gmd.de/~thorsten/svm\_light
- BOOK: An Introduction to Support Vector Machines
  N. Cristianini and J. Shawe-Taylor
  Cambridge University Press