Bayes Classifiers

• A formidable and sworn enemy of decision trees

Diagram:
- Input Attributes
- Classifier
- Prediction of categorical output

Symbols:
- DT
- BC
Bayes Classifier

Basic idea

• Assume you want to predict output \( y \) which has \( n_Y \) possible values \( v_1, v_2, \ldots v_{n_Y} \).
• Assume there are \( m \) input attributes \( \mathbf{x} = (x_1, x_2, \ldots x_m) \) – let’s assume they are discrete for now.
• Given an example \( (x_1 = u_1, x_2 = u_2, \ldots x_m = u_m) \), predict the value of \( y \) that has the highest value of \( P(y=v_i \mid x_1, x_2, \ldots x_m) \).

• How do we compute \( P(y=v_i \mid x_1, x_2, \ldots x_m) \)?
• Use Bayes rule:

\[
P(y = v_i \mid x_1 = u_1, x_2 = u_2, \ldots, x_m = u_m) \\
= \frac{P(x_1 = u_1, x_2 = u_2, \ldots, x_m = u_m \mid y = v_i)P(y = v_i)}{P(x_1 = u_1, x_2 = u_2, \ldots, x_m = u_m)}
\]
Recipe for a Bayes Classifier

• Given a set of training examples $S = \{(x,y)\}_i$

• Learn a conditional distribution of $p(x|y)$ for each possible $y$ value, $y = v_1, v_2, \ldots v_n$, we do this by:
  
  – Break training set into $n_y$ subsets called $S_1, S_2, \ldots S_n$ based on the $y$ values, i.e., $S_i$ contains examples whose $y=v_i$

  – For each $S_i$, learn a joint distribution of the input features: $p(x|y=v_i) = p(x_1, x_2, \ldots x_m | y=v_i ),$ e.g.,

  $$P(x_1 = u_1, x_2 = u_2, \ldots, x_m = u_m | y = v_i) = \frac{\text{# of } (x_1 = u_1, x_2 = u_2, \ldots, x_m = u_m) \text{ training examples in } S_i}{\text{# of total training examples in } S_i}$$

• Learn $P(y=v_i)$ for all $v_i$

  $$P(y = v_i) = \frac{|S_i|}{|S|}$$
Compute the class probability:

\[
P(y = v_i \mid x_1 = u_1 \cdots x_m = u_m) = \frac{P(x_1 = u_1 \cdots x_m = u_m \mid y = v_i)P(y = v_i)}{P(x_1 = u_1 \cdots x_m = u_m)}
\]

Note: If we need to compute the exact value of \( P(Y = v \mid X_1 = u_1 \cdots X_m = u_m) \) we need to compute the denominator.

However, if we just need to find the most probable \( y \) for prediction, you can skip the denominator, because it is the same for different \( y \) values.
Bayes Classifiers in a nutshell

1. Learn the $P(x_1, x_2, \ldots x_m \mid y=v_i)$ for each value $v_i$
3. Estimate $P(y=v_i)$ as fraction of records with $y=v_i$.
4. For a new prediction:

$$y^{\text{predict}} = \arg\max_v P(y = v \mid x_1 = u_1 \cdots x_m = u_m)$$
$$= \arg\max_v P(x_1 = u_1 \cdots x_m = u_m \mid y = v)P(y = v)$$

Estimating the joint distribution of $x_1, x_2, \ldots x_m$ given $y$ can be problematic!
Joint Density Estimator Overfits

- Typically we don’t have enough data to estimate the joint distribution accurately.
- It is common to encounter the following situation:
  - If no training examples have the exact \( x=(u_1, u_2, \ldots, u_m) \), then \( P(x|y=v_i) = 0 \) for all values of \( Y \).

- In that case, what can we do?
  - we might as well guess a random \( y \) based on the prior, i.e., \( p(y) \).
Example: Spam Filtering

• Bag-of-words representation is used for emails ($\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$)

• Assume that we have a dictionary containing all commonly used words and tokens

• We will create one attribute for each dictionary entry
  – E.g., $x_i$ is a binary variable, $x_i = 1$ (0) means the $i$th word in the dictionary is (not) present in the email
  – Other possible ways of forming the features exist, e.g., $x_i =$ the # of times that the $i$th word appears, but different probability model will be needed

• Assume that our vocabulary contains 10k commonly used words --- we have 10,000 attributes

• How many parameters that we need to learn?

\[
2 \times (2^{10,000} - 1)
\]

2 classes Parameters for each joint distribution $p(\mathbf{x}|y)$
• Clearly we don’t have enough data to estimate that many parameters

• What can we do?
  – Make some bold assumptions to simplify the joint distribution
The Naïve Bayes Assumption

• Assume that each attribute is independent of any other attributes given the class label

\[
P(x_1 = u_1 \cdots x_m = u_m \mid y = v_i)
= P(x_1 = u_1 \mid y = v_i) \cdots P(x_m = u_m \mid y = v_i)
\]
A note about independence

• Assume A and B are Boolean Random Variables. Then
  “A and B are independent”
  if and only if
  \[ P(A|B) = P(A) \]

• “A and B are independent” is often notated as
  \[ A \perp B \]
Independence Theorems

• Assume \( P(A|B) = P(A) \)
• Then \( P(A^\cap B) = \)

= \( P(A) P(B) \)

• Assume \( P(A|B) = P(A) \)
• Then \( P(B|A) = \)

= \( P(B) \)
Independence Theorems

- Assume \( P(A|B) = P(A) \)
- Then \( P(\sim A|B) = \)
  \[ = P(\sim A) \]

- Assume \( P(A|B) = P(A) \)
- Then \( P(A|\sim B) = \)
  \[ = P(A) \]
Examples of independent events

• Two separate coin tosses
• Consider the following four variables:
  – T: Toothache (I have a toothache)
  – C: Catch (dentist’s steel probe catches in my tooth)
  – A: Cavity
  – W: Weather
  – P(T, C, A, W) =?
Conditional Independence

- \( P(x_1|x_2,y) = P(x_1|y) \)
  - \( x_1 \) and \( x_2 \) are conditionally independent given \( y \)
- If \( X_1 \) and \( X_2 \) are conditionally independent given \( y \), then we have
  - \( P(X_1,X_2|y) = P(X_1|y) P(X_2|y) \)
Example of conditional independence

- T: Toothache (I have a toothache)
- C: Catch (dentist’s steel probe catches in my tooth)
- A: Cavity

T and C are conditionally independent given A: \( P(T, C|A) = P(T|A) \cdot P(C|A) \)

So, events that are not independent from each other might be conditionally independent given some fact.

It can also happen the other way around. Events that are independent might become conditionally dependent given some fact.

B=Burglar in your house; A = Alarm (Burglar) rang in your house
E = Earthquake happened
B is independent of E (ignoring some minor possible connections between them)
However, if we know A is true, then B and E are no longer independent. Why?
P(B|A) >> P(B|A, E) Knowing E is true makes it much less likely for B to be true
Naïve Bayes Classifier

• By assuming that each attribute is independent of any other attributes given the class label, we now have a Naïve Bayes Classifier

• Instead of learning a joint distribution of all features, we learn \( p(x_i|y) \) separately for each feature \( x_i \)

• Everything else remains the same
Naïve Bayes Classifier

• Assume you want to predict output \( y \) which has \( n_y \) values \( v_1, v_2, \ldots v_{n_y} \).

• Assume there are \( m \) input attributes called \( \mathbf{x} = (x_1, x_2, \ldots x_m) \).

• Learn a conditional distribution of \( p(\mathbf{x}|y) \) for each possible \( y \) value, \( y = v_1, v_2, \ldots v_{n_y} \), we do this by:
  - Break training set into \( n_y \) subsets called \( S_1, S_2, \ldots S_{n_y} \) based on the \( y \) values, i.e., \( S_i \) contains examples in which \( y=v_i \)
  - For each \( S_i \), learn \( p(y=v_i)=|S_i|/|S| \)
  - For each \( S_i \), learn the conditional distribution each input features, e.g.:

\[
P(x_1 = u_1 \mid y = v_i), \ldots, P(x_m = u_m \mid y = v_i)
\]

\[
y^{\text{predict}} = \arg\max_{v} P(x_1 = u_1 \mid y = v) \cdots P(x_m = u_m \mid y = v) P(y = v)
\]
Example

Apply Naïve Bayes, and make prediction for (1,0,1)?

1. Learn the prior distribution of y.
   \( P(y=0)=1/2, \ P(y=1)=1/2 \)

2. Learn the conditional distribution of \( x_i \) given y for each possible y values
   \( p(X_1|y=0), \ p(X_1|y=1) \)
   \( p(X_2|y=0), \ p(X_2|y=1) \)
   \( p(X_3|y=0), \ p(X_3|y=1) \)

For example, \( p(X_1|y=0) \):
   \( P(X_1=1|y=0)=2/3, \ P(X_1=1|y=1)=0 \)

To predict for (1,0,1):
\[
P(y=0|(1,0,1)) = \frac{\text{P}((1,0,1)|y=0)\text{P}(y=0)}{\text{P}((1,0,1))}
\]
\[
P(y=1|(1,0,1)) = \frac{\text{P}((1,0,1)|y=1)\text{P}(y=1)}{\text{P}((1,0,1))}
\]
Laplace Smoothing

- With the Naïve Bayes Assumption, we can still end up with zero probabilities
- E.g., if we receive an email that contains a word that has never appeared in the training emails
  - $P(x|y)$ will be 0 for all $y$ values
  - We can only make prediction based on $p(y)$
- This is bad because we had ignore all the other words in the email because of this single rare word
- Laplace smoothing can help
  $P(X_1 = 1 | y = 0) = (1 + \# \text{ of examples with } y = 0, X_1 = 1) / (k + \# \text{ of examples with } y = 0)$
  \[ k = \text{the total number of possible values of } x \]
- For a binary feature like above, $p(x|y)$ will be $\frac{1}{2}$
Final Notes about (Naïve) Bayes Classifier

• Any density estimator can be plugged in to estimate $P(x_1, x_2, \ldots, x_m | y)$, or $P(x_i | y)$ for Naïve bayes

• Real valued attributes can be modeled using simple distributions such as Gaussian (Normal) distribution

• Naïve Bayes is wonderfully cheap and survives tens of thousands of attributes easily
Bayes Classifier is a **Generative Approach**

- Generative approach:
  - Learn $p(y)$, $p(x|y)$, and then apply Bayes rule to compute $p(y|x)$ for making predictions.
  - This is equivalent to assuming that each data point is generated following a *generative process* governed by $p(y)$ and $p(X|y)$.
• Generative approach is just one type of learning approaches used in machine learning
  – Learning a correct generative model is difficult
  – And sometimes unnecessary

• KNN and DT are both what we call discriminative methods
  – They are not concerned about any generative models
  – They only care about finding a good discriminative function
  – For KNN and DT, these functions are deterministic, not probabilistic

• One can also take a probabilistic approach to learning discriminative functions
  – i.e., Learn $p(y|X)$ directly without assuming $X$ is generated based on some particular distribution given $y$ (i.e., $p(X|y)$)
  – Logistic regression is one such approach