Text Categorization (Classification)

- Automatically assign documents to a predefined number of categorizations

- Two phases
  - Learning classification knowledge/rules
  - Classifying documents using knowledge learned
Learning Classification Knowledge

• Given:
  – a set of training examples (documents)
  – each training example is a vector of features (term weights)
  – each training example is associated with its class membership

• Output:
  – classification rules which correctly classify most of the training examples and are able to classify unseen examples
Applications of Categorization

- learn to classify web pages by topic
- learn which news articles are of interest
- learn to dispatch/sort emails
Classification Algorithm

• Naïve Bayes
• Decision Trees
• Near Neighbor
• Neural Network
• Rule Induction
Naïve Bayes Classifier

Bayes’ Rule

\[ P(H \land E) = P(H|E)P(E) = P(E|H)P(H) \]

\[ P(H|E) = \frac{P(E|H)P(H)}{P(E)} \]
Naïve Bayes Classifier

Assume that the example is \( x = (x_1, x_2, \ldots, x_n) \) and there are \( m \)
classes \( c_1, c_2, \ldots, c_m \).

Compute the following conditional probabilities:

\[
P(c_1|x_1, x_2, \ldots, x_n)
\]

\[
P(c_2|x_1, x_2, \ldots, x_n)
\]

\[
P(c_m|x_1, x_2, \ldots, x_n)
\]

\( x \) is assigned to the class \( c_i \) with the largest \( P(c_i|x_1, x_2, \ldots, x_n) \).
Naïve Bayes Classifier

According to the Bayes’ rule

\[ P(c_i|x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n | c_i) * P(c_i)}{P(x_1, x_2, \ldots, x_n)} \]

Since \( P(x_1, x_2, \ldots, x_n) \) is the same for all \( c_i \) (\( i = 1, 2, \ldots, m \)), we can ignore \( P(x_1, x_2, \ldots, x_n) \). We need to compute

\[ P(x_1, x_2, \ldots, x_n | c_i) * P(c_i) \]
Naïve Bayes Classifier

Naïve Bayes Classifier assumption:
features are conditionally independent given all classes

\[
P(x_1, x_2, \ldots, x_n) = \prod_{j=1}^{n} P(x_j | c_i)
\]

Naïve Bayes Classifier:

\[
\text{Max}_{i=1,\ldots,m} P(c_i) \prod_{j=1}^{n} P(x_j | c_i)
\]
Naïve Bayes Classifier

Learning Classification Knowledge:

for each class $c_i$, estimate $P(c_i) = N_i/N$
for each class $c_i$ and each feature $x_j$,
estimate $P(x_j | c_i) = N_{ij}/N_i$

$N$ is the total number of training examples
$N_i$ is the number of training examples in class $c_i$
$N_{ij}$ is the number of training examples in class $c_i$ with value $x_j$
Naïve Bayes: Subtleties

Conditional independence assumption is often violated

$$P(x_1, x_2, \ldots, x_n) = \prod_{j=1}^{n} P(x_j | c_i)$$

but it works very well on many applications. It doesn’t need the conditional probabilities correct, it only needs their orders are correct.
Naïve Bayes: Subtleties

If none of training examples in class $c_i$ doesn’t have the value $x_j$, $P(x_j | c_i) = 0$ and $P(c_i) \prod_{j=1}^{n} P(x_j | c_i) = 0$

To solve the problem:

$P(x_j | c_i) = (N_{ij} + 1)/(N_i + m)$

$m$ is a weight larger than 1 and it depends on applications.
Learning to Classify Text

Each document is represented as a list of terms $t_j$.

Assume that we have $N$ documents, $N_i$ documents in $c_i$, $V$ is the number of distinct terms in all documents, $n$ is the total number of occurrences of all terms in all documents, $n_j$ is the total number of occurrences of term $t_j$ in all documents.

$$P(c_i) = \frac{N_i}{N}$$

$$P(t_j | c_i) = \frac{n_j + 1}{n + V}$$