Text Categorization (cont.)

Naïve Bayes Classifiers
Text Categorization: attributes

• Representations of text are very high dimensional (one feature for each word).
• High-bias algorithms that prevent overfitting in high-dimensional space are best.
• For most text categorization tasks, there are many irrelevant and many relevant features.
• Methods that combine evidence from many or all features (e.g. naive Bayes, kNN, neural-nets) tend to work better than ones that try to isolate just a few relevant features (standard decision-tree or rule induction)
Bayesian Methods

• Learning and classification methods based on probability theory.
• Bayes theorem plays a critical role in probabilistic learning and classification.
• Build a generative model that approximates how data is produced.
• 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

\[ P(C, X) = P(C|X)P(X) = P(X|C)P(C) \]

\[ P(C|X) = \frac{P(X|C)P(C)}{P(X)} \]
Naive Bayes Classifiers

Task: Classify a new instance based on a tuple of attribute values

\[ \langle x_1, x_2, \ldots, x_n \rangle \]

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

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

\[ c_{MAP} = \arg \max_{c_j \in C} P(x_1, x_2, \ldots, x_n \mid c_j)P(c_j) \]
Naïve Bayes Classifier: Assumptions

- \( P(c_j) \)
  - Can be estimated from the frequency of classes in the training examples.

- \( P(x_1, x_2, \ldots, x_n | c_j) \)
  - \( O(|X|^n \cdot |C|) \)
  - Could only be estimated if a very, very large number of training examples was available.

Conditional Independence Assumption:

⇒ Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities.
The Naïve Bayes Classifier

- **Conditional Independence Assumption:** features are independent of each other given the class:

\[ P(X_1, \ldots, X_5 \mid C) = P(X_1 \mid C) \cdot P(X_2 \mid C) \cdot \cdots \cdot P(X_5 \mid C) \]
Learning the Model

- Common practice: maximum likelihood
  - simply use the frequencies in the data

\[
\hat{P}(c_j) = \frac{N(C = c_j)}{N}
\]

\[
\hat{P}(x_i | c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}
\]
Problem with Max Likelihood

\[ P(X_1, \ldots, X_5 \mid C) = P(X_1 \mid C) \cdot P(X_2 \mid C) \cdots P(X_5 \mid C) \]

- What if we have seen no training cases where patient had no flu and muscle aches?

\[ \hat{P}(X_5 = t \mid C = \text{nf}) = \frac{N(X_5 = t, C = \text{nf})}{N(C = \text{nf})} = 0 \]

- Zero probabilities cannot be conditioned away, no matter the other evidence!

\[ \ell = \arg \max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c) \]
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

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

# of values of $X_i$

overall fraction in data where $X_i=x_{i,k}$

extent of “smoothing”
Using Naive Bayes Classifiers to Classify Text: Basic method

• Attributes are text positions, values are words.

\[
c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{i} P(x_i | c_j) \\
= \arg\max_{c_j \in C} P(c_j) P(x_1 = "our" | c_j) \cdots P(x_n = "text" | c_j)
\]

• Naive Bayes assumption is clearly violated.
• Still too many possibilities
• Assume that classification is *independent* of the positions of the words (Use same parameters for each position)
Training (learning)

- From training corpus, extract \textit{Vocabulary}
- Calculate required $P(c_j)$ and $P(x_k \mid c_j)$ terms
  - For each $c_j$ in $C$ do
    - $\text{docs}_j \leftarrow$ subset of documents for which the target class is $c_j$
    - $P(c_j) \leftarrow \frac{|\text{docs}_j|}{|\text{total \# documents}|}$
    - $\text{Text}_j \leftarrow$ single document containing all $\text{docs}_j$
    - for each word $x_k$ in \textit{Vocabulary}
      - $n_k \leftarrow$ number of occurrences of $x_k$ in $\text{Text}_j$
      - $P(x_k \mid c_j) \leftarrow \frac{n_k + 1}{n + |\text{Vocabulary}|}$
Testing (Classifying)

- **positions** ← all word positions in current document which contain tokens found in *Vocabulary*
- Return $c_{NB}$, where

$$c_{NB} = \underset{c_j \in C}{\arg\max} \ P(c_j) \ \prod_{i \in \text{positions}} P(x_i \mid c_j)$$