Multinomial Naïve Bayes for Text Categorization

School of Information Studies
Syracuse University

Naïve Bayes

Many Naïve Bayes (NB) models

Two common NB models for text classification

- Multinomial model (use word frequency)
- Benoulli model (use word presence/absence)

Multinomial Naïve Bayes

Pseudo code for MNB in the book *Machine Learning* by Tom Mitchell

Multinomial Naïve Bayes

LEARN_NAIVE_BAYES_TEXT($Examples$, V)

Examples is a set of text documents along with their target values. V is the set of all possible target values. This function learns the probability terms $P(w_k|v_j)$, describing the probability that a randomly drawn word from a document in class v_j will be the English word w_k. It also learns the class prior probabilities $P(v_j)$.

1. collect all words, punctuation, and other tokens that occur in $Examples$

- $Vocabulary \leftarrow$ the set of all distinct words and other tokens occurring in any text document from $Examples$

2. calculate the required $P(v_j)$ and $P(w_k|v_j)$ probability terms

- For each target value v_j in V do
  - $docs_j \leftarrow$ the subset of documents from $Examples$ for which the target value is v_j
  - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  - $Text_j \leftarrow$ a single document created by concatenating all members of $docs_j$
  - $n \leftarrow$ total number of distinct word positions in $Text_j$
  - for each word w_k in $Vocabulary$
    - $n_k \leftarrow$ number of times word w_k occurs in $Text_j$
    - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

CLASSIFY_NAIVE_BAYES_TEXT(Doc)

Return the estimated target value for the document Doc. a_i denotes the word found in the ith position within Doc.

- $positions \leftarrow$ all word positions in Doc that contain tokens found in $Vocabulary$
- Return v_{NB}, where

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_{i \in positions} P(a_i|v_j)$$

Bayesian Rule

$P(X, \text{class})$

$= P(X|\text{class}) * P(\text{class})$

$= P(\text{class}|X) * P(X)$

Our prediction goal

$P(\text{class}|X) = P(X|\text{class}) * P(\text{class})/P(X)$

Prior and Conditional Probabilities

Prior probability: P(class)

Conditional probability: P(X|class)

Both can be estimated from training data

Posterior Probability

Posterior probability: P(class|X)

Calculated based on prior and conditional probabilities using Bayes rules

$P(class|X) = P(X|class) * P(class)/P(X)$

Ignore P(X) because we just need to find the highest posterior

Naïve?

Why is this algorithm called “naïve” Bayes?

Because it assumes the occurrence of each word is independent of the occurrence of other words, which is oftentimes not true in text data.

$P(X|class)$

$= P(w_1|class) \times P(w_2|class) \times \ldots \times P(w_n|class)$

The Independence Assumption

Not true for natural language!

Still works quite well on a number of text classification tasks

- Newsgroup classification (Mitchell 1997)
- Movie review classification (Pang et al., 2002)

Theoretical explanation

- Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine learning, 29(2–3), 103–130.

Smoothing for Multinomial NB

$$P\left(w_k \middle| v_j\right) \leftarrow \frac{n_k + 1}{n + |Vocabulary|}$$

What Is Stored in a Trained MNB Model?

A look-up table of probabilities

	Class = 1	Class = 2	Class = …
P(class)	0.40	0.60	
$P(w_1 \| class)$	0.75	0.50	
$P(w_2 \| class)$	0.25	0.67	
$P(w_3 \| class)$	0.33	0.50	
$P(w_4 \| class)$	0.80	0.33	
…			
$P(w_n \| class)$	…	…	