BUILDING A SPAM FILTER USING NAIÇE BAYES

Review: Bayes’ Rule & Diagnosis

\[ P(a|b) = \frac{P(b|a) \cdot P(a)}{P(b)} \]

- Useful for assessing diagnostic probability from causal probability:

\[ P(\text{Cause}|\text{Effect}) = \frac{P(\text{Effect}|\text{Cause}) \cdot P(\text{Cause})}{P(\text{Effect})} \]

Review: Bayes’ Rule For Diagnosis II

\[ P(\text{Disease} | \text{Symptom}) = \frac{P(\text{Symptom} | \text{Disease}) \cdot P(\text{Disease})}{P(\text{Symptom})} \]

Imagine:
- disease = TB, symptom = coughing
- \( P(\text{disease} | \text{symptom}) \) is different in TB-indicated country vs. USA
- \( P(\text{symptom} | \text{disease}) \) should be the same
  - It is more widely useful to learn \( P(\text{symptom} | \text{disease}) \)
- What about \( P(\text{symptom})? \)
  - Last time: Use conditioning
  - For determining, e.g., the most likely disease given the symptom, we can just ignore \( P(\text{symptom})!!! \) (Coming up: Slide 11)

Review: Naïve Bayes I

By Bayes Rule

\[ P(C|T, X) = \frac{P(T, X|C)P(C)}{P(T, X)} \]

If \( T \) and \( X \) are conditionally independent given \( C \):

\[ P(C|T, X) = \frac{P(T|C)P(X|C)P(C)}{P(T, X)} \]

This is a Naïve Bayes Model:
- All effects assumed conditionally independent given Cause

Review: Bayes’ Rule II

- More generally, if \( \text{Effect} \) are conditionally independent given \( \text{Cause} \):

\[ P(\text{Cause}, \text{Effect}_1, ..., \text{Effect}_n) = P(\text{Cause}) \prod P(\text{Effect}_i | \text{Cause}) \]

- And total number of parameters is linear in \( n \)

Spam or not Spam: that is the question.

From: "clarkworld@hotmail.com"
Subject: real estate is the only way... gem oahykgay

Anyone can buy real estate with no money down
Stop paying rent TODAY!
There is no need to spend hundreds or even thousands for similar courses
I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.
Change your life NOW!

====================================================================
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm
====================================================================
Categorization/Classification Problems
• Given:
  • A description of an instance, \( x \in X \), where \( X \) is the instance language or instance space.
    — (Important Issue: how do we represent text documents?)
  • A fixed set of categories:
    \[ C = \{ c_1, c_2, \ldots, c_n \} \]
• To determine:
  • The category of \( x \): \( c(x) \in C \), where \( c(x) \) is a categorization function whose domain is \( X \) and whose range is \( C \).
    — We want to automatically build categorization functions (“classifiers”).

EXAMPLES OF TEXT CATEGORIZATION
• Categories = SPAM?
  • “spam” / “not spam”
• Categories = TOPICS
  • “finance” / “sports” / “asia”
• Categories = OPINION
  • “like” / “hate” / “neutral”
• Categories = AUTHOR
  • “Shakespeare” / “Marlowe” / “Ben Jonson”
  • The Federalist papers

A Graphical View of Text Classification

Bayesian Methods for Classification
• Uses Bayes theorem to build a generative model that approximates how data is produced.
• First step:
  \[
  P(C \mid X) \propto P(X 
  \mid C) P(C) 
  \]
  Where \( C \): Categories, \( X \): Instance to be classified
• 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 each instance.

Maximum a posteriori (MAP) Hypothesis
• Let \( c_{MAP} \) be the most probable category. Then goodbye to that nasty normalization!!
  \[
  c_{MAP} = \arg \max_{c \in C} P(c \mid X) 
  = \arg \max_{c \in C} \frac{P(X \mid c) P(c)}{P(X)} 
  = \arg \max_{c \in C} P(X \mid c) P(c) 
  \]
  No need to compute \( P(X) \)!!

Maximum likelihood Hypothesis
If all hypotheses are a priori equally likely, to find the maximally likely category \( c_{ML} \), we only need to consider the \( P(X \mid c) \) term:
  \[
  c_{ML} = \arg \max_{c \in C} P(X \mid c) 
  \]
  Maximum Likelihood Estimate (“MLE”)
Naïve Bayes Classifiers: Step 1

Assume that instance $x$ described by $n$-dimensional vector of attributes $x = \{x_1, x_2, \ldots, x_n\}$

then

$$c_{\text{MAP}} = \arg\max_{c \in \mathcal{C}} P(c \mid x_1, x_2, \ldots, x_n)$$

$$= \arg\max_{c \in \mathcal{C}} \frac{P(x_1, x_2, \ldots, x_n \mid c) P(c)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \arg\max_{c \in \mathcal{C}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

Naïve Bayes Classifiers: Step 2

To estimate:

$$c_{\text{MAP}} = \arg\max_{c \in \mathcal{C}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

- $P(c)$: Can be estimated from the frequency of classes in the training examples.
- $P(x_1, x_2, \ldots, x_n \mid c)$: Problem!!
  - $O(\mathcal{X}^n \cdot |\mathcal{C}|)$ parameters required to estimate full joint probability distribution

Solution:

**Naïve Bayes Conditional Independence Assumption:**

$$P(x_1, x_2, \ldots, x_n \mid c_j) = \prod_i P(x_i \mid c_j)$$

Learning the Model

- First attempt: maximum likelihood estimates
  - Given training data for $N$ individuals, where count($X=x$) is the number of those individuals for which $X=x$, e.g. Flu = true
  - For each category $c$ and each value $x$ for a variable $X$
    - $\hat{P}(c) = \frac{\text{count}(C=c)}{N}$
    - $\hat{P}(x | c) = \frac{\text{count}(X=x, C=c)}{\text{count}(C=c)}$

Problem with Max Likelihood for Naïve Bayes

- What if no training cases where patient with flu had a cough?
  - $\hat{P}(X_1 = \text{cough} | \text{flu}) = \frac{\text{count}(X_1 = \text{cough}, \text{flu})}{\text{count}(\text{flu})} = 0$

  So if $X_1 = \text{cough}$, $P(X_1 = \text{cough}, \ldots, x_n | \text{flu}) = 0$

  Zero probabilities overwhelm any other evidence!

"Add-1" Laplace Smoothing to Avoid Overfitting

$$\hat{P}(x | c) = \frac{\text{count}(X = x, C = c) + 1}{\text{count}(C = c) + |\mathcal{X}|}$$

- Slightly better version

$$\hat{P}(x | c) = \frac{\text{count}(X = x, C = c) + \alpha}{\text{count}(C = c) + \alpha \cdot |\mathcal{X}|}$$

# of values of $X_i$ here 2

extent of "smoothing"
Using Naive Bayes Classifiers to Classify Text: Basic method for Multinomial Variables

- As a generative model:
  1. Randomly pick a category \( c \) according to \( P(c) \)
  2. For a document of length \( N \), for each word \( w \):
     1. Generate \( w | c \) according to \( P(w | c) \)

\[
P(c, D = \{w_1, w_2, ..., w_N\}) = P(c) \prod_{i=1}^{N} P(w_i | c)
\]

- This is a Naive Bayes classifier for multinomial variables.
- Note that word order is assumed irrelevant here
  - Uses same parameters for each position
  - Result is bag of words model
  - Views document not as an ordered list of words, but as a multiset

Naive Bayes: Learning (First attempt)

- From training corpus, extract Vocabulary
- Calculate required estimates of \( P(c) \) and \( P(w_i | c) \) terms,
  - For each \( c \) in \( C \)
    \[
    P(c) = \frac{\text{count}_{\text{ docs}(c)} + \alpha}{\text{count}_{\text{ docs}(C)} + |V| + \alpha}
    \]
    where \( \text{count}_{\text{ docs}(x)} \) is the number of documents for which \( x \) is true.
  - For each word \( w \) in Vocabulary and \( c \) in \( C \), where \( \text{count}_{\text{ docs}(w | c)} \) is the number of tokens over all documents for which \( x \) is true of that document and that token,
    \[
    P(w | c) = \frac{\text{count}_{\text{ docs}(w | c)} + \alpha}{\text{count}_{\text{ docs}(c)} + |V| + \alpha}
    \]

Naive Bayes: Learning (Second attempt)

- Laplace smoothing must be done over the vocabulary items.
  - We can assume we have at least one instance of each category, so we don’t need to smooth these.
  - Assume a single new word UNK, that occurs nowhere within the training document set.
  - Map all unknown words in documents to be classified (test documents) to UNK.
  - with \( 0 \leq \alpha \leq 1 \)

\[
P(w | c) = \frac{\text{count}_{\text{ docs}(w | c)} + \alpha}{\text{count}_{\text{ docs}(c)} + |V| + \alpha}
\]

Naive Bayes: Classifying

- Compute \( c_{NB} \) using either
  - \( c_{NB} = \arg \max_c P(c) \prod_{i=1}^{N} P(w_i | c) \)
  - \( c_{NB} = \arg \max_c P(c) \prod_{i=0}^{\text{count}(w)} P(w_i | c) \)

\( \text{count}(w) \): the number of times word \( w \) occurs in doc

(The two are equivalent.)

PANTEL AND LIN: SPAMCOP

- Uses a Naive Bayes classifier
- \( M \) is spam if \( P(\text{Spam} | M) > P(\text{NonSpam} | M) \)

Method
- Tokenize message using Porter Stemmer
- Estimate \( P(x | c) \) using m-estimate (a form of smoothing)
- Remove words that do not satisfy certain conditions
- Train: 160 spams, 466 non-spams
- Test: 277 spams, 346 non-spams

Results: ERROR RATE of 4.33%
  - Worse results using trigrams

Naive Bayes is (was) Not So Naive

- Naive Bayes: First and Second place in KDD-CUP 97 competition, among 16 (then) state of the art algorithms
  - Goal: Financial services industry direct mail response prediction model: Predict if the recipient of mail will actually respond to the advertisement – 750,000 records.
  - A good dependable baseline for text classification
    - But not the best by itself.
  - Optimal if the Independence Assumptions hold:
    - If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
  - Very Fast:
    - Learning with one pass over the data.
    - Testing linear in the number of attributes, and document collection size
  - Low Storage requirements
Engineering: Underflow Prevention

- Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow.
- Since $\log(xy) = \log(x) + \log(y)$, it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.
- Class with highest final un-normalized log probability score is still the most probable.

\[
c_{NB} = \arg\max_{c_i \in C} \log P(c_j) + \sum_{i \in \text{positions}} \log P(w_i | c_j)
\]

REFERENCES