Bayes Theorem & Naïve Bayes

(some slides adapted from slides by Massimo Poesio, adapted from slides by Chris Manning)

Review: Bayes' Theorem & Diagnosis

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

- Useful for assessing diagnostic probability from causal probability:

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

The Solution: Independence

- Random variables A and B are independent iff
  \- \( P(A \cap B) = P(A) \cdot P(B) \)
  \- equivalently: \( P(A | B) = P(A) \) and \( P(B | A) = P(B) \)
- A and B are independent if knowing whether A occurred gives no information about B (and vice versa)
- Independence assumptions are essential for efficient probabilistic reasoning

Conditional Independence

- BUT absolute independence is rare
- Dentistry is a large field with hundreds of variables, none of which are independent. What to do?
- A and B are conditionally independent given C iff
  \- \( P(A | B, C) = P(A | C) \)
  \- \( P(B | A, C) = P(B | C) \)
  \- \( P(A \land B | C) = P(A | C) \cdot P(B | C) \)

- Toothache (T), Spot in Xray (X), Cavity (C)
  \- None of these are independent of the other two
  \- But T and X are conditionally independent given C

Conditional Independence II WHY??

- If I have a cavity, the probability that the XRay shows a spot doesn't depend on whether I have a toothache (and vice versa):
  \[ P(X|T,C) = P(X|C) \]
- From which follows:
  \[ P(T|X,C) = P(T|C) \] and \( P(T|X,C) = P(T|C) \cdot P(X|C) \)
- By the chain rule, given conditional independence:
  \[ P(T,X,C) = P(T|X,C) \cdot P(X,C) = P(T|C) \cdot P(X|C) \cdot P(C) \]
  \[ = P(T|C) \cdot P(X|C) \cdot P(C) \]
- \( P(\text{Toothache, Cavity, Xray}) \) has \( 2^3 - 1 \) = 7 independent entries
- Given conditional independence, chain rule yields
  \[ 2 \times 2 + 1 = 5 \] independent numbers

Review: Estimating joint probability distributions is often intractable

- Estimating the necessary joint probability distribution for many symptoms is intractable
  \- For \( |D| \) diseases, \( |S| \) symptoms where a person can have \( n \) of the diseases and \( m \) of the symptoms
    \- \( P(s|d_1, d_2, \ldots, d_n) \) requires \( |S| \cdot |D| \cdot n \) values
    \- \( P(s_1, s_2, \ldots, s_m) \) requires \( |S|^m \) values
- These numbers get big fast
  \- If \( |S| = 1000, |D| = 100, n = 4, m = 7 \)
    \- \( P(s_1, s_2, \ldots, s_m) \) requires \( 1000^7 \approx 10^{11} \) values
- 15 entries (\( 2^3 - 1 \)) reduced to 8 (\( 2^3 - 1 + 2 \))
  \- For \( n \) independent biased coins, \( O(2^n) \) entries \( \rightarrow O(n) \)
Conditional Independence III

- In most cases, the use of conditional independence reduces the size of the representation of the joint distribution from exponential in n to linear in n.
- Conditional independence is our most basic and robust form of knowledge about uncertain environments.

Naïve Bayes I

By Bayes Theorem

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

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

Computing the Normalizing Constant \( P(T,X) \)

\[ P(e|T,X) + P(\neg e|T,X) = 1 \]

\[ \frac{P(T|e)P(X|e)P(e)}{P(T,X)} + \frac{P(T|\neg e)P(X|\neg e)P(\neg e)}{P(T,X)} = 1 \]

\[ P(T|e)P(X|e)P(e) + P(T|\neg e)P(X|\neg e)P(\neg e) = P(T,X) \]

Bayes' Theorem II

- More generally, if Effect, are conditionally independent given Cause:

\[ P(Cause, Effect_1,..., Effect_n) = P(Cause) \prod_i P(Effect_i | Cause) \]

- The total number of parameters is linear in n

Spam or not Spam: that is the question.

From: **<clarkworld@gmail.com>**
Subject: real estate is the only way... gem oalykgay

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

2. BUILDING A SPAM FILTER USING NAÏVE BAYES
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
  - The Federalist papers: James Madison, Alexander Hamilton, or John Jay

Bayesian Methods for Classification

- Uses Bayes theorem to build a generative model that approximates how data is produced.
- First step:
  - 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 a particular instance.

Maximum a posteriori (MAP) Hypothesis

- Let \( c_{\text{MAP}} \) be the most probable category. Then goodbye to that nasty normalization!!
  
  \[
  c_{\text{MAP}} = \arg \max_{c \in C} P(X \mid c) P(c) \bigg/ P(X)
  \]

  Where: 
  - \( C \): Categories
  - \( X \): Instance to be classified

  Maximum Likelihood Estimate ("MLE")

Naive Bayes Classifiers: Step 1

Assume that instance \( x \) described by \( n \)-dimensional vector of attributes \( X = (x_1, x_2, \ldots, x_n) \) then

\[
\begin{align*}
c_{\text{MAP}} & = \arg \max_{c \in C} P(c \mid x_1, x_2, \ldots, x_n) \\
& = \arg \max_{c \in C} P(x_1, x_2, \ldots, x_n \mid c) P(c) \\
& = \arg \max_{c \in C} P(x_1, x_2, \ldots, x_n \mid c) P(c) \\
& = \frac{P(x_1, x_2, \ldots, x_n \mid c) P(c)}{P(x_1, x_2, \ldots, x_n)}
\end{align*}
\]
Naïve Bayes Classifier: Step 2

To estimate: \( \hat{c}_{\text{MAP}} = \arg \max_c P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j) \)

- \( P(c_j) \): Can be estimated from the frequency of classes in the training examples.
- \( P(x_p, x_2, \ldots, x_n \mid c_j) \): Problem!!
  - \( O(|X| \times |C|) \) parameters required to estimate full joint prob. distribution

Solution:

Naïve Bayes Conditional Independence Assumption:

\[
\hat{P}(x_1, x_2, \ldots, x_n \mid c_j) = \prod_i \hat{P}(x_i \mid c_j)
\]

Learning the Model

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

“Add-1” Laplace Smoothing to Avoid Overfitting

\[
\hat{P}(x \mid c) = \frac{\text{count}(X=x, C=c) + 1}{\text{count}(C=c) + \alpha |X|}
\]

- Slightly better version

Using Naïve 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 \( (1 \leq i \leq N) \):
     1. Generate \( w_i \) according to \( P(w_i \mid c) \)

\[
P(c, D = w_1, w_2, \ldots, w_n) = P(c) \prod_i P(w_i \mid c)
\]

- This is a Naïve Bayes classifier for multinomial variables.
- Note that word order is assumed irrelevant to estimates
  - 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

Problem with Max Likelihood for Naïve Bayes

- \( P(X_i, \ldots, X_n \mid \text{Flu}) = P(X_i \mid \text{Flu}) \cdot P(X_2 \mid \text{Flu}) \cdot \ldots \cdot P(X_n \mid \text{Flu}) \)
- What if no training cases where patient with flu had a cough?

\[
\hat{P}(X_i = t \mid \text{flu}) = \frac{\text{count}(X_i = t, \text{flu})}{\text{count}(\text{flu})} = 0
\]

So if \( X_i = t \), \( P(X_i, \ldots, X_n \mid \text{flu}) = 0 \)

Zero probabilities overwhelm any other evidence!
Naïve Bayes: Learning (First attempt)

- Extract \( V \), the vocabulary set of all words within the training corpus.
- Calculate required estimates of \( P(c) \) and \( P(w_c) \) terms,
  - For each \( c \in C \), calculate
    \[
    \hat{P}(c) = \frac{\text{count}(c)}{\text{total number of docs}}
    \]
    where \( \text{count}(c) \) is the number of docs in \( c \) and \( \text{total number of docs} \) is the total number of docs
  - For each word \( w_c \in V \), calculate
    \[
    \hat{P}(w_c | c) = \frac{\text{count}(w_c)}{\text{count}(c)}
    \]
    where \( \text{count}(w_c) \) is the number of tokens in all documents of category \( c \).

Naïve Bayes: Learning (Second attempt)

- Laplace smoothing must be done over \( V \), the set of vocabulary items.
- Idea (simple and works OK):
  - Add an additional fractional count of \( \alpha \) to each word and to a single new word \( \text{UNK} \) (for unknown), that occurs nowhere within \( V \), so our new vocabulary is \( V' = V \cup \{\text{UNK}\} \).
  - Map all unknown words in documents to be classified (test documents) to \( \text{UNK} \).
  - Compute
    \[
    \hat{P}(w|c) = \frac{\text{count}(w)+\alpha}{\text{count}(c)+\alpha|V'|+1}
    \]
    where \( \alpha \), \( 0 \leq \alpha \leq 1 \), is the smoothing constant, and \( \text{count}(w) \) is the count of the word \( w \) in the set of documents of category \( c \).

Naïve Bayes: Classifying

- Compute \( c_{NB} \) using either
  \[
  c_{NB} = \arg \max_c P(c) \prod_{i=1}^{N} P(w_i | c)
  \]
  \[
  c_{NB} = \arg \max_c P(c) \prod_{w \in w} P(w | c)^{\text{count}(w)}
  \]
  where \( \text{count}(w) \) is the number of times word \( w \) occurs in doc.

  (The two are equivalent.)

PANTEL AND LIN: SPAMCOP

- Uses a Naïve Bayes classifier
- M is spam if \( P(\text{Spam}|M) > P(\text{NonSpam}|M) \)
- Method
  - Tokenize message using Porter Stemmer
  - Estimate \( P(w|c) \) 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

Naïve Bayes is (was) Not So Naive

- Naïve: 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 products by summing logs of probabilities rather than multiplying probabilities.
- Class with highest final un-normalized log probability score is still the most probable.

\[
\hat{c}_{NB} = \arg \max_{c \in C} \log P(c_j) + \sum_{i \in \text{positions}} \log P(w_i | c_j)
\]
REFERENCES