Bayes Rule & Naïve Bayes

CIS 391 – Introduction to Artificial Intelligence

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

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

- Posterior

Likelihood \hspace{1cm} Prior

Normalization

- Useful for assessing diagnostic probability from causal probability:

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

For example:

\[ P(disease | symptom) = \frac{P(symptom | disease) \times P(disease)}{P(symptom)} \]
Bayes’ Rule II

\[ P(\text{disease} \mid \text{symptom}) = \frac{P(\text{symptom} \mid \text{disease}) \times P(\text{disease})}{P(\text{symptom})} \]

Imagine:

- disease = TB, symptom = coughing
- \( P(\text{disease} \mid \text{symptom}) \) is different in TB-indicated country vs. USA
- \( P(\text{symptom} \mid \text{disease}) \) should be the same
  - It is more widely useful to learn \( P(\text{symptom} \mid \text{disease}) \)

What about \( P(\text{symptom}) \)?

- Use \textit{conditioning} (next slide)
- For determining, e.g., the most likely disease given the symptom, we can just ignore \( P(\text{symptom}) \)!!! (see slide 21)
Conditioning

- **Idea:** Use *conditional probabilities* instead of joint probabilities

- \[ P(a) = P(a \land b) + P(a \land \neg b) \]
  \[ = P(a \mid b) \cdot P(b) + P(a \mid \neg b) \cdot P(\neg b) \]

  *Here:*

  \[ P(\text{symptom}) = P(\text{symptom} \mid \text{disease}) \cdot P(\text{disease}) + P(\text{symptom} \mid \neg \text{disease}) \cdot P(\neg \text{disease}) \]

- More generally: \( P(Y) = \sum_z P(Y \mid z) \cdot P(z) \)

- Marginalization and conditioning are useful rules for derivations involving probability expressions.
Bayes rule is widely used **BUT**

- Estimating the necessary joint probability distribution is often 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 $|S||D|^n$ values
    - $P(s_1, s_2, \ldots, s_m)$ requires $|S|^m$ values

- These numbers get big fast
  - If $|S| = 1,000$, $|D| = 100$, $n=4$, $m=7$
    - $P(s|d_1, \ldots d_n)$ requires $1000 \times 100^4 = 10^{11}$ values
    - $P(s_1 \ldots s_m)$ requires $1000^7 = 10^{21}$ values
Independence

- Random variables A and B are independent iff
  - $P(A \land B) = P(A) \times P(B)$
  - $P(A \mid B) = P(A)$
  - $P(B \mid A) = P(B)$
- Independence assumptions are essential for efficient probabilistic reasoning

- 32 entries reduced to 12
  For $n$ independent biased coins, $O(2^n)$ entries $\rightarrow O(n)$
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
  - \( P(A \mid B, C) = P(A \mid C) \)
  - \( P(B \mid A, C) = P(B \mid C) \)
  - \( P(A \land B \mid C) = P(A \mid C) \times P(B \mid C) \)

- Toothache (T), Spot in Xray (X), Cavity (C)
  - None of these propositions are independent of one another
  - But \( T \) and \( X \) are conditionally independent given \( C \)
Conditional Independence II

- If I have a cavity, the probability that the XRay shows a spot doesn’t depend on whether I have a toothache:
  \[ P(X|T,C) = P(X|C) \]

- From which follows:
  \[ P(T|X,C) = P(T|C) \]
  \[ P(T,X|C) = P(T|C) \times P(X|C) \]

- By the chain rule, given conditional independence:
  \[ P(T,X,C) = P(T|X,C) \times P(X,C) \]
  \[ = P(T|X,C) \times P(X|C) \times P(C) \]
  \[ = P(T|C) \times P(X|C) \times P(C) \]

- \( P(\text{Toothache, Cavity, Xray}) \) has \( 2^3 - 1 = 7 \) independent entries

- Given conditional independence, chain rule yields
  \[ 2 + 2 + 1 = 5 \] independent numbers
Conditional Independence III

- In most cases, the use of conditional independence reduces the size of the representation of the joint distribution from \textit{exponential in } n \textit{ to linear in } n.

- \textit{Conditional independence is our most basic and robust form of knowledge about uncertain environments.}
Another Example

- Battery is dead (B)
- Radio plays (R)
- Starter turns over (S)
- None of these propositions are independent of one another
- **BUT:** R and S are conditionally independent given B
Combining Evidence

★ Bayesian updating given two pieces of information

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

- Now assume that 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 simple \textit{Naïve Bayes Model}.

- Naïve Bayes models let us combine evidence incrementally and sequentially
Bayes' Rule and conditional independence

\[ P(Cavity \mid \text{toothache} \land xray) = \alpha P(\text{toothache} \land xray \mid Cavity) P(Cavity) \]
\[ = \alpha P(\text{toothache} \mid Cavity) P(xray \mid Cavity) P(Cavity) \]

- This is an example of a naïve Bayes model:

\[ P(\text{Cause}, \text{Effect}_1, \ldots, \text{Effect}_n) = P(\text{Cause}) \prod_i P(\text{Effect}_i \mid \text{Cause}) \]

- Total number of parameters is linear in \( n \)
Computing the Normalizing Constant ($\alpha$)

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

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

\[
P(T|c)P(X|c)P(c) + P(T|\neg c)P(X|\neg c)P(\neg c) = P(T, X)
\]
BUILDING A SPAM FILTER USING NAÏVE BAYES
Spam or not Spam: that is the question.

From: """" <takworlId@hotmail.com>
Subject: real estate is the only way... gem oalvgkay

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*.
    - *(Issue: how do we represent text documents?)*
  - A fixed set of categories:
    \[ C = \{c_1, c_2, \ldots, c_n\} \]

- **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”).*
Document Classification More Generally

**Test Data:**

- ML
- Planning
- Semantics
- Garb.Coll.
- Multimedia
- GUI

**Classes:**

- (AI)
- (Programming)
- (HCI)

**Training Data:**

- learning intelligence algorithm reinforcement network...
- planning temporal reasoning plan language...
- programming semantics language proof...
- garbage collection memory optimization region...

"planning language proof intelligence"
A Graphical View of Text Classification
EXAMPLES OF TEXT CATEGORIZATION

- **LABELS=BINARY**
  - “spam” / “not spam”

- **LABELS=TOPICS**
  - “finance” / “sports” / “asia”

- **LABELS=OPINION**
  - “like” / “hate” / “neutral”

- **LABELS=AUTHOR**
  - “Shakespeare” / “Marlowe” / “Ben Jonson”
  - The Federalist papers
Bayesian Methods for Text Classification

• Uses *Bayes theorem* to build a *generative Naïve Bayes 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 once more

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

\[
P(C | D) = \frac{P(D | C) P(C)}{P(D)}
\]

(C: Categories, D: Documents)
Maximum a posteriori (MAP) Hypothesis

\[ c_{MAP} \equiv \arg\max_{c \in C} P(c \mid D) \]

\[ = \arg\max_{c \in C} \frac{P(D \mid c)P(c)}{P(D)} \]

As \( P(D) \) is constant

No need to compute \( \alpha \), here \( P(D) \)!!!!
Maximum likelihood Hypothesis

If all hypotheses are a priori equally likely, we only need to consider the $P(D|c)$ term:

$$c_{ML} \equiv \arg\max_{c \in C} P(D | c)$$

Maximum Likelihood Estimate ("MLE")
Naive Bayes Classifiers

Task: Classify a new instance $D$ based on a tuple of attribute values $D = \langle x_1, x_2, \ldots, x_n \rangle$ into one of the classes $c_j \in C$

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

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

$$= \arg\max_{c \in C} P(x_1, x_2, \ldots, x_n \mid c)P(c)$$
Naïve Bayes Classifier: Assumption

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

- $P(x_1, x_2, \ldots, x_n | c_j)$
  - Again, $O(|X|^n \cdot |C|)$ parameters to estimate full joint probability distribution
  - As we saw, can only be estimated if a vast number of training examples was available.

Naïve Bayes Conditional Independence Assumption:

$$P(x_1, x_2, \ldots, x_n | c_j) = \prod_i P(x_i | c_j)$$
The Naïve Bayes Classifier

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

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

- This model is appropriate for binary variables
Learning the Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data, given N individuals

\[
\hat{P}(c_j) = \frac{\text{Count}_N(C = c_j)}{|N|}
\]

\[
\hat{P}(x_i | c_j) = \frac{\text{Count}_N(X_i = x_i, C = c_j)}{\text{Count}_N(C = c_j)}
\]
What if we have seen no training cases where patient had no flu and muscle aches?

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

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

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

\[ \ell = \arg \max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c) \]
“Add-1” Laplace Smoothing to Avoid Overfitting

\[
\hat{P}(x_i \mid c_j) = \frac{\text{Count}(X_i = x_i, C = c_j) + 1}{\text{Count}(C = c_j) + |X|}
\]

- Slightly better version

\[
\hat{P}(x_i \mid c_j) = \frac{\text{Count}(X_i = x_i, C = c_j) + \frac{1}{m}}{\text{Count}(C = c_j) + \frac{|X|}{m}}
\]

# of values of \(X_i\)

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

- As a generative model:
  1. Randomly pick a category $c$ according to $P(c)$
  2. For $word_i$ to $word_N$ of the document:
     1. Generate $word_i$ according to $P(w_i|c)$

$$P(c, w_1, w_2, ..., w_n) = P(c)P(w_1 | c)...P(w_n | c)$$

- We assume that classification is independent of the positions of the words
  - Use same parameters for each position
  - Result is bag of words model
    —Views document not as an ordered list of words, but as a multiset
Naïve Bayes: Learning (First attempt)

- From training corpus, extract *Vocabulary*

- Calculate required $P(c_j)$ and $P(w_k \mid c_j)$ terms

  (Notation: $docs_j \leftarrow$ subset of documents for which the target class is $c_j$)

  - For each $c_j$ in $C$ do
    \[ P(c_j) \leftarrow \frac{|docs_j|}{|docs|} \]

  - For each word $w_k$ in *Vocabulary*
    \[ P(w_k \mid c_j) \leftarrow \frac{|\text{tokens of } w_k \text{ in } docs_j|}{|\text{all tokens in } docs_j|} \]
Fix: Naïve Bayes Learning w/ Laplace Smoothing

- From training corpus, extract Vocabulary

- Calculate required $P(c_j)$ and $P(w_k \mid c_j)$ terms
  - For each $c_j$ in $C$ do
    \[ P(c_j) \leftarrow \frac{|docs_j|}{|docs|} \]

- For each word $w_k$ in Vocabulary
  \[ P(w_k \mid c_j) \leftarrow \frac{|\text{tokens of } w_k \text{ in } docs_j| + 1}{|\text{all tokens in } docs_j| + |\text{Vocabulary}|} \]
Naïve Bayes: Classifying

- \( \text{positions}_k \leftarrow \text{all word positions in current document which contain tokens of } w_k \)

- Return \( c_{NB} \), where

\[
c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{k \in \text{Vocab}} P(w_k | c_j)^{|\text{positions}_k|}
\]
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 \( P(x_k|C) \) 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

- Naïve 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.

$$c_{NB} = \arg\max_{c_j \in C} \log P(c_j) + \sum_{i \in \text{positions}} \log P(w_i \mid c_j)$$