DATA, MSML, BIOI 602
Principles of Data Science

Naïve Bayes and Text Classification

Heng Huang
Department of Computer Science

Classification

- **Learn**: h:**X** $\mapsto$ Y
  - **X** – features
  - Y – target classes

- Suppose you know P(Y|**X**) exactly, how should you classify?
  - Bayes classifier:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Which is shorthand for:

$$(\forall i, j) P(Y = y_i | X = x_j) = \frac{P(X = x_j | Y = y_i) P(Y = y_i)}{P(X = x_j)}$$

Ref: Carlos Guestrin

How to Learn the Classifier?

Sky	Temp	Humid	Wind	Water	Forecst	EnjoySpt
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	Yes

- How do we represent these? How many parameters?
  - Prior, P(Y):
    - Suppose Y is composed of *k* classes
  - Likelihood, P(**X**|Y):
    - Suppose **X** is composed of *n* binary features

Ref: Carlos Guestrin

Naive Bayes: Example

<u>Predict:</u> Which tag ("Sports" or "Not sports") does **the sentence** "A very close game" belong to?

P (Sports | a very close game)

P (Not sports | a very close game)

Text	Tag
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

Training Data

Naive Bayes: Example

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

Text	Tag
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

COMPARE:

$$P(sports|a\,very\,close\,game) = \frac{P(a\,very\,close\,game|sports) \times P(sports)}{P(a\,very\,close\,game)}$$

and

$$P(\overset{\textit{Not}}{sports}|a\,very\,close\,game) = \frac{P(a\,very\,close\,game|\overset{\textit{Not}}{sports}) \times P(\overset{\textit{Not}}{sports})}{P(a\,very\,close\,game)}$$

Naive Bayes: Example

Text	Tag
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

Simplify by removing divisor and compare:

$$P(a\,very\,close\,game|Sports) \times P(Sports)$$

$$P(a\,very\,close\,game|Not\,Sports) \times P(Not\,Sports)$$

Just count how many times **the sentence "A very close game" appears** in the Sports tag, divide it by the total, and obtain P (a very close game | Sports).

Did you see any issue?

"A very close game" doesn't appear in our training data, so this probability is zero.

Being Naive: Here comes the Naive Part!

Naive part: we assume that every word in a sentence is independent of the other ones. This means that we're no longer looking at entire sentences, but rather at individual words.

Assumption: Naive Bayes assumes that all features are **conditionally independent of each other given the class.**

We write this as:

$$P(a\,very\,close\,game) = P(a) \times P(very) \times P(close) \times P(game)$$

The Naïve Bayes Assumption

- Naïve Bayes assumption:
  - Features are independent given class:

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

  - More generally:

$$\boxed{P(X_1 \ldots X_n|Y) = \prod_i P(X_i|Y)}$$

- How many parameters now?
  - Suppose **X** is composed of *n* binary features

Ref: Carlos Guestrin

The Naïve Bayes Classifier

- Given:
  - Prior P(Y)
  - *n* conditionally independent features **X** given the class Y
  - For each X_i, we have likelihood $P(X_i|Y)$

- Decision rule:

$$
\begin{aligned}
y^* = h_{NB}(\mathbf{x}) &= \arg\max_y P(y) P(x_1, \ldots, x_n \mid y) \\
&= \arg\max_y P(y) \prod_i P(x_i|y)
\end{aligned}
$$

Ref: Carlos Guestrin

Text Classification

- Classify e-mails
  - Y = {Spam,NotSpam}
- Classify news articles
  - Y = {what is the topic of the article?}
- Classify webpages
  - Y = {Student, professor, project, …}

- What about the features **X**?
  - The text!

Ref: Carlos Guestrin

Features X Are Entire Document

Article from rec.sport.hockey

```
Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.e
From: xxx@yyy.zzz.edu (John Doe)
Subject: Re: This year's biggest and worst (opinic
Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most
obvious candidate for pleasant surprise is Alex
Zhitnik. He came highly touted as a defensive
defenseman, but he's clearly much more than that.
Great skater and hard shot (though wish he were
more accurate). In fact, he pretty much allowed
the Kings to trade away that huge defensive
liability Paul Coffey. Kelly Hrudey is only the
biggest disappointment if you thought he was any
good to begin with. But, at best, he's only a
mediocre goaltender. A better choice would be
Tomas Sandstrom, though not through any fault of
his own, but because some thugs in Toronto decidec
```


Ref: Carlos Guestrin

NB for Text Classification

- $P(\mathbf{X}|Y)$ is huge!!!
  - Article at least 1000 words, $\mathbf{X}=\{X_1,\ldots,X_{1000}\}$
  - X_i represents i^{th} word in document, i.e., the domain of X_i is entire vocabulary, e.g., Webster Dictionary (or more), 10,000 words, etc.

- NB assumption helps a lot!!!
  - $P(X_i=x_i|Y=y)$ is just the probability of observing word x_i in a document on topic y

$$h_{NB}(\mathbf{x}) = \arg\max_y P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

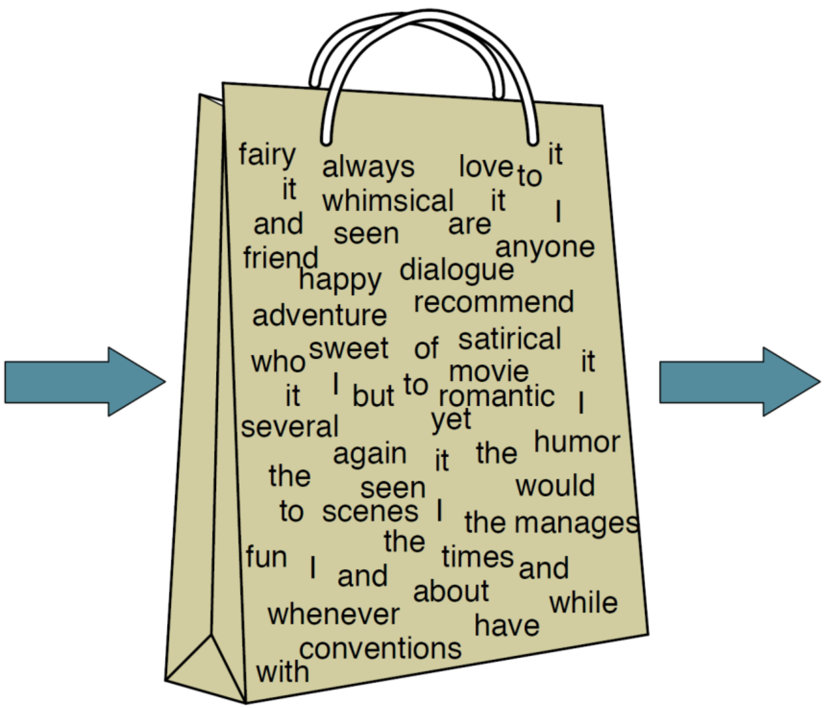
Ref: Carlos Guestrin

Bag of Words Model

- Typical additional assumption – **Position in document doesn't matter**: $P(X_i=x_i|Y=y) = P(X_k=x_i|Y=y)$
  - "Bag of words" model – order of words on the page ignored
  - Sounds really silly, but often works very well!

$$P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
…	…

NB with Bag of Words for Text Classification

- Learning phase:
  - Prior P(Y)
    - Count how many documents you have from each topic (+ prior)
  - $P(X_i|Y)$
    - For each topic, count how many times you saw word in documents of this topic (+ prior)
- Test phase:
  - For each document
    - Use naïve Bayes decision rule

We use log

$$h_{NB}(\mathbf{x}) = \arg\max_y P(y) \prod_{i=1}^{LengthDoc} P(x_i|y)$$

Ref: Carlos Guestrin

Training the Naive Bayes Classifier

- For the class prior P(y) we ask what percentage of the documents in our training set are in each class y. Let Ny be the number of documents in our training data with class c and Ndoc be the total number of documents:

$$P(y) = Ny/Ndoc$$

- To learn the likelihood, we assume a feature is just the existence of a word in the document's bag of words, and so we want $P(x_i|y)$, which we compute as the fraction of times the word x_i appears among all words in all documents of topic y. We first concatenate all documents with category y into one big "category y" text. Then we use the frequency of x_i in this concatenated document to give a maximum likelihood estimate of the probability:

$$P(x_i|y) = \text{count}(x_i, y) \; / \; \text{sum}(\text{for all } i, \text{count}(x_i, y))$$

Address Some Issues

- Imagine we are trying to estimate the likelihood of the word "fantastic" given class positive, but suppose there are no training documents that both contain the word "fantastic" and are classified as positive. Perhaps the word "fantastic" happens to occur (sarcastically?) in the class negative. In such a case the probability for this feature will be zero.

- The simplest solution is the add-one (Laplace) smoothing:

  $P(x_i|y)$ = count(x_i, y) + 1/ sum(for all i, (count(x_i, y) +1))

  = count(x_i, y) + 1/ (sum(for all i, count(x_i, y)) + |V|),

  where |V| is the size of total vocabulary

- Note that it is crucial that the vocabulary V consists of the union of all the word types in all classes, not just the words in one class y

Address Some Issues

- Unknown words
  - Ignore them -- remove them from the test document and not include any probability for them at all.
- Stop words:
  - Very frequent words like the and a.
  - This can be done by sorting the vocabulary by frequency in the training set, and defining the top 20 vocabulary entries as stop words.
  - Alternatively using one of the many predefined stop word lists available online. Each instance of these stop words is removed from both training and test documents as if it had never occurred.
  - In most text classification applications, using a stop word list doesn't improve performance, and so it is more common to make use of the entire vocabulary and not use a stop word list.

Application: Sentimental Analysis

- We'll use a sentiment analysis domain with the two classes positive (+) and negative (-), and take the following miniature training and test documents simplified from actual movie reviews.

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

Sentimental Analysis: Training

- The prior P(c) for the two classes is computed

$$P(-) = \frac{3}{5} \qquad P(+) = \frac{2}{5}$$

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

- The word with doesn't occur in the training set, so we drop it completely. The likelihoods from the training set for the remaining three words "predictable", "no", and "fun", are as follows:

$$P(\text{“predictable”}|-) = \frac{1+1}{14+20} \qquad P(\text{“predictable”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“no”}|-) = \frac{1+1}{14+20} \qquad P(\text{“no”}|+) = \frac{0+1}{9+20}$$

$$P(\text{“fun”}|-) = \frac{0+1}{14+20} \qquad P(\text{“fun”}|+) = \frac{1+1}{9+20}$$

Sentimental Analysis: Testing

- For the test sentence S = "predictable with no fun", after removing the word 'with', the chosen class is therefore computed as follows:

$$
\begin{aligned}
P(-)P(S|-) &= \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5} \\
P(+)P(S|+) &= \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}
\end{aligned}
$$

- The model thus predicts the class negative for the test sentence.

A Summarized NB Algorithm

function TRAIN NAIVE BAYES(D, C) **returns** V, log $P(c)$, log $P(w|c)$

for each class $c \in C$ # Calculate $P(c)$ terms
  N_{doc} = number of documents in D
  N_c = number of documents from D in class c
  $logprior[c] \leftarrow \log \frac{N_c}{N_{doc}}$
  $V \leftarrow$ vocabulary of D
  $bigdoc[c] \leftarrow$ **append**(d) **for** d $\in$ D **with** class c
  for each word w in V # Calculate $P(w|c)$ terms
    $count(w,c) \leftarrow$ # of occurrences of w in $bigdoc[c]$
    $loglikelihood[w,c] \leftarrow \log \frac{count(w,c) + 1}{\sum_{w' \text{ in } V} (count(w',c) + 1)}$
return *logprior*, *loglikelihood*, V

function TEST NAIVE BAYES(*testdoc*, *logprior*, *loglikelihood*, C, V) **returns** best c

for each class $c \in C$
  $sum[c] \leftarrow logprior[c]$
  for each position i in *testdoc*
    $word \leftarrow testdoc[i]$
    if $word \in V$
      $sum[c] \leftarrow sum[c] + loglikelihood[word,c]$
return $\text{argmax}_c \; sum[c]$

Notations:
c <--> y
C <--> Y
w <--> x

Twenty News Groups Results

Given 1000 training documents from each group
Learn to classify new documents according to which newsgroup it came from

comp.graphics	misc.forsale
comp.os.ms-windows.misc	rec.autos
comp.sys.ibm.pc.hardware	rec.motorcycles
comp.sys.mac.hardware	rec.sport.baseball
comp.windows.x	rec.sport.hockey
alt.atheism	sci.space
soc.religion.christian	sci.crypt
talk.religion.misc	sci.electronics
talk.politics.mideast	sci.med
talk.politics.misc	
talk.politics.guns	

Naive Bayes: 89% classification accuracy

Ref: Carlos Guestrin

Example from Twenty News Groups

```
Newsgroup: rec.sport.baseball
document_id: 100521
From: admiral@jhunix.hcf.jhu.edu (Steve C Liu)
Subject: spring records

    The Orioles' pitching staff again is having a fine exhibition season.
Four shutouts, low team ERA, (Well, I haven't gotten any baseball news since
March 14 but anyways) Could they contend, yes. Could they win it all?  Maybe.

But for all those fans of teams with bad spring records, remember Earl
Weaver's first law of baseball (From his book on managing)

No one gives a damn in July if you lost a game in March. :)

BTW, anyone have any idea on the contenders for the O's fifth starter?
It's pretty much set that Sutcliffe, Mussina, McDonald and Rhodes are the
first four in the rotation.
```