DATA ANALYSIS AND VISUALIZATION

INSTRUCTOR: UMME AMMARAH

NATURAL LANGUAGE PROCESSING (NLP)

TEXT CLASSIFICATION

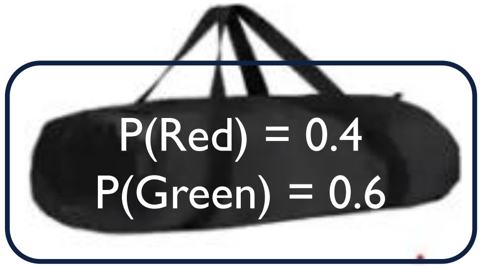
PROBABILITIES (RECAP)

- $P(X = 0) =$
- $P(Y = 3) =$
- $P(X = 1, Y = 2) =$
- $P(Y = 2, X = 1) =$
- $P(X = 1 \mid Y = 2) =$
- $P(Y = 2 \mid X = 1) =$

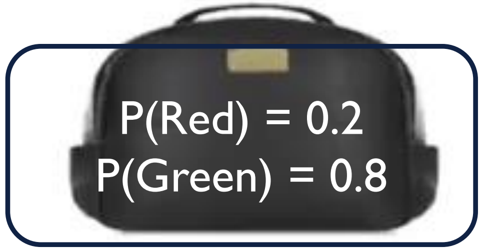
X	Y
0	0
0	1
1	0
1	2
2	3
2	0
2	3
1	3
1	2
0	3
0	2
0	0

BAYES RULE

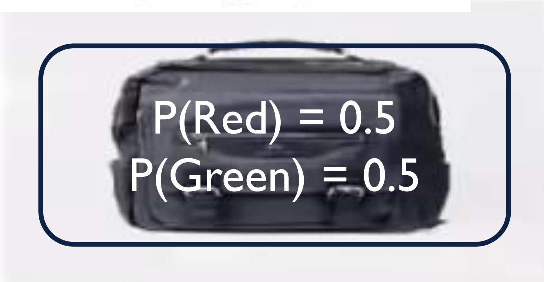
P(bag1) = 0.3



P(bag2) = 0.5



P(bag3)=0.2



- P(bag2) =
- P(Red|bag1) =
- P (Red) =
- P(bag1|Red) =

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

CONT…

- $P(A, B)$ or $P(A \cap B) = ?$
- $P(A, B) = P(A \mid B) * P(B) = P(B \mid A) * P(A)$
- $P(A, B \mid C) = P(A \mid B, C) * P(B \mid C) = P(B \mid A, C) * P(A)$
- $P(A \cap B \cap C) = P(A, B, C)$

$= P(A|B,C) * P(B,C) = P(A|B,C) * P(B|C) * P(C)$

- $P(A \cap B \cap C \cap D) = ?$

CONT.

if all events are independent

- $P(A, B) = P(A) * P(B)$

- $P(A, B, C) =$
- $P(A, B, C, D) =$
- $P(A1, A2, \ldots, An) =$

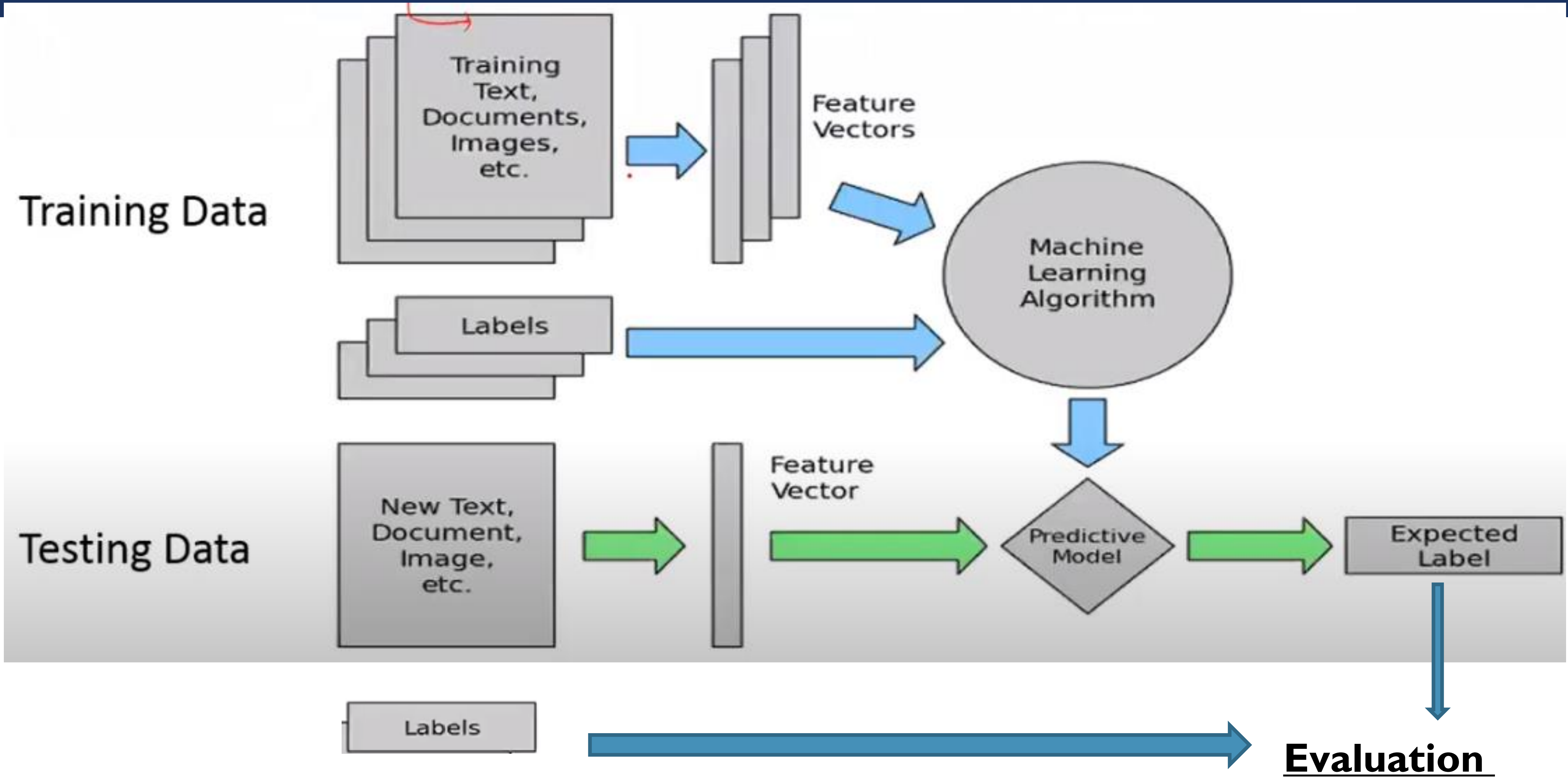
Conditionally independent

- $P(A, B \mid C) = P(A \mid C) * P(B \mid C)$

- $P(A, B, C \mid D) =$
- $P(A1, A2, \ldots, An \mid Z) =$

NAÏVE BAYES CLASSIFIER

NAÏVE BAYES CLASSIFIER



ASSUMPTION

- Conditional independence: Assume the feature probabilities x are independent given the class c.

$$P(x \mid c)$$

$$P(x_1, x_2, \ldots, x_n \mid c)$$

$$P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

NAÏVE BAYES

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)\,P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

NAIVE BAYES CLASSIFIER

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(c \mid d)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c)P(c)}{P(d)}$$

Bayes Rule

$$= \underset{c \in C}{\operatorname{argmax}} P(d \mid c)P(c)$$

Dropping the denominator

EXAMPLE

- Training Data

c(x2=1,y)/c(Y) =2/3

x1	x2	x3	x4	x5	x6	Class
1	0	0	1	1	1	y
1	1	0	1	1	1	y
0	1	0	0	0	0	y
0	0	1	0	0	1	n
1	1	0	1	1	0	n
0	1	0	1	1	0	n

- Testing Data

X1 X2 X3 X4 X5 X6 c

1, 1, 0, 0, 0, 1 y

1, 0, 1, 1, 1, 1 y

y: 2/3* 2/3 *1*1/3*1/3*2/3*3/6=0.016

n: 1/3*2/3*2/3*1/3*1/3*2/3*1/2 =0.005

P(y) = 1/2	P(n) = 1/2
P(x1=1|y) =2/3	P(x1=0|y) =1/3
P(x1=1|n) = 1/3	P(x1=0|n) = 2/3
P(x2=1|y) = 2/3	P(x2=0|y) = 1/3
P(x2=1|n) = 2/3	P(x2=0|n) = 1/3
P(x3=1|y) = 0	P(x3=0|y) = 1
P(x3=1|n) = 1/3	P(x3=0|n) = 2/3
P(x4=1|y) = 2/3	P(x4=0|y) = 1/3
P(x4=1|n) = 2/3	P(x4=0|n) = 1/3
P(x5=1|y) = 2/3	P(x5=0|y) = 1/3
P(x5=1|n) = 2/3	P(x5=0|n) = 1/3
P(x6=1|y) = 2/3	P(x6=0|y) = 1/3
P(x6=1|n) = 1/3	P(x6=0|n) = 2/3

SMOOTHING

- A solution would be **Laplace smoothing** , which is a technique for smoothing categorical data.
- A small-sample correction, or **pseudo-count**, will be incorporated in every probability estimate.
- Consequently, no probability will be zero.
- This is a way of regularizing Naive Bayes, and when the pseudo-count is zero, it is called Laplace smoothing.
- While in the general case it is often called **Lidstone smoothing**.

- Training Data

x1	x2	x3	x4	x5	x6	Class
1	0	0	1	1	1	y
1	1	0	1	1	1	y
0	1	0	0	0	0	y
0	0	1	0	0	1	n
1	1	0	1	1	0	n
0	1	0	1	1	0	n

p(x1=1|Y) = 2+1 / 3+2

3/5

P(y) = 1/2	P(n) = 1/2
P(x1=1|y) =3/5	P(x1=0|y) =2/3
P(x1=1|n) = 2/5	P(x1=0|n) = 3/5
P(x2=1|y) = 3/5	P(x2=0|y) = 2/5
P(x2=1|n) = 3/5	P(x2=0|n) = 2/5
P(x3=1|y) = 1/5	P(x3=0|y) = 4/5
P(x3=1|n) = 2/5	P(x3=0|n) = 3/5
P(x4=1|y) = 3/5	P(x4=0|y) = 2/5
P(x4=1|n) = 3/5	P(x4=0|n) = 2/5
P(x5=1|y) = 3/5	P(x5=0|y) = 2/5
P(x5=1|n) = 3/5	P(x5=0|n) = 2/5
P(x6=1|y) = 3/5	P(x6=0|y) = 2/5
P(x6=1|n) = 2/5	P(x6=0|n) = 3/5

- Testing Data

1, 1, 0, 0, 0, 1 y

1, 0, 1, 1, 1, 1 y

$$\underset{c \in C}{\operatorname{argmax}} \; P(x \mid c) P(c)$$

TEXT CLASSIFICATION

IS THIS SPAM?

Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients:;

Greats News!

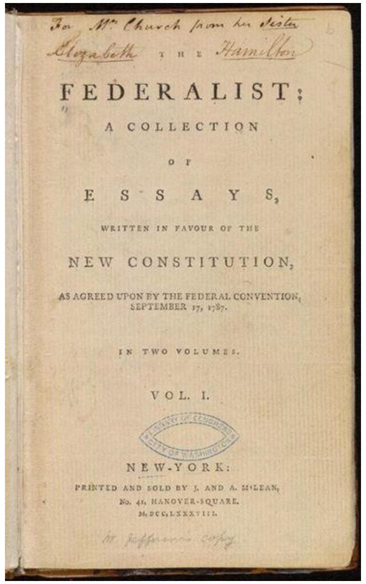
You can now access the latest news by using the link below to login to Stanford University News Forum.

http://www.123contactform.com/contact-form-StanfordNew1-236335.html

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information
about the new services.

WHO WROTE WHICH FEDERALIST PAPERS?

- 1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.
- Authorship of 12 of the letters in dispute
- 1963: solved by Mosteller and Wallace using Bayesian methods



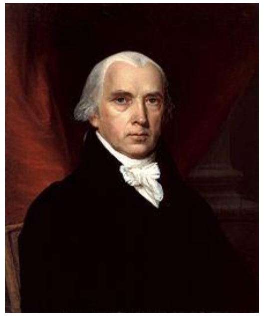
For Mrs Church from her Sister Elizabeth Hamilton

THE FEDERALIST:
A COLLECTION
OF
ESSAYS,
WRITTEN IN FAVOUR OF THE
NEW CONSTITUTION,
AS AGREED UPON BY THE FEDERAL CONVENTION, SEPTEMBER 17, 1787.

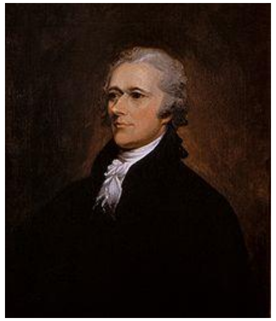
IN TWO VOLUMES.

VOL. I.

NEW-YORK:
PRINTED AND SOLD BY J. AND A. M'LEAN,
No. 41, HANOVER-SQUARE.
M,DCC,LXXXVIII.



James Madison



Alexander Hamilton

WHAT IS THE SUBJECT OF THIS MEDICAL ARTICLE?

MEDLINE Article


Brain and Cognition

Syntactic frame and verb bias in aphasia: Plausibility judgments of undergoer-subject sentences

?

MeSH Subject Category Hierarchy

- Antogonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- …

POSITIVE OR NEGATIVE MOVIE REVIEW?

+ *...zany characters and richly applied satire, and some great plot twists*

− *It was pathetic. The worst part about it was the boxing scenes...*

+ *...awesome caramel sauce and sweet toasty almonds. I love this place!*

− *...awful pizza and ridiculously overpriced...*

POSITIVE OR NEGATIVE MOVIE REVIEW?

+ *...zany characters and **richly** applied satire, and some **great** plot twists*

− *It was **pathetic**. The **worst** part about it was the boxing scenes...*

+ *...**awesome** caramel sauce and sweet toasty almonds. I **love** this place!*

− *...**awful** pizza and **ridiculously** overpriced...*

WHY SENTIMENT ANALYSIS?

- *Movie*: is this review positive or negative?
- *Products*: what do people think about the new iPhone?
- *Public sentiment*: how is consumer confidence?
- *Politics*: what do people think about this candidate or issue?
- *Prediction*: predict election outcomes or market trends from sentiment

SUMMARY: TEXT CLASSIFICATION

- Sentiment analysis
- Spam detection
- Authorship identification
- Language Identification
- Assigning subject categories, topics, or genres…

TEXT CLASSIFICATION: DEFINITION

- *Input*:
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

- *Output*: a predicted class $c \in C$

CLASSIFICATION METHODS: HAND-CODED RULES

- Rules based on combinations of words or other features
  - spam: black-list-address OR (“dollars” AND “you have been selected”)
- Accuracy can be high
  - If rules carefully refined by expert
- But building and maintaining these rules is expensive

CLASSIFICATION METHODS: SUPERVISED MACHINE LEARNING

- *Input:*
  - a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - A training set of m hand-labeled documents $(d_1,c_1),\ldots,(d_m,c_m)$
- *Output:*
  - a learned classifier $\gamma: d \rightarrow c$

CLASSIFICATION METHODS: SUPERVISED MACHINE LEARNING

- Any kind of classifier
  - Naïve Bayes
  - Logistic regression
  - Neural networks
  - k-Nearest Neighbors
  - …

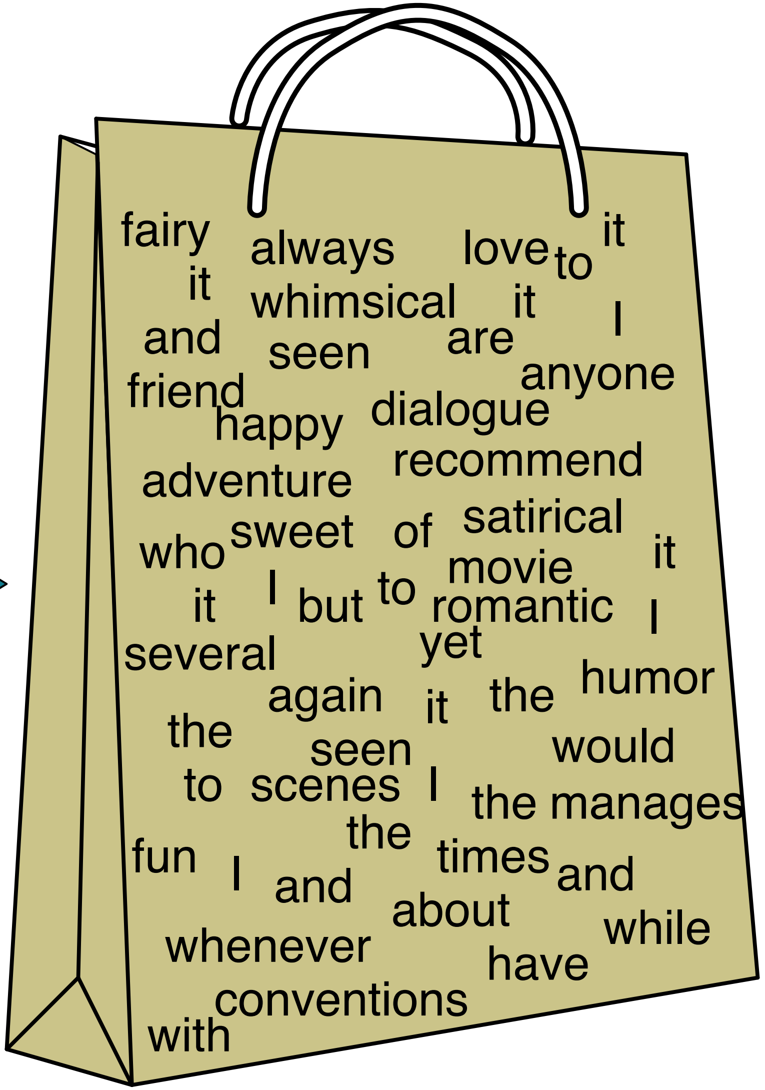
NAÏVE BAYES FOR TEXT CLASSIFICATION

NAIVE BAYES INTUITION

- Simple ("naive") classification method based on Bayes rule
- Relies on very simple representation of document
  - **Bag of words**

THE BAG OF WORDS REPRESENTATION

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
…	…

THE BAG OF WORDS REPRESENTATION

Y(

seen	2
sweet	1
whimsical	1
recommend	1
happy	1
...	...

)=c

BAYES' RULE APPLIED TO DOCUMENTS AND CLASSES

- For a document *d* and a class *c*

$$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}$$

NAIVE BAYES CLASSIFIER

"Likelihood" "Prior"

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(d \mid c) P(c)$$

$$= \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

Document d represented as features x1..xn

NAÏVE BAYES CLASSIFIER

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}}\, P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$O(|X|^n \bullet |C|)$ parameters

How often does this class occur?

Could only be estimated if a very, very large number of training examples was available.

We can just count the relative frequencies in a corpus

MULTINOMIAL NAIVE BAYES INDEPENDENCE ASSUMPTIONS

$$P(x_1, x_2, \ldots, x_n \mid c)$$

- **Bag of Words assumption**: Assume position doesn't matter
- **Conditional Independence**: Assume the feature probabilities $P(x_i|c_j)$ are independent given the class *c.*

$$P(x_1, \ldots, x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet P(x_3 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

MULTINOMIAL NAIVE BAYES CLASSIFIER

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c) P(c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c_j) \prod_{x \in X} P(x \mid c)$$

APPLYING MULTINOMIAL NAIVE BAYES CLASSIFIERS TO TEXT CLASSIFICATION

positions $\leftarrow$ all word positions in test document

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

PROBLEMS WITH MULTIPLYING LOTS OF PROBS

- There's a problem with this:

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

- Multiplying lots of probabilities can result in floating-point underflow!
- .0006 * .0007 * .0009 * .01 * .5 * .000008….
- Idea: Use logs, because log(*ab*) = log(*a*) + log(*b*)
- We'll sum logs of probabilities instead of multiplying probabilities!

WE ACTUALLY DO EVERYTHING IN LOG SPACE

Instead of this: $$c_{NB} = \operatorname*{argmax}_{c_j \in C} P(c_j) \prod_{i \in positions} P(x_i | c_j)$$

This: $$c_{\mathrm{NB}} = \operatorname*{argmax}_{c_j \in C} \left[\log P(c_j) + \sum_{i \in \mathrm{positions}} \log P(x_i | c_j) \right]$$

Notes:

1) Taking log doesn't change the ranking of classes!

   The class with highest probability also has highest log probability!

2) It's a linear model:

   Just a max of a sum of weights: a **linear** function of the inputs

   So naive bayes is a **linear classifier**

LEARNING THE MULTINOMIAL NAIVE BAYES MODEL

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

PARAMETER ESTIMATION

$$\hat{P}(w_i \mid c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

fraction of times word w_i appears among all words in documents of topic c_j

- Create mega-document for topic *j* by concatenating all docs in this topic
  - Use frequency of *w* in mega-document

PROBLEM WITH MAXIMUM LIKELIHOOD

- What if we have seen no training documents with the word ***fantastic*** and classified in the topic **positive** (***thumbs-up)***?

$$\hat{P}(\text{"fantastic"} \mid \text{positive}) = \frac{count(\text{"fantastic", positive})}{\sum_{w \in V} count(w, \text{positive})} = 0$$

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

$$c_{MAP} = \text{argmax}_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

LAPLACE (ADD-1) SMOOTHING FOR NAÏVE BAYES

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} \left(count(w, c) + 1 \right)}$$

$$= \frac{count(w_i, c) + 1}{\left(\sum_{w \in V} count(w, c) \right) + |V|}$$

MULTINOMIAL NAÏVE BAYES: LEARNING

- From training corpus, extract *Vocabulary*

- Calculate $P(c_j)$ terms
  - For each c_j in C do

    $docs_j \leftarrow$ all docs with class $= c_j$

$$P(c_j) \leftarrow \frac{|docs_j|}{|\text{total \# documents}|}$$

- Calculate $P(w_k \mid c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - For each word w_k in *Vocabulary*

    $n_k \leftarrow$ # of occurrences of w_k in $Text_j$

$$P(w_k \mid c_j) \neg \frac{n_k + a}{n + a|Vocabulary|}$$

UNKNOWN WORDS

- What about unknown words
  - that appear in our test data
  - but not in our training data or vocabulary?
- We **ignore** them
  - Remove them from the test document!
  - Pretend they weren't there!
  - Don't include any probability for them at all!
- Why don't we build an unknown word model?
  - It doesn't help: knowing which class has more unknown words is not generally helpful!

STOP WORDS

- Some systems ignore stop words
  - **Stop words:** very frequent words like *the* and *a*.
    - Sort the vocabulary by word frequency in training set
    - Call the top 10 or 50 words the **stopword list**.
    - Remove all stop words from both training and test sets
      - As if they were never there!
- But removing stop words doesn't usually help
  - So in practice most NB algorithms use **all** words and **don't** use stopword lists

LET'S DO A WORKED SENTIMENT EXAMPLE!

	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

	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

1. Prior from training:

$$\hat{P}(c_j) = \frac{N_{c_j}}{N_{total}}$$

P(-) = 3/5
P(+) = 2/5

2. Drop "with"

3. Likelihoods from training:

$$p(w_i|c) = \frac{count(w_i, c) + 1}{(\sum_{w \in V} count(w, c)) + |V|}$$

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

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

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

4. Scoring the test set:

$$\boxed{P(-)P(S|-) = \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5}}$$

$$P(+)P(S|+) = \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

EVALUATION

EVALUATION

- Let's consider just binary text classification tasks
- Imagine you're the CEO of Delicious Pie Company
- You want to know what people are saying about your pies
- So you build a "Delicious Pie" tweet detector
  - Positive class: tweets about Delicious Pie Co
  - Negative class: all other tweets

THE 2-BY-2 CONFUSION MATRIX

		gold standard labels		
		gold positive	gold negative	
system output labels	system positive	**true positive**	**false positive**	$\textbf{precision} = \frac{tp}{tp+fp}$
	system negative	**false negative**	**true negative**	
		$\textbf{recall} = \frac{tp}{tp+fn}$		$\textbf{accuracy} = \frac{tp+tn}{tp+fp+tn+fn}$

EVALUATION: ACCURACY

- Why don't we use **accuracy** as our metric?
- Imagine we saw 1 million tweets
  - 100 of them talked about Delicious Pie Co.
  - 999,900 talked about something else
- We could build a dumb classifier that just labels every tweet "not about pie"
  - It would get 99.99% accuracy!!! Wow!!!!
  - But useless! Doesn't return the comments we are looking for!
  - That's why we use **precision** and **recall** instead

EVALUATION: PRECISION

- % of items the system detected (i.e., items the system labeled as positive) that are in fact positive (according to the human gold labels)

$$\textbf{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

EVALUATION: RECALL

- % of items actually present in the input that were correctly identified by the system.

$$\textbf{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

WHY PRECISION AND RECALL

- Our dumb pie-classifier
  - Just label nothing as "about pie"

Accuracy=99.99%

but

Recall = 0

  - (it doesn't get any of the 100 Pie tweets)

Precision and recall, unlike accuracy, emphasize true positives:

  - finding the things that we are supposed to be looking for.

A COMBINED MEASURE: F

- F measure: a single number that combines P and R:

$$F_\beta = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

- We almost always use balanced F_1 (i.e., $\beta = 1$)

$$\mathrm{F}_1 = \frac{2PR}{P+R}$$