Mid-Term

- Thursday, October 5 (Class Period)
- In-Person Sections: Regular room (SH118)
- On-line Sections: Arrange location <u>in advance</u> with IIT-Online. Look for email from Charles Scott
- Please arrange needed accommodations in advance; we may not be able to help on the morning of exam
- Covers readings and lecture slides (with focus on material in that appears in both)
- Format:
  - Closed Book – No printouts, no electronics
  - Multiple Choice, True/False, or similar

Text Categorization and Naïve Bayes

CS-585

Natural Language Processing

Sonjia Waxmonsky

(with slides from William W. Cohen and Chris Manning)

TEXT CATEGORIZATION (CLASSIFICATION)

Text Classification: Definition

- The classifier:
  - *Input*: a document x
  - *Output*: a predicted class y from some fixed set of labels $y_1, \dots, y_k$
- The learner:
  - *Input:* a set of *m* hand-labeled documents $(x_1, y_1), \dots, (x_m, y_m)$
  - *Output:* a learned classifier $f: x \rightarrow y$

Text Classification: Examples

- Classify news stories as *World, US, Business, SciTech, Sports, Entertainment, Health, Other*
- Classify business names by industry.
- Classify student essays as *A,B,C,D*, or *F*.
- Classify email as *Spam, Other*.
- Classify email to tech staff as *Mac, Windows, ..., Other*.
- Classify pdf files as *ResearchPaper, Other*
- Classify documents as *WrittenByReagan, GhostWritten*
- Classify movie reviews as *Favorable,Unfavorable,Neutral*.
- Classify technical papers as *Interesting, Uninteresting*.
- Classify web sites of companies by Standard Industrial Classification (SIC) code.
- Classify jokes as *Funny, NotFunny*.

Text Classification: Examples

- Best-studied benchmark: *Reuters-21578* newswire stories
  - 9603 train, 3299 test documents, 80-100 words each, 93 classes

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS

BUENOS AIRES, Feb 26

Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

→ Categories: grain, wheat (of 93 binary choices)

Representing text for classification

f(

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
BUENOS AIRES, Feb 26

Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

)=y

?

What is the ~~best~~ simplest useful representation for the document *x* being classified?

Bag of words representation

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS
BUENOS AIRES, Feb 26

Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

→ Categories: grain, wheat

Bag of words representation

xxxxxxxxxxxxxxxxxxx GRAIN/OILSEED xxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxx

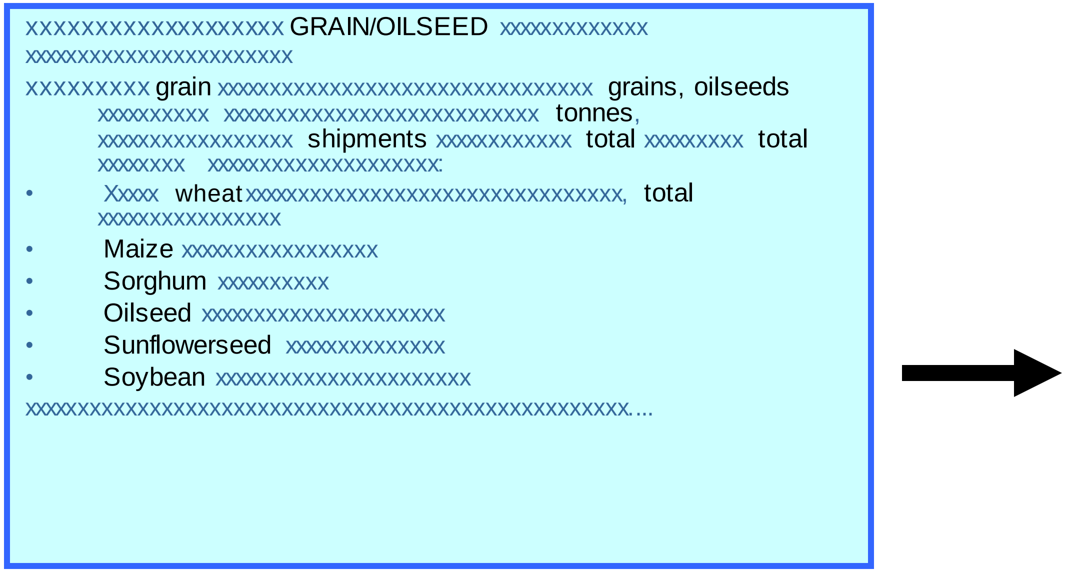
xxxxxxxxx grain xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx grains, oilseeds xxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxx tonnes, xxxxxxxxxxxxxxxxx shipments xxxxxxxxxxxx total xxxxxxxxx total xxxxxxxx xxxxxxxxxxxxxxxxxxxx:

- Xxxxx wheat xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, total xxxxxxxxxxxxxxxx
- Maize xxxxxxxxxxxxxxxxx
- Sorghum xxxxxxxxxx
- Oilseed xxxxxxxxxxxxxxxxxxxxx
- Sunflowerseed xxxxxxxxxxxxxx
- Soybean xxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx....

→ Categories: grain, wheat

Bag of words representation



word	*freq*
grain(s)	3
oilseed(s)	2
total	3
wheat	1
maize	1
soybean	1
tonnes	1
...	...

Categories: grain, wheat

Bag of N-grams?

xxxxxxxxxxxxxxxxxxx GRAIN/OILSEED xxxxxxxxxxxxx
BUENOS AIRES xxxxxxxxxxxxxxxx
xxxxxxxxx grain xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx grains, oilseeds
xxxxxxxxxx xxxxxxxcrop registries xxxxx tonnes, xxxxxxxxxxxxxxxxx
shipments xxxxxxxxxxxx total xxxxxxxxx total
xxxxxxxx xxxxxxxxxxxxxxxxxxxx:

- Xxxxx wheat xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, total xxxxxxxxxxxxxxxx
- Maize xxxxxxxxxxxxxxxxx
- Sorghum xxxxxxxxxx
- Oilseed xxxxxxxxxxxxxxxxxxxxx
- Sunflowerseed xxxxxxxxxxxxxx
- Soybean xxxxxxxxxxxxxxxxxxxxxx

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx....

word	*freq*
grain(s)	3
oilseed(s)	2
total	3
wheat	1
…	…
Buenos aires	1
crop registries	1
…	…

N-gram features:

- Detect collections
- Detect negation and local context (e.g. "not arson")
- Increase vocabulary size

NAÏVE BAYES

Text Classification with Naive Bayes

- Represent document x as set of $(w_i, Count(w_i))$ pairs:
  - $x = \{(\text{grain}, 3), (\text{wheat}, 1), \ldots, (\text{the}, 6)\}$
- For each y, build a probabilistic model $\Pr(X|Y = y)$ of "documents" in class y
  - $\Pr(X = \{(\text{grain}, 3), \ldots\}|Y = \text{wheat}) = \ldots$
  - $\Pr(X = \{(\text{grain}, 3), \ldots\}|Y = \text{nonWheat}) = \ldots$
- To classify, find the y which was most likely to *generate* x—*i.e.*, which gives x the best score according to $\Pr(x|y)$
  - $f(x) = \text{argmax}_y \Pr(x|y) \times \Pr(y)$

Bayes Rule

$$\Pr(y \mid x) \cdot \Pr(x) = \Pr(x, y) = \Pr(x \mid y) \cdot \Pr(y)$$

$$\Rightarrow \qquad \Pr(y \mid x) = \frac{\Pr(x \mid y) \cdot \Pr(y)}{\Pr(x)}$$

$$\Rightarrow \quad \arg\max_y \Pr(y \mid x) = \arg\max_y \Pr(x \mid y) \cdot \Pr(y)$$

Text Classification with Naive Bayes

- How to estimate $\Pr(X|Y)$?
- *Simplest useful* process to generate a bag of words:
  - pick word 1 according to $\Pr(W|Y)$
  - repeat for word 2, 3, ...
  - each word is generated *independently* of the others (which is clearly not true) but means

$$\Pr(w_1,\ldots,w_n \mid Y = y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y = y)}$$

How to estimate Pr(W | Y)?

Two Unreasonable Assumptions

- Bag-of-words:

  The order of the words in document *d* makes no difference (but repetitions do)
- Conditional Independence:

  Words appear independently of each other, given the document class

  (e.g., if you see "car", the word "drive" is no more likely to appear than if you saw "dog")

Text Classification with Naive Bayes

- How to estimate Pr(X | Y) ?

$$\Pr(w_1,\ldots,w_n \mid Y = y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y = y)}$$

Estimate $Pr(w \mid y)$ by looking at the data...

$$\Pr(W = w \mid Y = y) = \frac{\text{count}(W = w \text{ and } Y = y)}{\text{count}(Y = y)}$$

Simple Smoothing

- If X contains a vocabulary word that does not occur with class $Y = y$ in the training:

$P(X|Y = y) = 0$, no matter what else is there!

- Solution:
  - Assign small probability to unseen words,
  - Taking away probability from seen words
  - Every word that occurred N times with class $Y = y$, we will pretend actually occurred $N + \alpha$ times

N-gram Smoothing

- Goal: Estimate the probability of n-grams with zero count
- Witten-Bell smoothing
  - Interpolates probabilities between order n and (n-1)
  - N-grams with zero counts assigned a non-zero probability based on lower-order n-gram counts

$$p_{WB}(c_i|c_{i-n+1}^{i-1}) = \lambda_{c_{i-n+1}^{i-1}} p_{ML}(c_i|c_{i-n+1}^{i-1}) + (1 - \lambda_{c_{i-n+1}^{i-1}}) p_{WB}(c_i|c_{i-n+2}^{i-1})$$

Text Classification with Naive Bayes

- How to estimate Pr(X | Y) ?

$$\Pr(w_1,\ldots,w_n \mid Y=y) = \prod_{i=1}^{n} \underbrace{\Pr(w_i \mid Y=y)}$$

... and also imagine α "pseudo-occurrences" of w_i in class $Y = y$

- $\Pr(w_i|Y=y) = \frac{count(w_i \wedge Y=y)+\alpha}{count(Y=y)+\alpha|V|}$

Text Classification with Naive Bayes

- How to estimate Pr(X | Y) ?

$$\Pr(w_1,\ldots,w_n \mid Y=y)=\prod_{i=1}^{n}\underbrace{\Pr(w_i \mid Y=y)}$$

For instance, α=3

- $\Pr(w_i|Y=y)=\dfrac{count(w_i \wedge Y=y)+3}{count(Y=y)+3|V|}$

Avoiding Underflow

- Consider:
  - Many docs have more than 100 words
  - Word probabilities will each be <0.1
  - So, P(X | Y)<10^{-100} for any document X
  - ➔UNDERFLOW!!

- Solution: $\log a > \log b$ iff $a > b$

  Use $\log[P(X|Y)P(Y)] = \log P(X|Y) + \log P(Y)$

  $$\log P(X|Y) = \sum_{w_i \varepsilon X} \log P(w_i|Y)$$

Text Classification with Naive Bayes

- Putting this together:

```
for each document xi with label yi
    d_count[yi]++
    d_count++
    for each word wij in xi
        w_count[wij][yi]++
        w_count[yi]++
```

  - to classify a new $x = w_1 \ldots w_n$, pick y with top *score*:

$$score(y, w_1, \cdots, w_n) = \log \frac{d\_count[y]}{d\_count} + \sum_{i=1}^{n} \log \frac{w\_count[w_i][y] + \alpha}{w\_count[y] + \alpha|V|}$$

key point: we only need counts for words that actually appear in *x*

Naïve Bayes: Putting it all together

$$\log\big(P(Y = y, X)\big) = \log(P(X|Y = y)) + \log(P(Y = Y))$$

$$\log\big(P(Y = y)\big) = \log\frac{d\_count[y]}{d\_count}$$

$$\log\big(P(X|Y = y)\big) = \sum_{w \in X} \log\frac{w\_count[w][y] + \alpha}{w\_count[y] + \alpha|V|}$$

Some numerical
care required

$$P(Y = y|X) = \frac{P(Y = y, X)}{\sum_{y' \in Y} P(Y = y', X)}$$

WebKB Experiment (1998)

- Classify webpages from CS departments into:
  - student, faculty, course, project
- Train on ~5,000 hand-labeled web pages
  - Cornell, Washington, U.Texas, Wisconsin
- Crawl and classify a new site (CMU) using Naïve Bayes

Results

	Student	Faculty	Person	Project	Course	Departmt
Extracted	180	66	246	99	28	1
Correct	130	28	194	72	25	1
Accuracy:	72%	42%	79%	73%	89%	100%

Faculty

associate	0.00417
chair	0.00303
member	0.00288
ph	0.00287
director	0.00282
fax	0.00279
journal	0.00271
recent	0.00260
received	0.00258
award	0.00250

Students

resume	0.00516
advisor	0.00456
student	0.00387
working	0.00361
stuff	0.00359
links	0.00355
homepage	0.00345
interests	0.00332
personal	0.00332
favorite	0.00310

Courses

homework	0.00413
syllabus	0.00399
assignments	0.00388
exam	0.00385
grading	0.00381
midterm	0.00374
pm	0.00371
instructor	0.00370
due	0.00364
final	0.00355

Departments

departmental	0.01246
colloquia	0.01076
epartment	0.01045
seminars	0.00997
schedules	0.00879
webmaster	0.00879
events	0.00826
facilities	0.00807
eople	0.00772
postgraduate	0.00764

Research Projects

investigators	0.00256
group	0.00250
members	0.00242
researchers	0.00241
laboratory	0.00238
develop	0.00201
related	0.00200
arpa	0.00187
affiliated	0.00184
project	0.00183

Others

type	0.00164
jan	0.00148
enter	0.00145
random	0.00142
program	0.00136
net	0.00128
time	0.00128
format	0.00124
access	0.00117
begin	0.00116

Naive Bayes Summary

- Pros:
  - Very fast and easy-to-implement
  - Well-understood formally & experimentally
    - see "Naive (Bayes) at Forty", Lewis, ECML98
- Cons:
  - Seldom gives the very best performance
  - "Probabilities" $\Pr(y|x)$ are not accurate
    - Probabilities tend to be close to zero or one

ALTERNATE BAG-OF-WORDS METHODS

Gradient Boosting Machines (GBM)

- Ensemble method based on decision trees
- Gradient Boosting: Build models that sequentially correct errors of previous model
- Text features: BOW N-gram vocabulary
- Benefits for NLP:
  - Easy to blend text and structured data inputs
    - Handles outliers well without preprocessing
  - Interpretable? Allows for ranking of features
  - Trains (relatively) quickly while managing interactions
- Popular implementations: XGBoost, Catboost, LightGBM

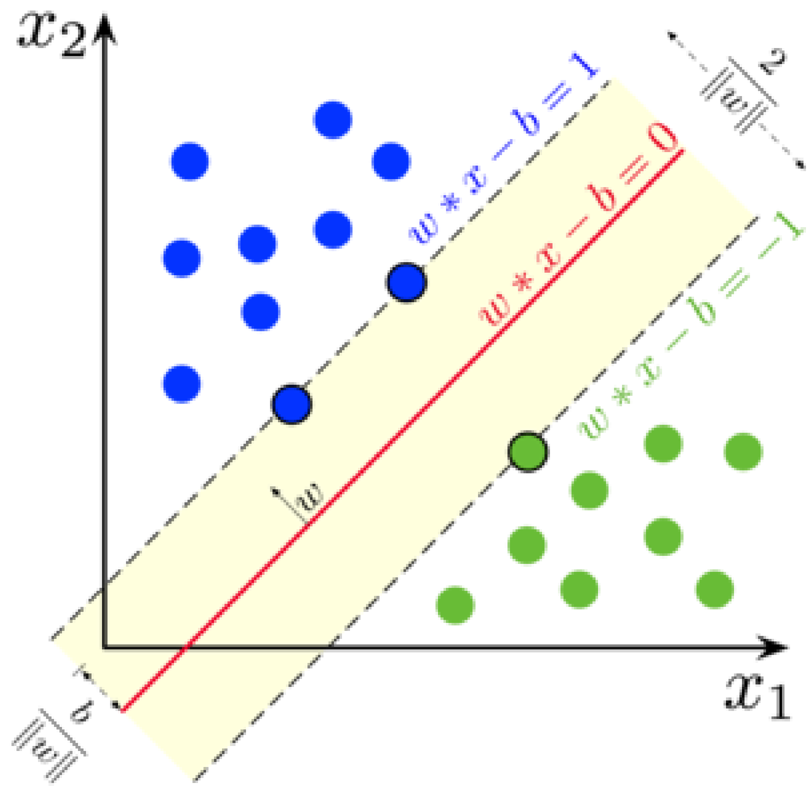
Support Vector Machines (SVMs)

Goal:

maximize separation between classes

Kernel-based SVMs:

Able to model non-linear relationship with input



https://en.wikipedia.org/wiki/Support_vector_machine

TEXT CLASSIFIER EVALUATION

Metrics: Binary Classification

	Prediction: False	Prediction: True
Ground Truth: False	True Negative (TN)	False Positive (FP)
Ground Truth: True	False Negative (FN)	True Positive (TP)

RECALL:
TP/(TP+FN)

PRECISION:
TP/(TP+FP)

F1-SCORE:
2 * P * R / (P + R)

F-BETA SCORE:
(1+beta^2) * P * R / (beta^2 P + Recall)

Threshold-Free metrics

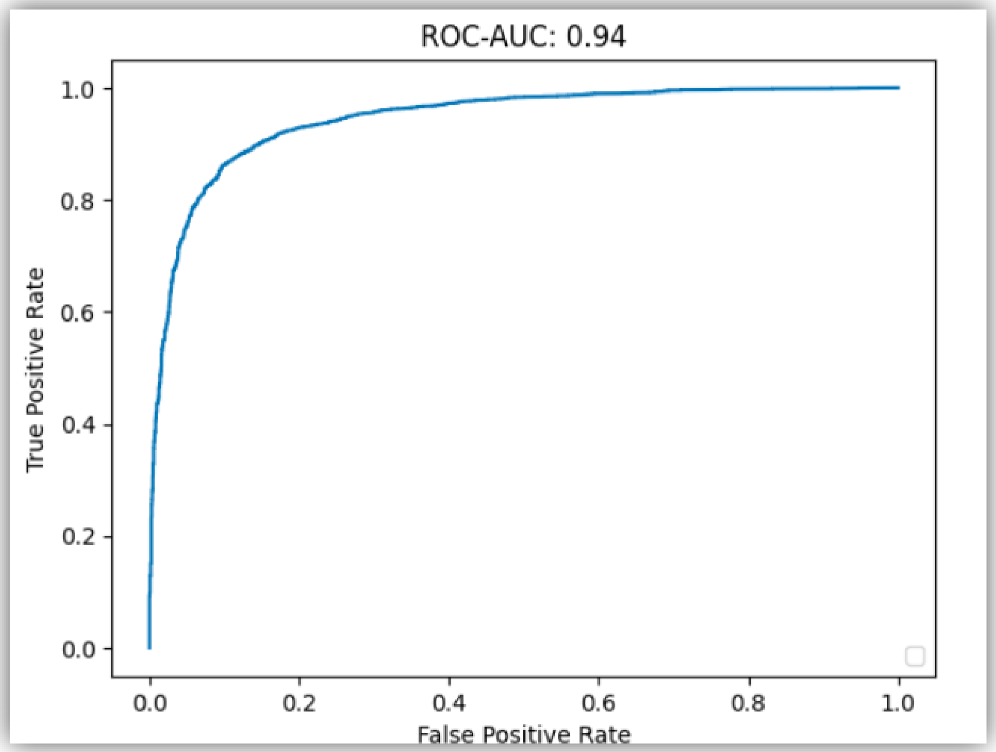
ROC-AUC score

ROC:

"Receiver Operator Characteristic"

AUC:

"Area Under Curve"



Precision-Recall Curve