

# CS 6120/CS4120: Natural Language Processing

Instructor: Prof. Lu Wang

College of Computer and Information Science

Northeastern University

Webpage: [www.ccs.neu.edu/home/luwang](http://www.ccs.neu.edu/home/luwang)

# Outline

- Text Categorization/Classification
- Naïve Bayes
- Evaluation

# Positive or negative movie review?



- unbelievably disappointing
- Full of zany characters and richly applied satire, and some great plot twists



- this is the greatest screwball comedy ever filmed



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

# Male or female author?

1. By 1925 present-day Vietnam was divided into three parts under French colonial rule. The southern region embracing Saigon and the Mekong delta was the colony of Cochin-China; the central area with its imperial capital at Hue was the protectorate of Annam...
2. Clara never failed to be astonished by the extraordinary felicity of her own name. She found it hard to trust herself to the mercy of fate, which had managed over the years to convert her greatest shame into one of her greatest assets...

# Text Classification

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language Identification
- Sentiment analysis
- ...

# Text Classification: definition

- *Input:*

- a document  $d$
- a fixed set of classes  $C = \{c_1, c_2, \dots, c_J\}$

- *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 “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  $C = \{c_1, c_2, \dots, c_J\}$
- A training set of  $m$  hand-labeled documents  $(d_1, y_1), \dots, (d_m, y_m)$ ,  $y_i$  is in  $C$

- *Output:*

- a learned classifier  $\gamma: d \rightarrow c$



# Classification Methods: Supervised Machine Learning

- Any kind of classifier
  - Naïve Bayes
  - Logistic regression
  - Support-vector machines
  - k-Nearest Neighbors
  - Neural networks
  - ...

# Outline

- Text Categorization/Classification
- • Naïve Bayes
- Evaluation

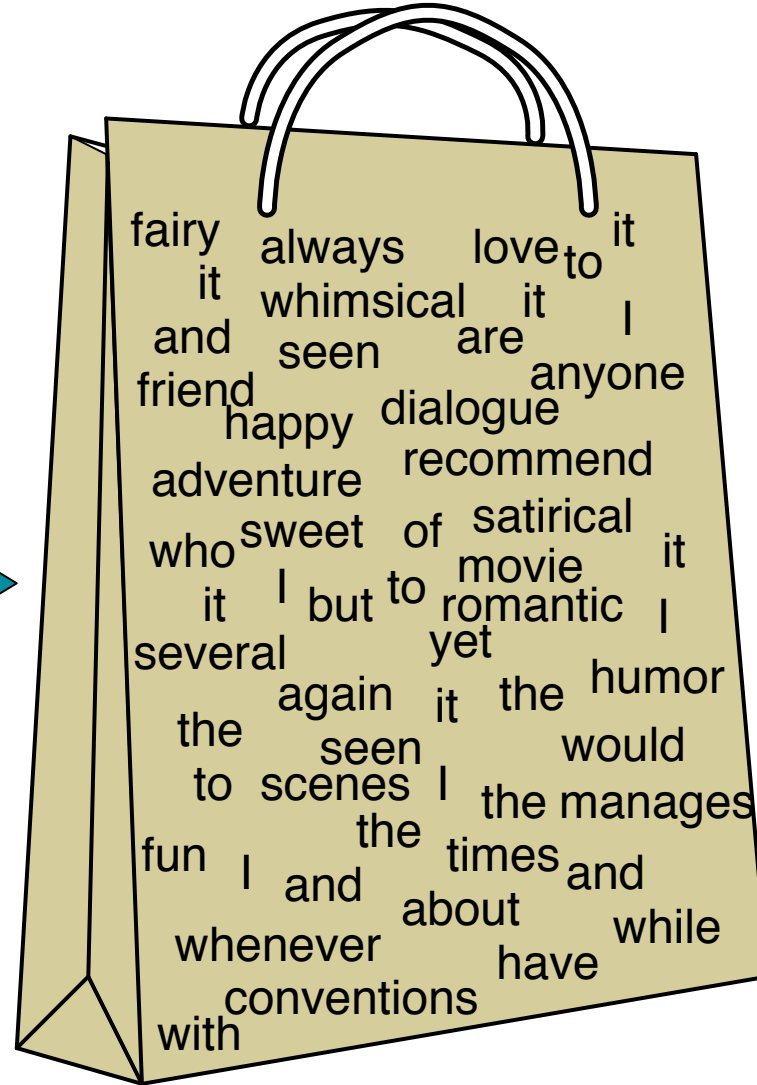
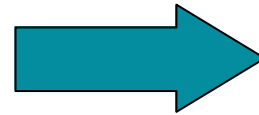
# Naïve Bayes Classifier

# Naïve Bayes Intuition

- Simple (“naïve”) 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



# Bayes' Rule Applied to Documents and Classes

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

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

# Naïve Bayes Classifier (I)

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

MAP is “maximum a posteriori” = most likely class

$$= \operatorname{argmax}_{c \in C} \frac{P(d | c)P(c)}{P(d)}$$

Bayes Rule

$$= \operatorname{argmax}_{c \in C} P(d | c)P(c)$$

Dropping the denominator



# Naïve Bayes Classifier (I)

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

MAP is “maximum a posteriori” = most likely class

$$= \operatorname{argmax}_{c \in C} \frac{P(d | c)P(c)}{P(d)}$$

Bayes Rule

$$= \operatorname{argmax}_{c \in C} P(d | c)P(c)$$

Dropping the denominator

Why we can do this?

# Naïve Bayes Classifier (II)

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

$$= \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c)P(c)$$

Document  $d$   
represented as  
features  $x_1..x_n$

# Naïve Bayes Classifier (IV)

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c) P(c)$$

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

$|X|$  represents the maximum number of possible values for  $x_i$

$$P(x_1, x_2, \dots, x_n | c)$$

- **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  $c$ .

$$P(x_1, \dots, x_n | c) = P(x_1 | c) \cdot P(x_2 | c) \cdot P(x_3 | c) \cdot \dots \cdot P(x_n | c)$$

# Multinomial Naïve Bayes Classifier

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c) P(c)$$

$$c_{NB} = \operatorname{argmax}_{c \in C} P(c) \prod_{x \in X} P(x | c)$$

# Applying Multinomial Naive Bayes Classifiers to Text Classification

positions  $\leftarrow$  all word positions in test document

$$c_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i \in \text{positions}} P(x_i | c_j)$$

# Learning for Naïve Bayes Model

# Learning the Multinomial Naïve Bayes Model

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

$$\hat{P}(c_j) = \frac{\text{doccount}(C = c_j)}{N_{doc}}$$

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



# Parameter estimation

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

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

# Problem with Maximum Likelihood

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

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

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

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

# Laplace (add-1) smoothing for Naïve Bayes

$$\begin{aligned}\hat{P}(w_i | c) &= \frac{\mathit{count}(w_i, c) + 1}{\sum_{w \in V} (\mathit{count}(w, c) + 1)} \\ &= \frac{\mathit{count}(w_i, c) + 1}{\left( \sum_{w \in V} \mathit{count}(w, c) \right) + |V|}\end{aligned}$$

# 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  $P(w_k | c_j)$  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  $Text_j$

$$P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

# 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  $P(w_k | c_j)$  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  $Text_j$

$$P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha |Vocabulary|}$$

A more general form: add- $\alpha$  smoothing!

# Naïve Bayes and Language Modeling

- Naïve bayes classifiers can use any sort of feature
  - URL, email address, dictionaries, network features
- But if, as in the previous slides
  - We use **only** word features
  - we use **all** of the words in the text (not a subset)
- Then
  - Naïve bayes has an important similarity to language modeling.

# Each class = a unigram language model

- Assigning each word:  $P(\text{word} \mid c)$
- Assigning each sentence:  $P(\text{sentence} \mid c) = \prod P(\text{word} \mid c)$

Class *pos*

0.1 I

0.1 love

0.01 this

0.05 fun

0.1 film

...

<u>I</u>	<u>love</u>	<u>this</u>	<u>fun</u>	<u>film</u>
0.1	0.1	.05	0.01	0.1

$$P(\text{sentence} \mid \text{pos}) = 0.0000005$$

# Naïve Bayes as a Language Model

- Which class assigns the higher probability to s?

Model pos	
0.1	I
0.1	love
0.01	this
0.05	fun
0.1	film

Model neg	
0.2	I
0.001	love
0.01	this
0.005	fun
0.1	film

<u>I</u>	<u>love</u>	<u>this</u>	<u>fun</u>	<u>film</u>
0.1	0.1	0.01	0.05	0.1
0.2	0.001	0.01	0.005	0.1

$$P(s | \text{pos}) > P(s | \text{neg})$$



# An Example

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w|c) = \frac{\text{count}(w,c) + 1}{\text{count}(c) + |V|}$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

**Priors:**

$$P(c) = \frac{3}{4}$$

$$P(j) = \frac{1}{4}$$

**Conditional Probabilities:**

$$P(\text{Chinese}|c) = \frac{(5+1)}{(8+6)} = \frac{6}{14} = \frac{3}{7}$$

$$P(\text{Tokyo}|c) = \frac{(0+1)}{(8+6)} = \frac{1}{14}$$

$$P(\text{Japan}|c) = \frac{(0+1)}{(8+6)} = \frac{1}{14}$$

$$P(\text{Chinese}|j) = \frac{(1+1)}{(3+6)} = \frac{2}{9}$$

$$P(\text{Tokyo}|j) = \frac{(1+1)}{(3+6)} = \frac{2}{9}$$

$$P(\text{Japan}|j) = \frac{(1+1)}{(3+6)} = \frac{2}{9}$$

**Choosing a class:**

$$P(c|d5) \propto \frac{3}{4} * \left(\frac{3}{7}\right)^3 * \frac{1}{14} * \frac{1}{14} \approx 0.0003$$

$$P(j|d5) \propto \frac{1}{4} * \left(\frac{2}{9}\right)^3 * \frac{2}{9} * \frac{2}{9} \approx 0.0001$$

# Summary: Naive Bayes is Not So Naive

- Very Fast, low storage requirements
- Robust to Irrelevant Features
  - Irrelevant Features cancel each other without affecting results
- Very good in domains with many equally important features
- Optimal if the independence assumptions hold:
  - If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- A good dependable baseline for text classification

# Outline

- Text Categorization/Classification
- Naïve Bayes
- • Evaluation

# Evaluation

# The 2-by-2 contingency table (or confusion matrix)

	correct	not correct
selected	tp (true positive)	fp (false positive)
not selected	fn (false negative)	tn (true negative)

For example,

- Which set of documents are related to the topic of NLP?
- Which set of documents are written by Shakespeare?

# The 2-by-2 contingency table

	correct	not correct
selected	tp	fp
not selected	fn	tn

# Precision and recall

- **Precision:** % of selected items that are correct,  $tp/(tp+fp)$   
**Recall:** % of correct items that are selected,  $tp/(tp+fn)$

	correct	not correct
selected	tp	fp
not selected	fn	tn



# A combined measure: F-measure or F-score

- A combined measure that assesses the P/R tradeoff is F measure (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}$$

- People usually use balanced F1 measure
  - i.e.,  $\alpha = \frac{1}{2}$ ,  $F = \frac{2PR}{P+R}$

# Text Classification Evaluation

# More Than Two Classes: Sets of binary classifiers

- Dealing with **any-of** or **multivalued** classification
  - A document can belong to 0, 1, or >1 classes.
- For each class  $c \in C$ 
  - Build a classifier  $\gamma_c$  to distinguish  $c$  from all other classes  $c' \in C$
- Given test doc  $d$ ,
  - Evaluate it for membership in each class using each  $\gamma_c$
  - $d$  belongs to **any** class for which  $\gamma_c$  returns true

# More Than Two Classes: Sets of binary classifiers

- **One-of** or **multinomial** classification
  - Classes are mutually exclusive: each document in exactly one class
- For each class  $c \in C$ 
  - Build a classifier  $\gamma_c$  to distinguish  $c$  from all other classes  $c' \in C$
- Given test doc  $d$ ,
  - Evaluate it for membership in each class using each  $\gamma_c$
  - $d$  belongs to the **one** class with maximum score

# Confusion matrix c

- For each pair of classes  $\langle c_1, c_2 \rangle$  how many documents from  $c_1$  were incorrectly assigned to  $c_2$ ?
  - $c_{3,2}$ : 90 wheat documents incorrectly assigned to poultry

Docs in test set	Assigned UK	Assigned poultry	Assigned wheat	Assigned coffee	Assigned interest	Assigned trade
True UK	95	1	13	0	1	0
True poultry	0	1	0	0	0	0
True wheat	10	90	0	1	0	0
True coffee	0	0	0	34	3	7
True interest	0	1	2	13	26	5
True trade	0	0	2	14	5	10

# Per class evaluation measures

## Recall:

Fraction of docs in class  $i$  classified correctly:

$$\frac{c_{ii}}{\sum_j c_{ij}}$$

## Precision:

Fraction of docs assigned class  $i$  that are actually about class  $i$ :

$$\frac{c_{ii}}{\sum_j c_{ji}}$$

## Accuracy: (1 - error rate)

Fraction of docs classified correctly:

$$\frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$$

# Micro- vs. Macro-Averaging

- If we have more than one class, how do we combine multiple performance measures into one quantity?
- **Macroaveraging:** Compute performance for each class, then average.
- **Microaveraging:** Collect decisions for all classes, compute contingency table, evaluate.

# Micro- vs. Macro-Averaging: Example

## Class 1

	Truth: yes	Truth: no
Classifier: yes	10	10
Classifier: no	10	970

## Class 2

	Truth: yes	Truth: no
Classifier: yes	90	10
Classifier: no	10	890

## Micro Ave. Table

	Truth: yes	Truth: no
Classifier: yes	100	20
Classifier: no	20	1860



# Micro- vs. Macro-Averaging: Example

Class 1

	Truth: yes	Truth: no
Classifier: yes	10	10
Classifier: no	10	970

Class 2

	Truth: yes	Truth: no
Classifier: yes	90	10
Classifier: no	10	890

Micro Ave. Table

	Truth: yes	Truth: no
Classifier: yes	100	20
Classifier: no	20	1860

- Macroaveraged precision:  $(0.5 + 0.9)/2 = 0.7$
- Microaveraged precision:  $100/120 = .83$

# Development Test Sets and Cross-validation

Training set

Development/tuning/held-out Set

Test Set

Metric: P/R/F1 or Accuracy

Unseen test set

- avoid overfitting ('tuning to the test set')
- more conservative estimate of performance

Cross-validation over multiple splits

- Handle sampling errors from different datasets
- Pool results over each split
- Compute pooled dev set performance

Training Set Dev Test

Training Set

Dev Test

Dev Test

Training Set

Test Set