Lecture 5: Text Classification (I)

Instructor: Stephanie Schoch

CSCI 680: Natural Language Processing
Department of Computer Science, College of William & Mary
September 4, 2026

WILLIAM & MARY

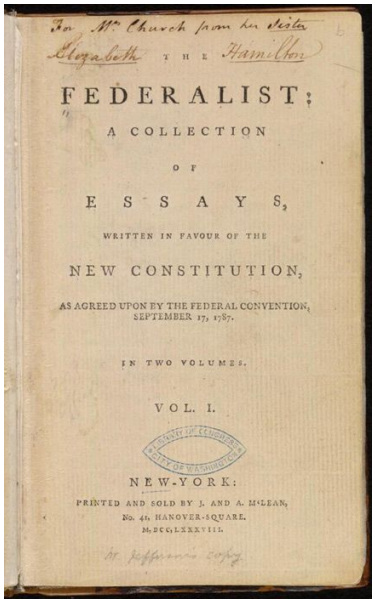
Many slides are adapted from Jurafsky & Martin (3rd ed.)

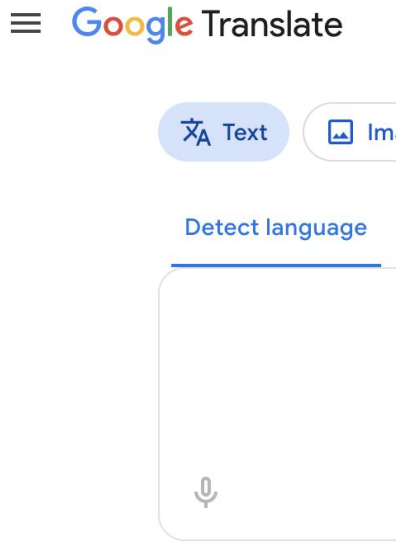
Lecture Outline

- The Task of Text Classification
- Naive Bayes
- Logistic Regression

The Task of Text Classification

Text Classification





Assigning subject categories, topics, or genres, Spam detection, Etc.

Authorship Identification

Sentiment analysis

Language identification

Question: Can language modeling be viewed as text classification?

Text Classification: Definition

Input:

- a "document" (which can be any text) d
- a fixed set of classes $Y = \{y_1, y_2, \ldots, y_J\}$

Output: a predicted class $\hat{y} \in Y$

The hat or circumflex notation $\hat{y}$ is used to refer to an estimated or predicted value

Most Common Classification Method: Supervised Machine Learning

Training:

Input:

- a fixed set of classes $Y = \{y_1, y_2, \ldots, y_J\}$
- a training set of *m hand-labeled inputs* $(x^{(1)},y^{(1)}),\ldots,(x^{(m)},y^{(m)})$

Output:

- a learned classifier $\gamma:d \rightarrow \hat{y}$

Inference *or* **Test**

Input: a document d

Output: a class $\hat{y}$

Supervised Machine Learning for Classification

Many kinds of classifiers!

- Logistic regression
- Naive Bayes
- Neural networks
- k-nearest neighbors
- LLMs
- Fine-tuned as classifiers
- Prompted to give a classification

Naive Bayes 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

$\gamma($

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

$) = y$

Bayes' Rule Applied to Documents and Classes

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

posterior | likelihood | prior

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

marginal

Naive Bayes Classifier (I)

$$y_{MAP} = \arg\max_{y \in Y} P(y \mid d)$$

MAP is “maximum a posteriori” = most likely class

$$y_{MAP} = \arg\max_{y \in Y} \frac{P(d \mid y)P(y)}{P(d)}$$

Bayes Rule

$$y_{MAP} = \arg\max_{y \in Y} P(d \mid y)P(y)$$

Dropping the denominator (marginal is the same across classes)

Naive Bayes Classifier (II)

likelihood

prior

$$y_{MAP} = \arg\max_{y \in Y} P(d \mid y) P(y)$$

$$\arg\max_{y \in Y} P(x_1, x_2, \ldots, x_n \mid y) P(y)$$

Document *d* represented as features $x_1, \ldots, x_n$

Multinomial Naive Bayes Independence Assumptions

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

Bag-of-Words Assumption: Assume that position doesn't matter

Conditional Independence: Assume the feature probabilities $P(x_i|y_j)$ are independent given the class *y*.

$$P(x_1, x_2, \ldots, x_n \mid y) = P(x_1 \mid y)P(x_2 \mid y) \cdots P(x_n \mid y)$$

Multinomial Naive Bayes Classifier

$$y_{MAP} = \arg\max_{y \in Y} P(x_1, x_2, \ldots, x_n \mid y) P(y)$$

$$y_{NB} = \arg\max_{y_j \in Y} P(y_j) \prod_{i=1}^{n} P(x_i \mid y_j)$$

Applying Multinomial Naive Bayes Classifiers to Text Classification

positions ← all word positions in test document

$$y_{NB} = \arg\max_{y_j \in Y} P(y_j) \prod_{i \in positions} P(x_i \mid y_j)$$

$$y_{NB} = \arg\max_{y_j \in Y} \left[\log P(y_j) + \sum_{i \in positions} \log P(x_i \mid y_j) \right]$$

Learning the Multinomial Naive Bayes Model

Use frequencies in the data:

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

Documents belonging to class y_j

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

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

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

Laplace (add-1) smoothing for Naive Bayes

With smoothing!

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

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

Multinomial Naive Bayes: Learning

From training corpus, extract *Vocabulary V*

Calculate $P(y_j)$ terms:

- For each y_j in Y do
  $docs_j \leftarrow$ all docs with class y_j

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

Calculate $P(w_k \mid y_j)$ 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 $Text_j$

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

Unknown Words and Stop 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!
  - Don't include any probability for them at all
- 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

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

Prior from training

Likelihoods from training

Scoring the test set

$$\hat{P}(y_j) = \frac{N_{y_j}}{N_{total}} = \frac{count(w_i, y) + 1}{\left(\sum_{w \in V} count(w, y)\right) + |V|}$$

$$\arg\max_{y_j \in Y} \left[\log P(y_j) + \sum_{i \in positions} \log P(w_i \mid y_j) \right]$$

Worked out Sentiment Example

Prior from training

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

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

Likelihoods from training

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

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

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

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

Negative class score

$$\log P(-) + \log P(S \mid -)$$
$$= \log \frac{3}{5} + \log \frac{2}{34} + \log \frac{2}{34} + \log \frac{1}{34}$$
$$\approx -9.70$$

Positive class score

$$\log P(+) + \log P(S \mid +)$$
$$= \log \frac{2}{5} + \log \frac{1}{29} + \log \frac{1}{29} + \log \frac{2}{29}$$
$$\approx -10.33$$

Scoring the test set

$$\arg\max_{y_j \in Y} \left[\log P(y_j) + \sum_{i \in positions} \log P(w_i \mid y_j) \right]$$

Decision $-9.70 > -10.33 \quad \Rightarrow \quad \hat{y} = -$

Logistic Regression Classification